Thesis

Thiyaghessan

2023-04-30

Analysis Replication Guide

This markdown file replicates all plots produced in the accompanying thesis.

Load packages and configure environment

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

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

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.1     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- 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 errors
##
## Attaching package: 'igraph'
##
##
## The following objects are masked from 'package:lubridate':
##
##     %--%, 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:dplyr':
##
##     as_data_frame, groups, union
##
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
##
## The following object is masked from 'package:base':
##
##     union

## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'

##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
##
## Loading required package: network
##
## 'network' 1.18.1 (2023-01-24), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information
##
##
## Attaching package: 'network'
##
## The following objects are masked from 'package:igraph':
##
##     %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
##     get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,
##     is.directed, list.edge.attributes, list.vertex.attributes,
##     set.edge.attribute, set.vertex.attribute
##
##
## 'ergm' 4.4.0 (2023-01-26), part of the Statnet Project
## * 'news(package="ergm")' for changes since last version
```

```
## * 'citation("ergm")' for citation information
## * 'https://statnet.org' for help, support, and other information
##
## 'ergm' 4 is a major update that introduces some backwards-incompatible
## changes. Please type 'news(package="ergm")' for a list of major
## changes.
##
## Version:  1.38.6
## Date:     2022-04-06
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").
##
## Attaching package: 'texreg'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

Global Variables

```
# These podcasts are far away from the center of the network and are excluded from visualization
Extra_Exclude <- c("POLITICO Energy", "Washington Today")

# Set color pallette for network visualizations
Viz_Colors <- c("darkred", "tomato", "gray80", "skyblue", "darkblue", "gray80")

# Configurations for Degree Distribution Graphs
Degree_Ideologies <- c("Reactionary", "Conservative", "Moderate", "Liberal", "Radical", "NULL")
Degree_Colors <- c("tomato", "skyblue", "gray50", "black", "darkblue", "darkred")

# Configurations for Betweenness Centrality Distribution Plot
Btwn_Ideologies <- c("Conservative", "Liberal", "Moderate", "Radical", "Reactionary", "NULL")
Btwn_Colors <- c("tomato", "skyblue", "gray50", "darkblue", "darkred", "black")

# Configurations for Power Law Ideologies
Plaw_Ideologies <- c("Reactionary", "Conservative", "Moderate", "Liberal", "Radical")
```

Required Functions

```
# Function to create a dataframe that aggregates degree distribution by ideology
make_deg_df <- function(g, mode, ideologies){
 for (i in 1:6){
   deg_ideology <- igraph::degree(g,
                                  v = V(g)$ideology == i,
                                  mode = mode)
   deg_df <- as.data.frame(table(deg_ideology)) %>%
     mutate(ideology = ideologies[i])
   if (i == 1){
```

```
    master_deg_df <- deg_df
    } else {
      master_deg_df <- rbind(master_deg_df, deg_df)
    }
 }
  return(master_deg_df)
}

# Function to create a dataframe that contains power law fits for degree distributions by ideology
make_deg_plaw_df <- function(ideologies, deg_df){
  for (id in ideologies){
  df = as.data.frame(power.law.fit(deg_df$Freq[deg_df$ideology == id]))
  if (id == "Reactionary"){
    plaw_df = df
  } else {
    plaw_df = rbind(plaw_df, df)
  }
  }
  return(plaw_df)
}
```

Data Preprocessing

```
# Import csv containing information on podcasts
pod_df <- read.csv("podcast_hosts.csv") # Data on podcasts
pod_bias_df <- read.csv("podcast_bias.csv") # Data on the level of Bias
hosts_df <- read.csv("podcast_hostattribs.csv") # Data on the podcast hosts' occupations/source of fame

# Merge dataframes
nodes_df <- left_join(pod_df, pod_bias_df, by = "podcasts")

# Summarise the number of unique distributors present in the dataset
distrib_df <- nodes_df %>%
  group_by(parent) %>%
  summarise(count = n())
# Isolate distributors who own/represent more than one podcast in the dataset
distributors <- distrib_df$parent[distrib_df$count > 1]

# Add this information to nodes_df and recode other variables accordingly
nodes_df <- nodes_df %>%
  mutate(distributor = ifelse(parent %in% distributors, parent,
                              ifelse(parent != "Independent", "Single Company", "Independent")),
         distributor_code = ifelse(distributor == "Independent", 3,
                                   ifelse(distributor == "Single Company", 2, 1)),
         bias_ratio = numbiased / (numbiased + numunbiased),
         bias_ratio = ifelse(is.nan(bias_ratio), mean(bias_ratio, na.rm = TRUE), bias_ratio),
         bias_ratio = ifelse(bias_ratio > 0, bias_ratio, mean(bias_ratio)),
         id_code = ifelse(main_ideology == "reactionary", 1,
                          ifelse(main_ideology == "conservative", 2,
                                 ifelse(main_ideology == "moderate", 3,
                                        ifelse(main_ideology == "liberal", 4,
                                               ifelse(main_ideology == "radical", 5, 6))))))
```

```r
# Import data frame with podcast collaborations
collab_df <- read.csv("podcast_collabs.csv") %>%
  rename("from" = "From",
         "to" = "To") %>%
  group_by(from, to) %>%
  summarise(weight = n()) # weight edges by number of repeat appearances
```

```
## `summarise()` has grouped output by 'from'. You can override using the
## `.groups` argument.
```

```r
# Import data summarizing total number of cross-partisan/homophilic collaborations
crosspart_df <- read.csv("crosspart2.csv") %>%
                group_by(crosspart) %>%
                summarise(count = sum(count),
                          weight = sum(weight),
                          homophilic = mean(homophilic),
                          cross = mean(cross))

# create social network graph
g_full <- graph_from_data_frame(collab_df,
                           directed = TRUE,
                           vertices = nodes_df$podcasts) # Directed edges

# Set node attributes
g_full <- g_full %>%
  set_vertex_attr("ideology",
                  index = V(g_full),
                  value = nodes_df$id_code) %>%
  set_vertex_attr("distributor",
                  index = V(g_full),
                  value = nodes_df$distributor_code) %>%
  set_vertex_attr("bias_count",
                  index = V(g_full),
                  value = nodes_df$numbiased) %>%
  set_vertex_attr("bias_ratio",
                  index = V(g_full),
                  value = nodes_df$bias_ratio)
```

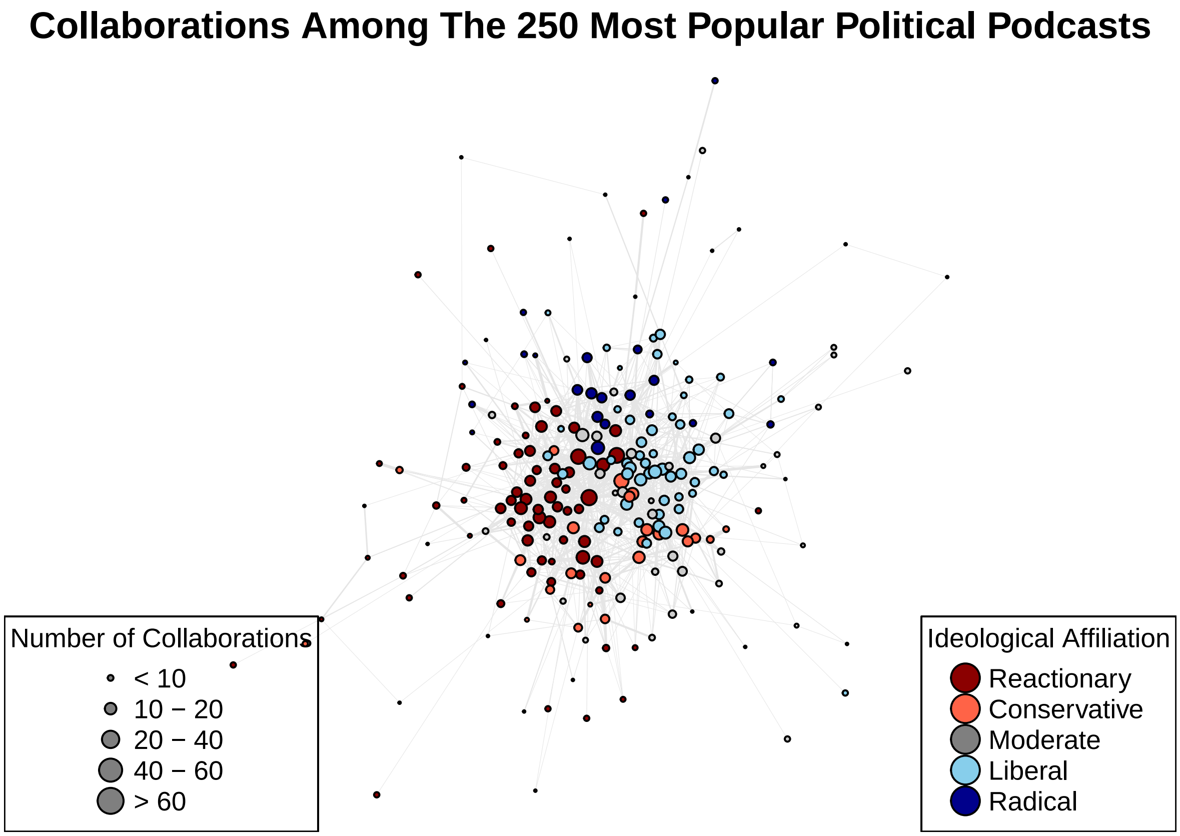
Data Preprocessing for Visualization

The first step is to load in the necessary data and functions

Network Visualization

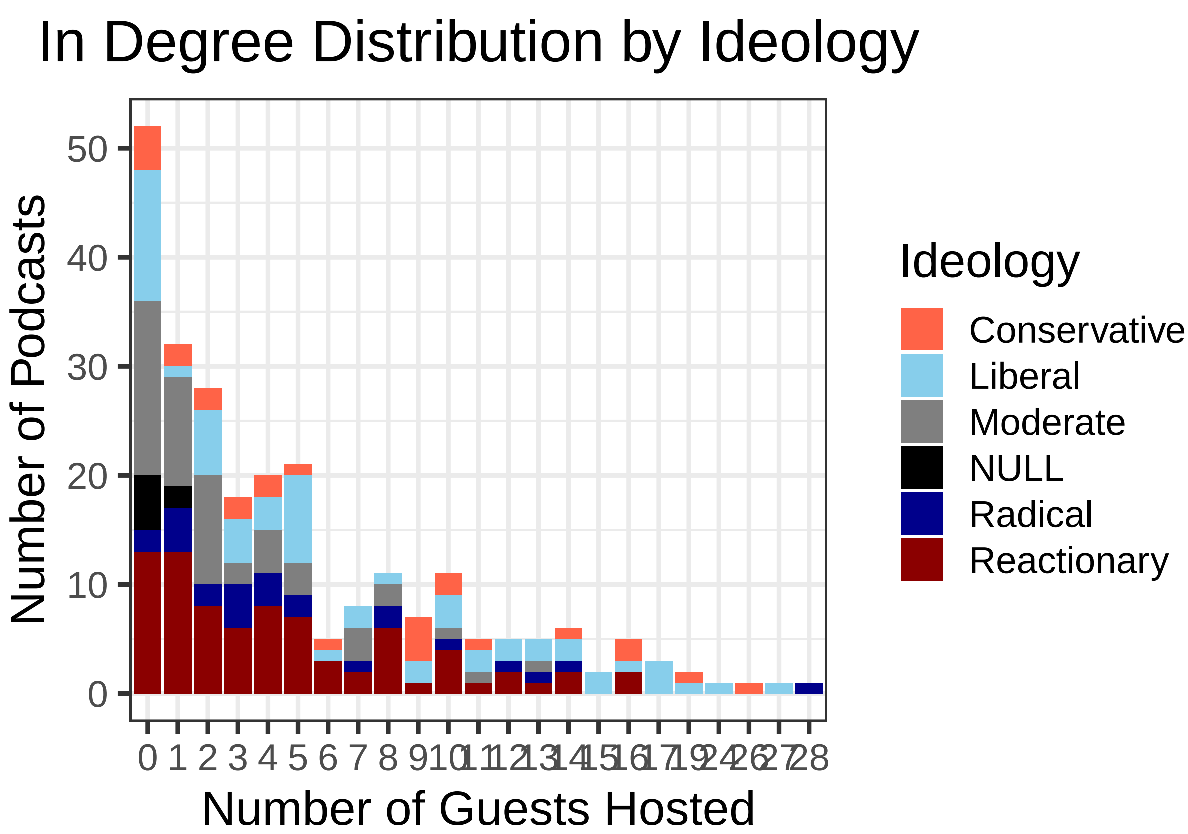
```r
par(mar=c(0,0,0,0)+1)
plot(g_viz,
     vertex.label = ifelse(deg[V(g_viz)$name] > 100, V(g_viz)$name, NA), # Display labels for podcasts
     vertex.label.family = "Arial",
```

```r
     vertex.label.color = "black",
     vertex.label.cex = 0.5,
     edge.color = "gray90",
     main = "Collaborations Among The 250 Most Popular Political Podcasts",
     margin = 0)

legend(
  "bottomright",
  legend = c("Reactionary", "Conservative", "Moderate", "Liberal", "Radical"),
  pt.bg  = c("darkred", "tomato", "gray50", "skyblue", "darkblue"),
  pt.cex = c(2, 2, 2),
  pch    = 21,
  cex    = 0.8,
  bty    = "o",
  title  = "Ideological Affiliation"
)

legend(
  "bottomleft",
  legend = c("< 10", "10 - 20", "20 - 40", "40 - 60", "> 60"),
  pt.bg  = "gray50",
  pt.cex = c(0.4, 0.8, 1.2, 1.6, 1.8),
  pch    = 21,
  cex    = 0.8,
  bty    = "o",
  title  = "Number of Collaborations"
)
```



Indegree Distribution Plot

```r
# Create DataFrame with distribution
indeg_df <- make_deg_df(g_full, "in", Degree_Ideologies)
# Reorder factor levels
indeg_df$deg_ideology <- factor(indeg_df$deg_ideology,
                                levels = c("0", "1", "2", "3", "4", "5", "6", "7",
                                           "8", "9" , "10", "11", "12", "13", "14",
                                           "15", "16", "17", "19", "24", "26", "27", "28"))
# Bar Plot
indeg_dist_barplot <- ggplot(indeg_df,
                             aes(x = deg_ideology,
                                 y = Freq,
                                 fill = ideology)) +
                        geom_bar(stat = "identity",
                                 position = "stack") +
                        xlab("Number of Guests Hosted") +
                        ylab("Number of Podcasts") +
                        labs(title = "In Degree Distribution by Ideology") +
                        scale_fill_manual(name = "Ideology",
                                          values = Degree_Colors) +
                        theme_bw(base_size = 18) +
                        theme(plot.title = element_text(hjust = 0.5))
indeg_dist_barplot
```




```r
# Compute mean and median
sprintf("The Mean of this distribution is %.2f", mean(as.integer(indeg_df$deg_ideology)))
```

```
## [1] "The Mean of this distribution is 9.01"
```

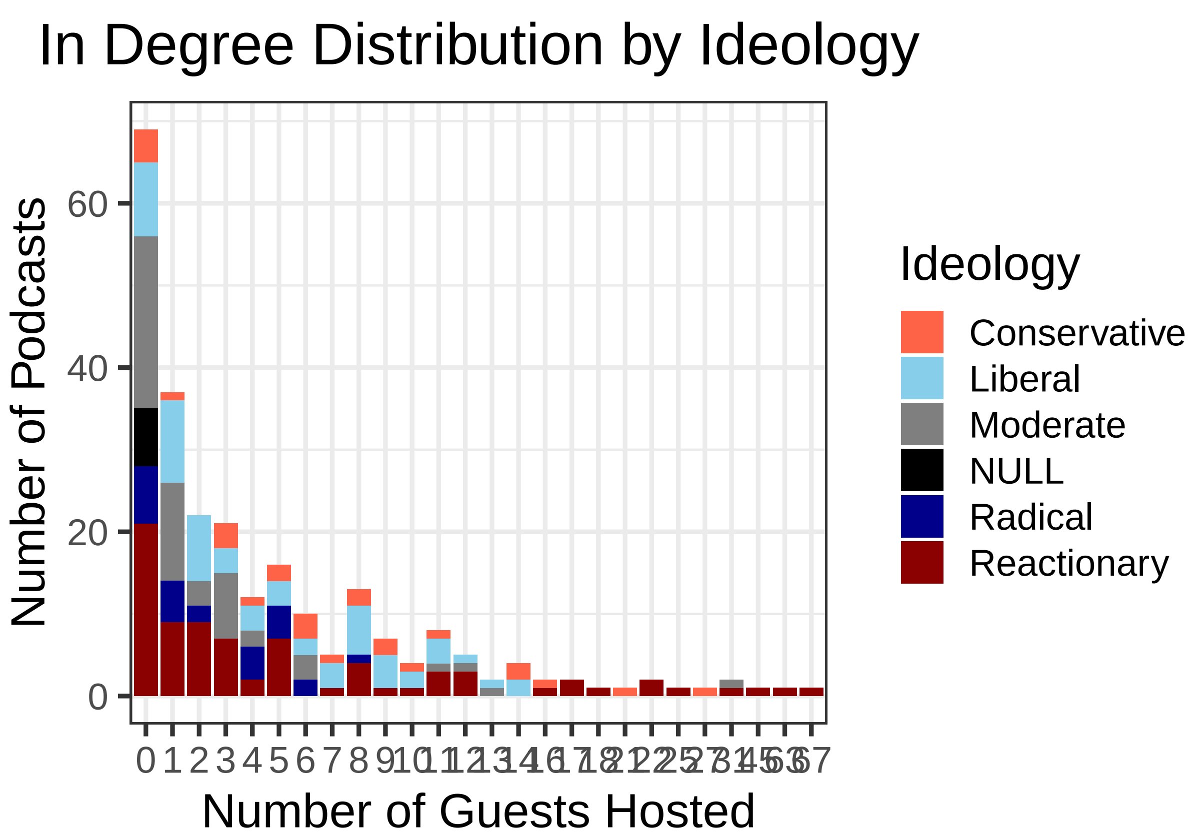
```r
sprintf("The Median of this distribution is %.2f", median(as.integer(indeg_df$deg_ideology)))
```

```
## [1] "The Median of this distribution is 8.00"
```

Outdegree Distribution Plot

```r
# Create DataFrame with distribution
outdeg_df <- make_deg_df(g_full, "out", Degree_Ideologies)
# Reorder factor levels
outdeg_df$deg_ideology <- factor(outdeg_df$deg_ideology,
                                levels = c("0", "1", "2", "3", "4", "5", "6", "7",
                                           "8", "9" , "10", "11", "12", "13","14","16", "17",
                                           "18", "21", "22", "25", "27", "31", "45", "63", "67"))
# Bar Plot
outdeg_dist_barplot <- ggplot(outdeg_df,
                              aes(x = deg_ideology,
                                  y = Freq,
```

```
                                 fill = ideology)) +
                       geom_bar(stat = "identity",
                                position = "stack") +
                       xlab("Number of Guests Hosted") +
                       ylab("Number of Podcasts") +
                       labs(title = "In Degree Distribution by Ideology") +
                       scale_fill_manual(name = "Ideology",
                                         values = Degree_Colors) +
                       theme_bw(base_size = 18) +
                       theme(plot.title = element_text(hjust = 0.5))
outdeg_dist_barplot
```




```
# Compute mean and median
sprintf("The Mean of this distribution is %.2f", mean(as.integer(outdeg_df$deg_ideology)))
```

```
## [1] "The Mean of this distribution is 9.55"
```

```
sprintf("The Median of this distribution is %.2f", median(as.integer(outdeg_df$deg_ideology)))
```

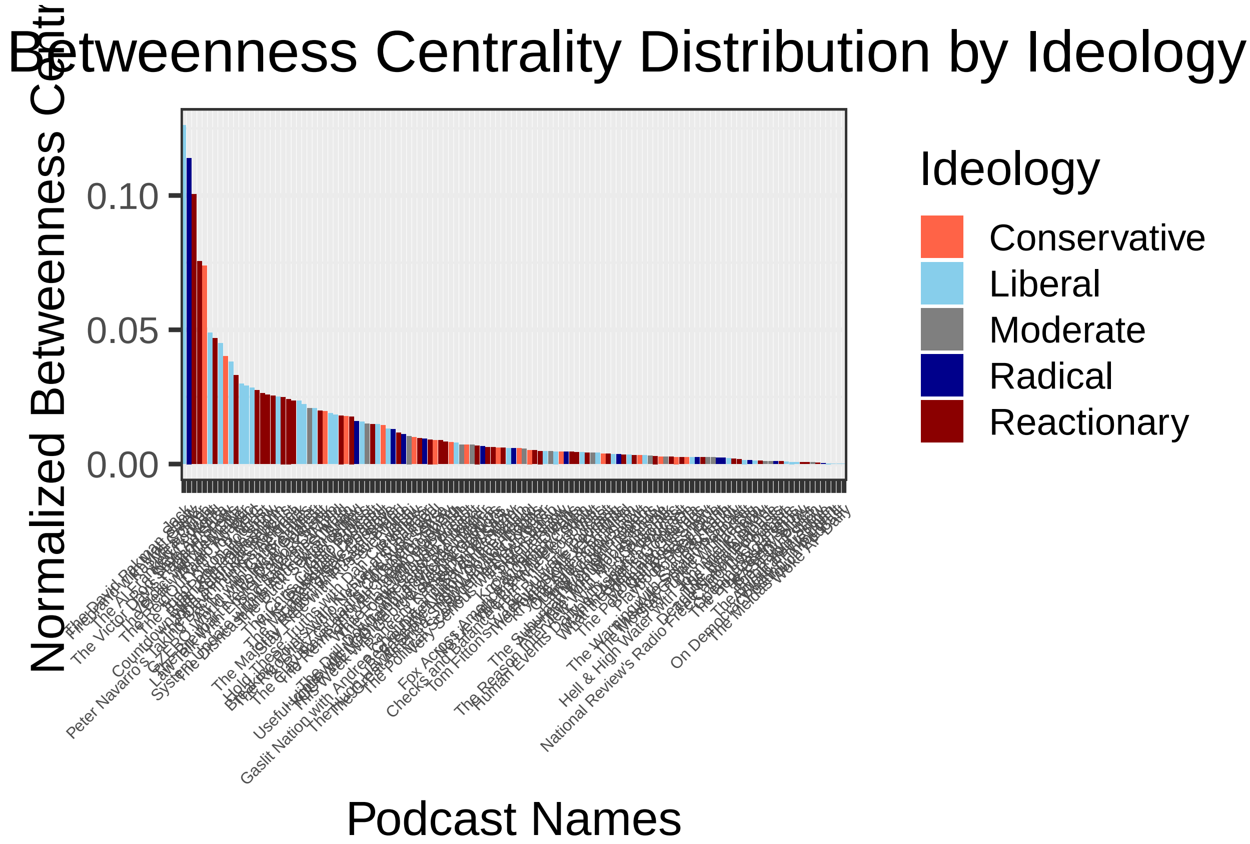
```
## [1] "The Median of this distribution is 8.00"
```

Betweenness Centrality Plot

```r
# Betweenness centrality

# Calculate betweenness centrality
btwn <- igraph::betweenness(
  g_full,
  v = V(g_full),
  directed = TRUE,
  weights = E(g_full)$weight,
  nobigint = TRUE,
  normalized = TRUE,
  cutoff = -1 # remove those with negative betweenness (these are anomalous)
)
```

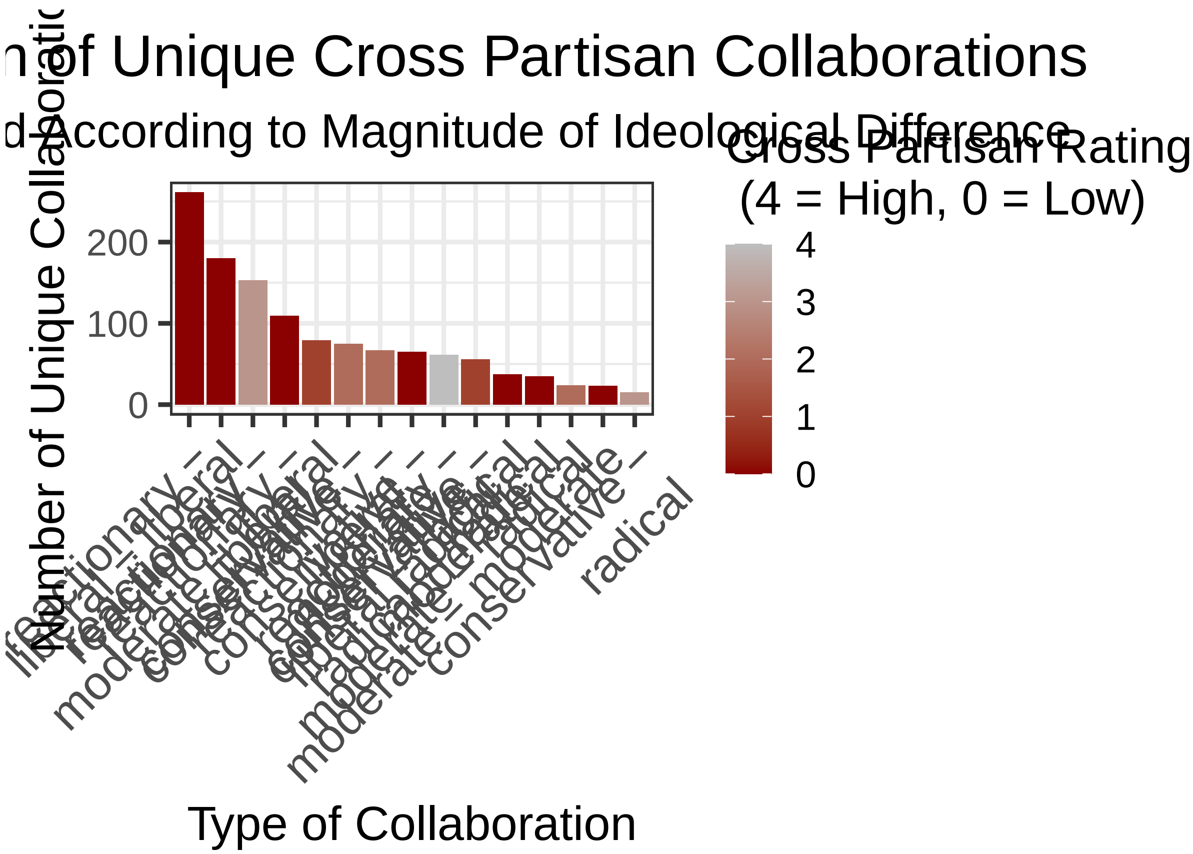
```
## Warning in igraph::betweenness(g_full, v = V(g_full), directed = TRUE, weights
## = E(g_full)$weight, : 'nobigint' is deprecated since igraph 1.3 and will be
## removed in igraph 1.4
```

```r
# Store betweenness centrality into a dataframe
btwn_df <- as.data.frame(btwn) %>%
  rownames_to_column() %>%
  rename(podcasts = rowname) %>%
  left_join(nodes_df, by = "podcasts") %>%
  mutate(ideology = as.factor(main_ideology)) %>%
  filter(btwn > 0.0001)

# Plot distribution
btwn_plot <- ggplot(btwn_df,
             aes(x = reorder(podcasts, -btwn),
                 y = btwn,
                 fill = main_ideology)) +
             geom_bar(stat = "identity")  +
             xlab("Podcast Names") +
             ylab("Normalized Betweenness Centrality") +
             labs(title = "Podcast Betweenness Centrality Distribution by Ideology") +
             scale_fill_manual(name = "Ideology",
                               values = Btwn_Colors,
                               labels = Btwn_Ideologies) +
             theme_bw(base_size = 18) +
             theme(plot.title = element_text(hjust = 0.5)) +
             theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 6))
btwn_plot
```



Distribution of Unique Collaborations

```
crosspart_uniq_plot <- ggplot(crosspart_df,
                              aes(x = reorder(crosspart, -count),
                                  y = count,
                                  fill = cross)) +
                      geom_bar(stat = "identity") +
                      labs(title = "Distribution of Unique Cross Partisan Collaborations",
                           subtitle = "Color Scaled According to Magnitude of Ideological Difference") +
                      xlab("Type of Collaboration") +
                      ylab("Number of Unique Collaborations") +
                      scale_fill_gradient(name = "Cross Partisan Rating \n (4 = High, 0 = Low)",
                                          low = "darkred", high = "grey")+
                      scale_x_discrete(labels = function(x) str_wrap(x, width = 20)) +
                      theme_bw(base_size = 18) +
                      theme(plot.title = element_text(hjust = 0.5)) +
                      theme(plot.subtitle = element_text(hjust = 0.5)) +
                      theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 18))
crosspart_uniq_plot
```

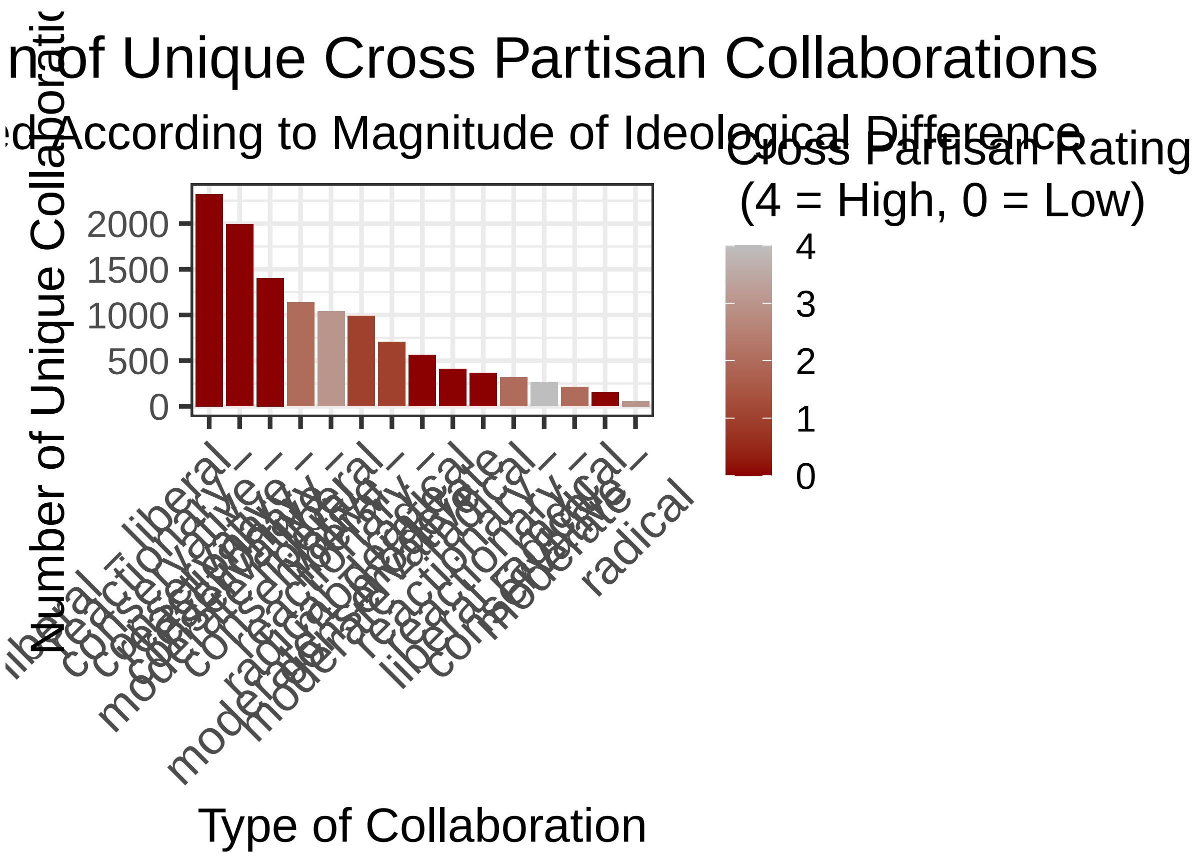


Distribution of All Collaborations

```
crosspart_totl_plot <- ggplot(crosspart_df,
                              aes(x = reorder(crosspart, -weight),
                                  y = weight,
                                  fill = cross)) +
                      geom_bar(stat = "identity") +
                      labs(title = "Distribution of Unique Cross Partisan Collaborations",
                           subtitle = "Color Scaled According to Magnitude of Ideological Difference") +
                      xlab("Type of Collaboration") +
                      ylab("Number of Unique Collaborations") +
                      scale_fill_gradient(name = "Cross Partisan Rating \n (4 = High, 0 = Low)",
                                          low = "darkred", high = "grey")+
                      scale_x_discrete(labels = function(x) str_wrap(x, width = 20)) +
                      theme_bw(base_size = 18) +
                      theme(plot.title = element_text(hjust = 0.5)) +
                      theme(plot.subtitle = element_text(hjust = 0.5)) +
                      theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 18))
crosspart_totl_plot
```

Table 1: Power Law Fit on In Degree Distribution

Fit	alpha	xmin	logLik	KS.stat	KS.p
Reactionary	3.120888	6	-15.611225	0.1533310	0.9965702
Conservative	3.370761	2	-7.771363	0.1362759	0.9983901
Moderate	1.801150	2	-21.221794	0.1851788	0.9466829
Liberal	2.391217	2	-24.718784	0.0633723	1.0000000
Radical	2.843976	2	-8.791595	0.1098867	0.9999960



Power Law Fits

```r
# Indegree distibution
indeg_plaw_df <- make_deg_plaw_df(Plaw_Ideologies, indeg_df)
# Create table visualization
indeg_plaw_df %>%
  rename(Fit = continuous) %>%
  mutate(Fit = Plaw_Ideologies) %>%
  kbl(caption = "Power Law Fit on In Degree Distribution") %>%
  kable_classic(full_width = F, html_font = "Garamond")
```

Table 2: Power Law Fit on Out Degree Distribution

Fit	alpha	xmin	logLik	KS.stat	KS.p
Reactionary	1.753123	1	-43.891745	0.0865225	0.9975299
Conservative	3.999316	3	-2.571572	0.0978381	1.0000000
Moderate	1.639565	2	-18.234130	0.2028436	0.9659160
Liberal	2.547867	3	-20.055532	0.1448716	0.9846805
Radical	3.456098	4	-5.857113	0.1140595	1.0000000

Table 3: Power Law Fit on Betweenness Centrality Distribution

Fit	alpha	xmin	logLik	KS.stat	KS.p
Betweenness	1.736006	0.0024938	339.5301	0.1080547	0.1846106

```
# Out degree Distribution
outdeg_plaw_df <- make_deg_plaw_df(Plaw_Ideologies, outdeg_df)
# Create table visualization
outdeg_plaw_df %>%
  rename(Fit = continuous) %>%
  mutate(Fit = Plaw_Ideologies) %>%
  kbl(caption = "Power Law Fit on Out Degree Distribution") %>%
  kable_classic(full_width = F, html_font = "Garamond")
```

```
# Betweenness centrality Power Law fit
as.data.frame(power.law.fit(btwn_df$btwn)) %>%
  rename(Fit = continuous) %>%
  mutate(Fit = c("Betweenness")) %>%
  kbl(caption = "Power Law Fit on Betweenness Centrality Distribution") %>%
  kable_classic(full_width = F, html_font = "Garamond")
```

```
# Unique Collaborations Power Law Fit
as.data.frame(power.law.fit(crosspart_df$count)) %>%
  rename(Fit = continuous) %>%
  mutate(Fit = c("Betweenness")) %>%
  kbl(caption = "Power Law Fit on Unique Cross-Partisan Collaborations") %>%
  kable_classic(full_width = F, html_font = "Garamond")
```

```
# Total Collaborations Power Law Fit
as.data.frame(power.law.fit(crosspart_df$weight)) %>%
  rename(Fit = continuous) %>%
  mutate(Fit = c("Betweenness")) %>%
  kbl(caption = "Power Law Fit on All Cross-Partisan Collaborations") %>%
  kable_classic(full_width = F, html_font = "Garamond")
```

Table 4: Power Law Fit on Unique Cross-Partisan Collaborations

Fit	alpha	xmin	logLik	KS.stat	KS.p
Betweenness	2.631312	56	-49.78813	0.1399152	0.9896177

Table 5: Power Law Fit on All Cross-Partisan Collaborations

Fit	alpha	xmin	logLik	KS.stat	KS.p
Betweenness	1.779169	211	-100.1882	0.1907516	0.7314954

ERGM Models

Data Preprocessing

```
# Create dataframe for edges in ergm.
# Exclude collaboraitons involving "NULL" ideologies (remove NULL values since they are too sparse for

# Map ideologies to podcasts involved in collaborations
ideology_df <- nodes_df %>%
  select(podcasts, main_ideology)

# First merge to get ideology of hosts
collab_ergm_df <- collab_df %>%
  merge(x = collab_df,
        y = ideology_df,
        by.x = "to",
        by.y = "podcasts") %>%
  rename(ideology_host = main_ideology)

# Second merge to get ideology of guests and filter out "NULL" ideologies
collab_ergm_df <- collab_ergm_df %>%
  merge(x = collab_ergm_df,
        y = ideology_df,
        by.x = "from",
        by.y = "podcasts") %>%
  rename(ideology_guest = main_ideology) %>%
  filter(! ideology_host %in% "NULL",
         ! ideology_guest %in% "NULL") %>%
  mutate(crosspart = str_c(ideology_guest, " - ", ideology_host)) %>%
  select(from, to, weight)



# Load in additional data on host bios
hostbio_df <- read.csv("podcast_hostattribs.csv") %>%
  rename(podcasts = Podcast,
         alternative_media_personality = other,
         political_personality = politician,
         professional_personality = political_professional,
         academic_personality = academic) %>%
  mutate(legacy_media_personality = tv_personality + print_personality + radio_personality) %>%
  select(2, 4, 8:12) %>%
  mutate(sum = rowSums(across(where(is.numeric)))) %>%
  mutate(across(2:7, ~ . / sum)) %>%
  mutate_at(2:7, ~replace(., is.nan(.), 0)) %>%
  select(1:7)
```

```r
# Subset nodes_df to exclude "NULL" Values
nodes_ergm_df <- nodes_df %>%
  left_join(hostbio_df, by = "podcasts") %>%
  filter(! main_ideology %in% "NULL") %>%
  mutate_at(12:17, ~ifelse(. > 0, 1, 0))

# create social network graph
g_ergm <- graph_from_data_frame(collab_ergm_df,
                                directed = TRUE,
                                vertices = nodes_ergm_df$podcasts) # Directed edges

# Set node attributes
g_ergm <- g_ergm %>%
  set_vertex_attr("ideology",
                  index = V(g_ergm),
                  value = nodes_ergm_df$id_code) %>%
  set_vertex_attr("distributor",
                  index = V(g_ergm),
                  value = nodes_ergm_df$distributor_code) %>%
  set_vertex_attr("bias_count",
                  index = V(g_ergm),
                  value = nodes_ergm_df$bias_std) %>%
  set_vertex_attr("bias_ratio",
                  index = V(g_ergm),
                  value = nodes_ergm_df$bias_ratio)  %>%
  set_vertex_attr("political_personality",
                  index = V(g_ergm),
                  value = nodes_ergm_df$political_personality)  %>%
  set_vertex_attr("academic_personality",
                  index = V(g_ergm),
                  value = nodes_ergm_df$academic_personality)  %>%
  set_vertex_attr("professional_personality",
                  index = V(g_ergm),
                  value = nodes_ergm_df$professional_personality)  %>%
  set_vertex_attr("alt_media_personality",
                  index = V(g_ergm),
                  value = nodes_ergm_df$alternative_media_personality)  %>%
  set_vertex_attr("religious_personality",
                  index = V(g_ergm),
                  value = nodes_ergm_df$religious_personality)  %>%
  set_vertex_attr("legacy_media_personality",
                  index = V(g_ergm),
                  value = nodes_ergm_df$legacy_media_personality)

# Network object for ERGM
network <- intergraph::asNetwork(g_ergm)
```

Model 1

Build Model

```r
mod_1 <- readRDS("mod_1.RData")
table <- htmlreg(list(mod_1),
        custom.coef.names = c("Number of Edges",
                              "Reactionary",
                              "Conservative",
                              "Moderate",
                              "Liberal",
                              "Radical",
                              "Difference in In Degree - Conservative",
                              "Difference in In Degree - Moderate",
                              "Difference in In Degree - Liberal",
                              "Difference in In Degree - Radical",
                              "Difference in Out Degree - Conservative",
                              "Difference in Out Degree - Moderate",
                              "Difference in Out Degree - Liberal",
                              "Difference in Out Degree - Radical",
                              "Ratio of Biased to Unbiased Statements",
                              "Host is a political figure",
                              "Host is a legacy media figure",
                              "Host is a alternative media figure",
                              "Host is a relgious leader",
                              "Host is an academic",
                              "Independent Podcast",
                              "Podcast owned by a Single Company",
                              "Podcast part of a Distribution Network",
                              "0 Guest Appearances",
                              "1 Guest Appearances",
                              "2 Guest Appearances",
                              "3 Guest Appearances",
                              "4 Guest Appearances",
                              "5 Guest Appearances",
                              "Presence of Mutual Connection"),
        bold = 0.5)
htmltools::HTML(table)
```

Model Results

```r
gof(mod_1)
```

Diagnostics

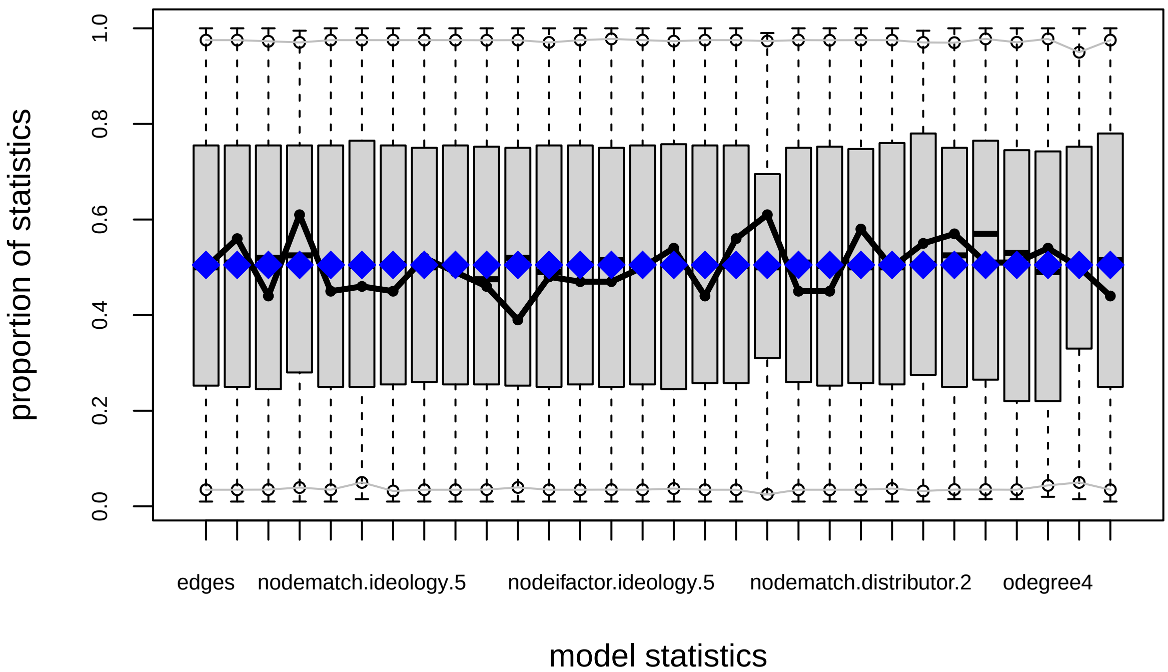
```
##
## Goodness-of-fit for in-degree
##
##             obs min  mean max MC p-value
## idegree0     47   1  6.20  12       0.00
## idegree1     30  10 17.75  31       0.02
## idegree2     28  12 26.38  44       0.82
```

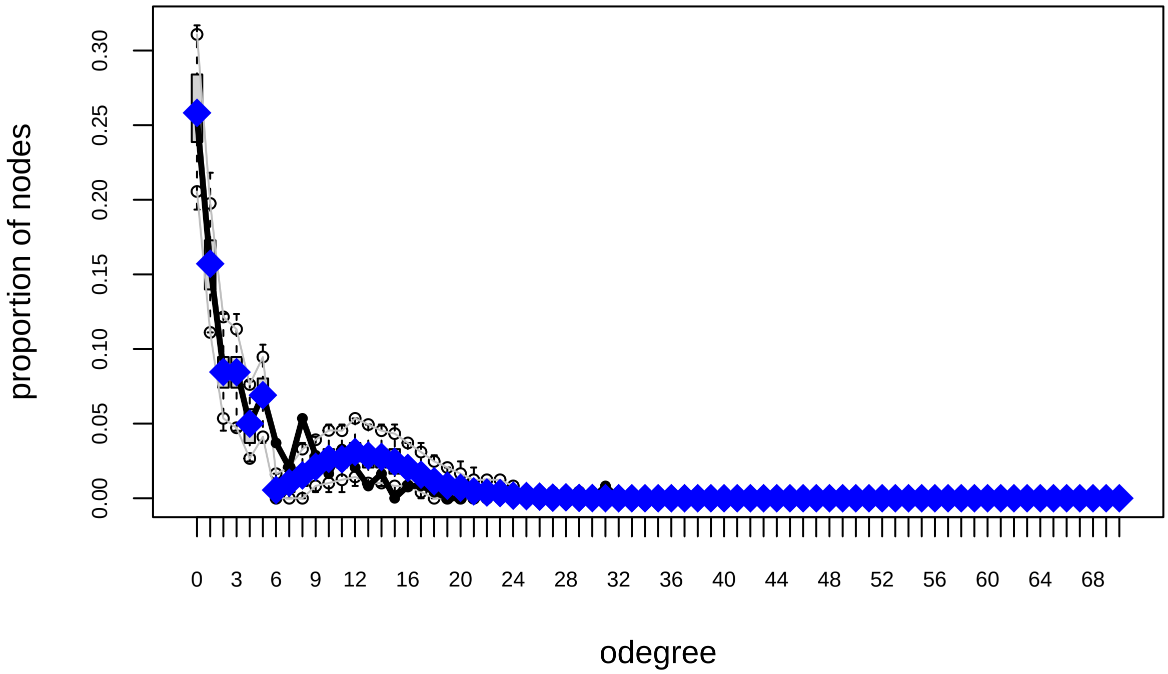
```
## idegree3   18  17 31.40  44       0.06
## idegree4   20  23 32.67  45       0.00
## idegree5   21  16 31.28  44       0.04
## idegree6    5  14 26.30  37       0.00
## idegree7    8  13 22.02  32       0.00
## idegree8   11   8 16.77  28       0.22
## idegree9    7   4 12.25  22       0.18
## idegree10  11   0  7.87  17       0.38
## idegree11   5   1  5.27  14       1.00
## idegree12   5   0  3.03  12       0.40
## idegree13   5   0  1.70  10       0.14
## idegree14   6   0  0.97   6       0.02
## idegree15   2   0  0.56   4       0.22
## idegree16   5   0  0.27   2       0.00
## idegree17   3   0  0.17   1       0.00
## idegree18   0   0  0.08   1       1.00
## idegree19   2   0  0.04   1       0.00
## idegree21   0   0  0.01   1       1.00
## idegree23   0   0  0.01   1       1.00
## idegree24   1   0  0.00   0       0.00
## idegree26   1   0  0.00   0       0.00
## idegree27   1   0  0.00   0       0.00
## idegree28   1   0  0.00   0       0.00
##
## Goodness-of-fit for out-degree
##
##           obs min  mean max MC p-value
## odegree0   62  46 61.79  81       1.00
## odegree1   38  23 38.41  55       1.00
## odegree2   21  12 20.67  29       1.00
## odegree3   21  13 21.78  30       0.90
## odegree4   12   6 12.19  20       1.00
## odegree5   17   8 16.64  29       1.00
## odegree6    9   0  1.30   5       0.00
## odegree7    5   0  2.50   7       0.20
## odegree8   13   0  3.58   9       0.00
## odegree9    7   1  4.99  11       0.48
## odegree10   4   1  5.94  13       0.54
## odegree11   8   3  6.76  12       0.76
## odegree12   5   2  7.29  17       0.54
## odegree13   2   1  7.01  14       0.06
## odegree14   4   2  6.83  12       0.34
## odegree15   0   1  5.95  11       0.00
## odegree16   2   1  5.18  14       0.20
## odegree17   2   1  3.66   9       0.48
## odegree18   1   0  2.56   7       0.56
## odegree19   0   0  2.24   7       0.20
## odegree20   0   0  1.46   7       0.44
## odegree21   1   0  1.21   4       1.00
## odegree22   2   0  1.03   4       0.60
## odegree23   0   0  0.59   3       1.00
## odegree24   0   0  0.40   3       1.00
## odegree25   1   0  0.39   3       0.60
## odegree26   0   0  0.22   1       1.00
```

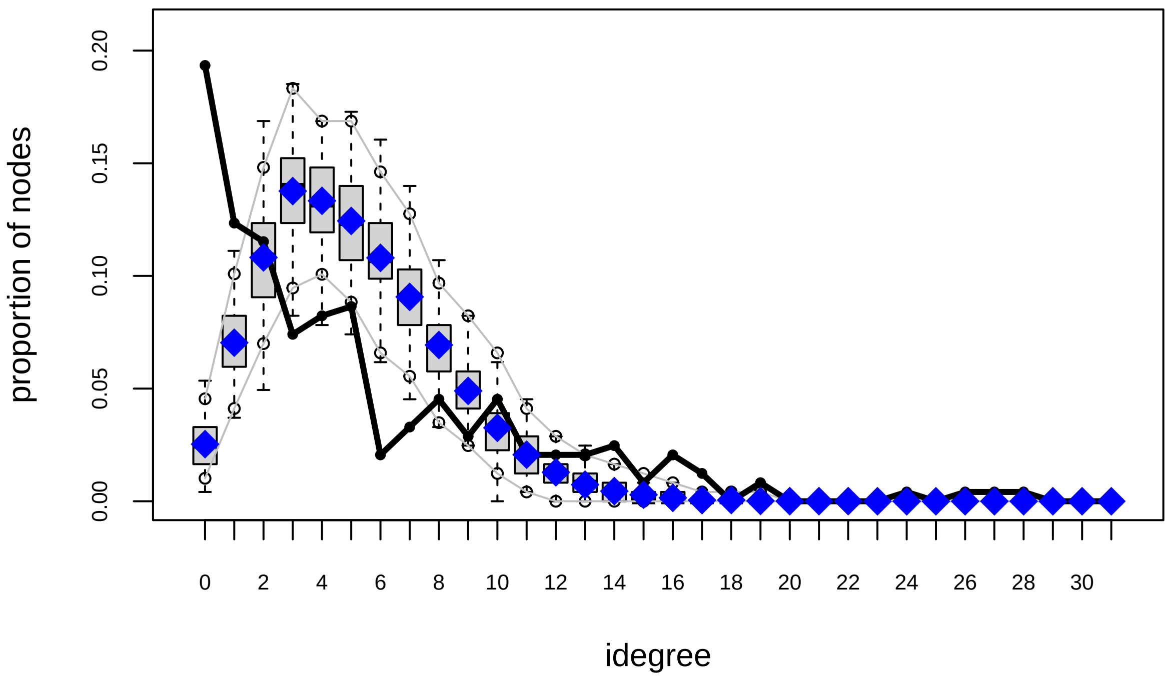
```
## odegree27   1   0  0.14   2       0.22
## odegree28   0   0  0.13   1       1.00
## odegree29   0   0  0.07   1       1.00
## odegree30   0   0  0.02   1       1.00
## odegree31   2   0  0.03   1       0.00
## odegree33   0   0  0.03   1       1.00
## odegree34   0   0  0.01   1       1.00
## odegree45   1   0  0.00   0       0.00
## odegree63   1   0  0.00   0       0.00
## odegree67   1   0  0.00   0       0.00
##
## Goodness-of-fit for edgewise shared partner
##
##       obs min   mean max MC p-value
## esp0  463 783 873.04 968       0.00
## esp1  339 199 282.58 382       0.16
## esp2  208  27  65.48 107       0.00
## esp3  111   2  11.91  29       0.00
## esp4   57   0   1.82   8       0.00
## esp5   30   0   0.23   3       0.00
## esp6    8   0   0.04   1       0.00
## esp7   14   0   0.00   0       0.00
## esp8    6   0   0.00   0       0.00
## esp10   2   0   0.00   0       0.00
## esp12   2   0   0.00   0       0.00
##
## Goodness-of-fit for minimum geodesic distance
##
##       obs   min     mean   max MC p-value
## 1    1240  1077  1235.10  1417       0.86
## 2    4316  4869  6164.10  7962       0.00
## 3    8643 11431 14063.00 17622       0.00
## 4    8555  9452 11678.57 13815       0.00
## 5    4416  3538  4835.92  6846       0.52
## 6    1423   586  1398.13  2755       0.80
## 7     279    42   357.66  1131       0.86
## 8      37     0    88.99   517       0.90
## 9       2     0    19.67   319       1.00
## 10      0     0     4.45   178       1.00
## 11      0     0     0.91    54       1.00
## 12      0     0     0.07     4       1.00
## 13      0     0     0.01     1       1.00
## Inf 29895 13864 18959.42 23450       0.00
##
## Goodness-of-fit for model statistics
##
##                                    obs       min      mean       max
## edges                        1240.0000 1077.0000 1235.1000 1417.0000
## nodematch.ideology.1          261.0000  196.0000  259.5200  316.0000
## nodematch.ideology.2           65.0000   33.0000   65.9100   96.0000
## nodematch.ideology.3           23.0000    7.0000   22.9400   41.0000
## nodematch.ideology.4          180.0000  119.0000  178.8900  282.0000
## nodematch.ideology.5           35.0000   18.0000   33.8200   74.0000
## nodeofactor.ideology.2        196.0000  133.0000  198.6800  279.0000
```

```
## nodeofactor.ideology.3            134.0000   77.0000  132.8700  185.0000
## nodeofactor.ideology.4            284.0000  190.0000  282.1800  421.0000
## nodeofactor.ideology.5             65.0000   38.0000   62.8600  150.0000
## nodeifactor.ideology.2            189.0000  149.0000  190.0400  243.0000
## nodeifactor.ideology.3            138.0000  106.0000  137.6100  179.0000
## nodeifactor.ideology.4            420.0000  349.0000  419.1800  514.0000
## nodeifactor.ideology.5            142.0000   99.0000  139.3500  195.0000
## nodecov.bias_ratio                701.5113  593.9929  695.8628  798.0268
## nodecov.political_personality     388.0000  327.0000  387.1800  450.0000
## nodecov.legacy_media_personality 1203.0000  993.0000 1200.3300 1445.0000
## nodecov.alt_media_personality     885.0000  682.0000  875.3700 1079.0000
## nodecov.religious_personality       4.0000    0.0000    3.7700   10.0000
## nodecov.academic_personality      136.0000   91.0000  137.0100  199.0000
## nodematch.distributor.1           220.0000  157.0000  218.5800  292.0000
## nodematch.distributor.2           124.0000   84.0000  125.2000  175.0000
## nodematch.distributor.3           132.0000   89.0000  129.5100  188.0000
## odegree0                           62.0000   46.0000   61.7900   81.0000
## odegree1                           38.0000   23.0000   38.4100   55.0000
## odegree2                           21.0000   12.0000   20.6700   29.0000
## odegree3                           21.0000   13.0000   21.7800   30.0000
## odegree4                           12.0000    6.0000   12.1900   20.0000
## odegree5                           17.0000    8.0000   16.6400   29.0000
## mutual                             88.0000   59.0000   86.7400  121.0000
##                                  MC p-value
## edges                                  0.86
## nodematch.ideology.1                   1.00
## nodematch.ideology.2                   1.00
## nodematch.ideology.3                   1.00
## nodematch.ideology.4                   0.86
## nodematch.ideology.5                   0.78
## nodeofactor.ideology.2                 0.98
## nodeofactor.ideology.3                 0.94
## nodeofactor.ideology.4                 0.88
## nodeofactor.ideology.5                 0.70
## nodeifactor.ideology.2                 1.00
## nodeifactor.ideology.3                 1.00
## nodeifactor.ideology.4                 0.84
## nodeifactor.ideology.5                 0.82
## nodecov.bias_ratio                     0.84
## nodecov.political_personality          0.92
## nodecov.legacy_media_personality       0.76
## nodecov.alt_media_personality          0.86
## nodecov.religious_personality          1.00
## nodecov.academic_personality           0.98
## nodematch.distributor.1                0.86
## nodematch.distributor.2                0.96
## nodematch.distributor.3                0.82
## odegree0                               1.00
## odegree1                               1.00
## odegree2                               1.00
## odegree3                               0.90
## odegree4                               1.00
## odegree5                               1.00
## mutual                                 0.76
```

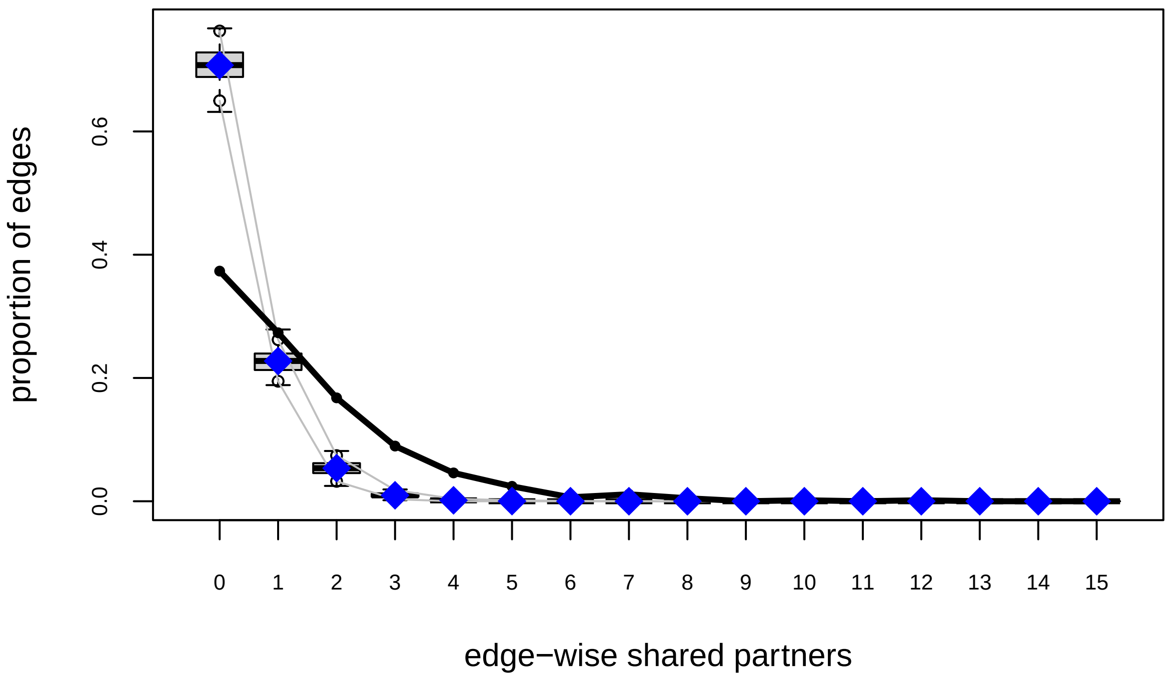
```
plot(gof(mod_1))
```



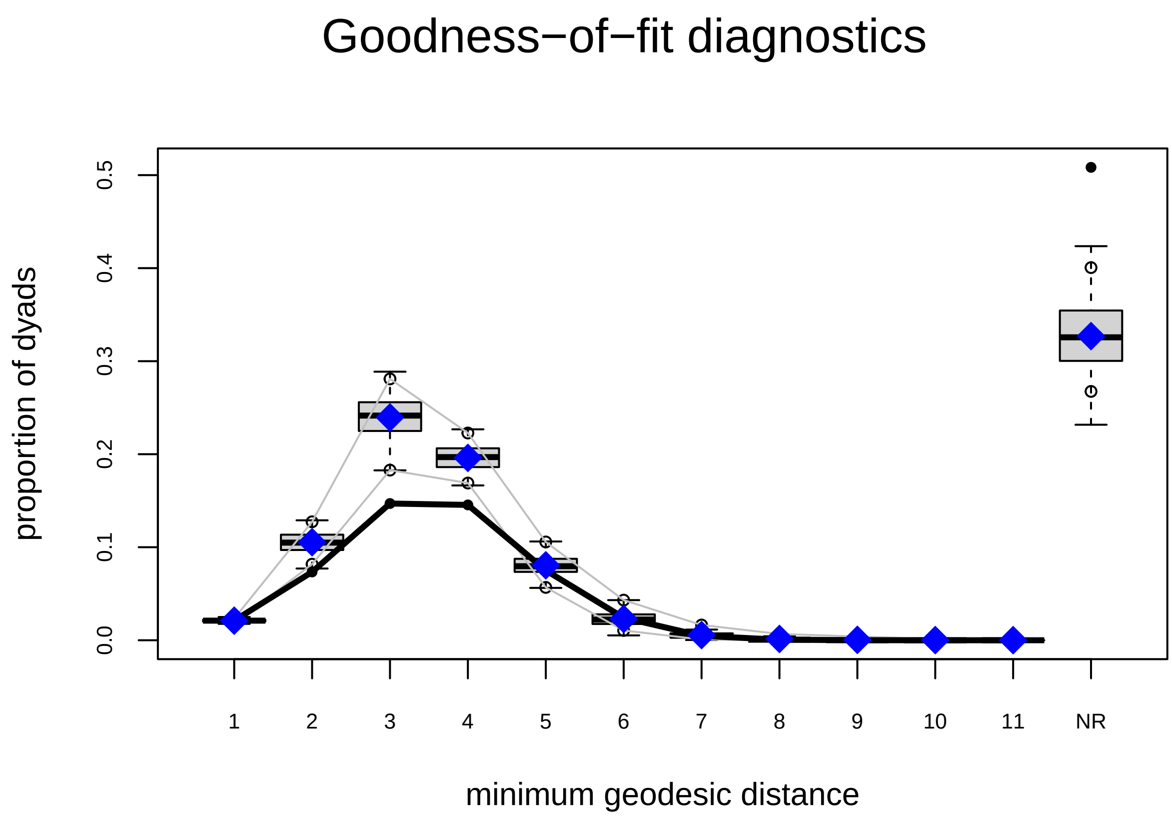

proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0.30
odegree
0 3 6 9 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68



proportion of nodes
0.00
0.05
0.10
0.15
0.20
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
idegree



proportion of edges
0.0
0.2
0.4
0.6
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
edge−wise shared partners



Model 2

Build model

```
mod_2 <- readRDS("mod_2.RData")
table <- htmlreg(list(mod_2),
       custom.coef.names = c("Number of Edges",
                             "Conservative -> Reactionary",
                             "Moderate -> Reactionary",
                             "Liberal -> Reactionary",
                             "Radical -> Reactionary",
                             "Reactionary -> Conservative",
                             "Conservative -> Conservative",
                             "Moderate -> Conservative",
                             "Liberal -> Conservative",
                             "Radical -> Conservative",
                             "Reactionary -> Moderate",
                             "Conservative -> Moderate",
                             "Moderate -> Moderate",
                             "Liberal -> Moderate",
                             "Radical -> Moderate",
                             "Reactionary -> Liberal",
```

```
                                  "Conservative -> Liberal",
                                  "Moderate -> Liberal",
                                  "Liberal -> Liberal",
                                  "Radical -> Liberal",
                                  "Reactionary -> Radical",
                                  "Conservative -> Radical",
                                  "Moderate -> Radical",
                                  "Liberal -> Radical",
                                  "Radical -> Radical",
                                  "Ratio of Biased to Unbiased Statements",
                                  "Presence of Reciprocal Collaboration"),
        bold = 0.05)
htmltools::HTML(table)
```

Model Results

```
gof(mod_2)
```

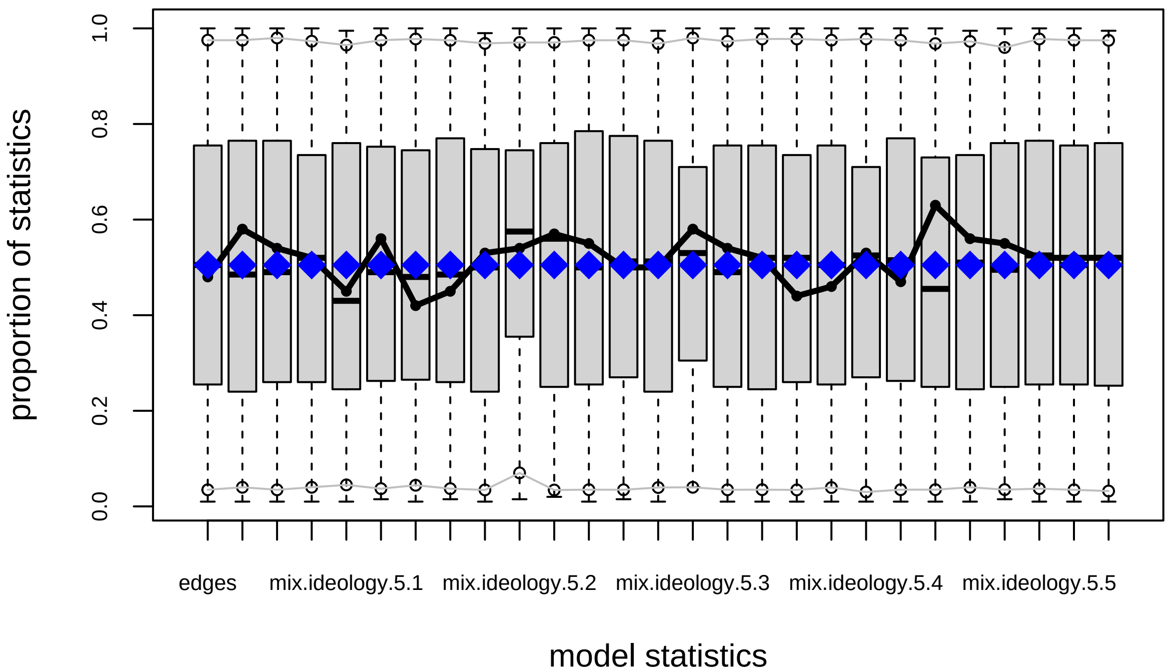
Diagnostics

```
## In term 'nodemix' in package 'ergm': Argument 'base' has been superseded by
## 'levels2', and it is recommended to use the latter.  Note that its
## interpretation may be different.

##
## Goodness-of-fit for in-degree
##
##            obs min  mean max MC p-value
## idegree0    47   0  5.20  13       0.00
## idegree1    30  11 16.85  25       0.00
## idegree2    28  16 26.68  40       0.82
## idegree3    18  19 31.82  46       0.00
## idegree4    20  21 34.32  47       0.00
## idegree5    21  18 31.33  48       0.04
## idegree6     5  15 26.60  39       0.00
## idegree7     8  10 22.15  32       0.00
## idegree8    11   7 16.21  25       0.28
## idegree9     7   6 11.94  21       0.28
## idegree10   11   2  8.03  14       0.38
## idegree11    5   0  5.15  12       1.00
## idegree12    5   0  2.87   8       0.30
## idegree13    5   0  1.88   5       0.06
## idegree14    6   0  0.94   4       0.00
## idegree15    2   0  0.55   3       0.28
## idegree16    5   0  0.28   3       0.00
## idegree17    3   0  0.11   1       0.00
## idegree18    0   0  0.06   1       1.00
## idegree19    2   0  0.01   1       0.00
## idegree20    0   0  0.02   1       1.00
## idegree24    1   0  0.00   0       0.00
```

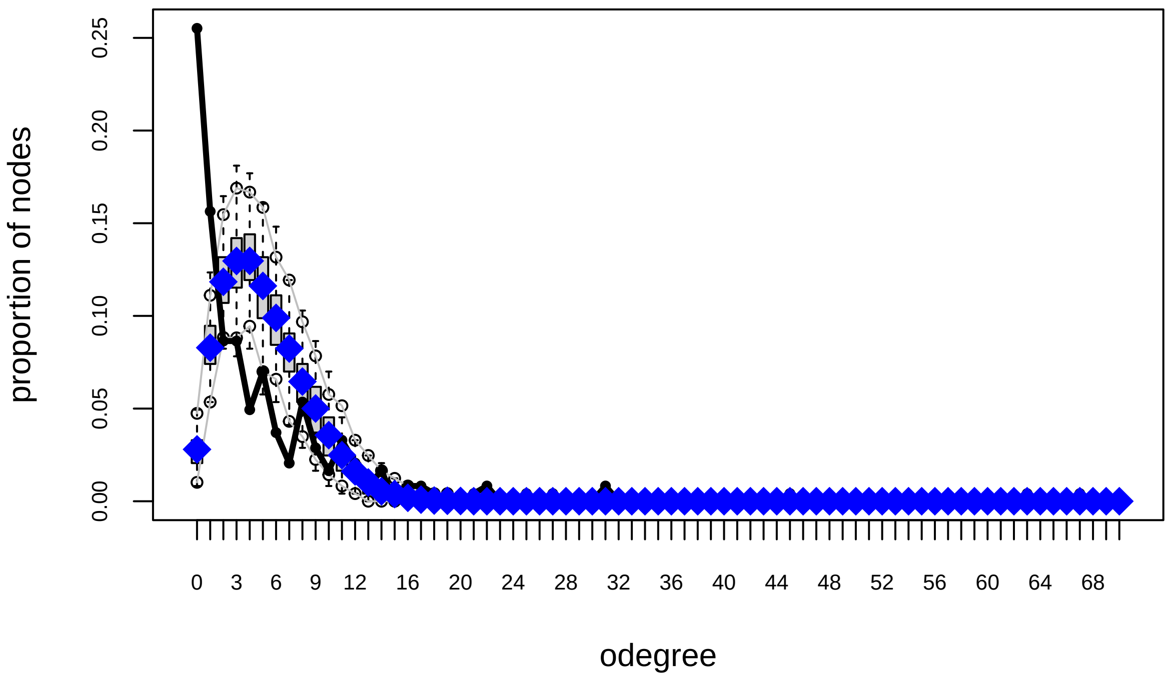
```
## idegree26   1   0  0.00   0       0.00
## idegree27   1   0  0.00   0       0.00
## idegree28   1   0  0.00   0       0.00
##
## Goodness-of-fit for out-degree
##
##           obs min  mean max MC p-value
## odegree0   62   1  7.09  14       0.00
## odegree1   38  10 19.26  28       0.00
## odegree2   21  17 28.18  38       0.20
## odegree3   21  18 31.83  42       0.02
## odegree4   12  21 31.28  46       0.00
## odegree5   17  17 28.92  43       0.02
## odegree6    9  14 23.78  35       0.00
## odegree7    5  10 20.64  34       0.00
## odegree8   13   7 16.49  32       0.50
## odegree9    7   7 12.53  21       0.12
## odegree10   4   4  8.51  15       0.10
## odegree11   8   1  5.64  10       0.46
## odegree12   5   0  3.86   9       0.72
## odegree13   2   0  2.06   8       1.00
## odegree14   4   0  1.39   5       0.08
## odegree15   0   0  0.80   3       0.90
## odegree16   2   0  0.43   3       0.18
## odegree17   2   0  0.23   2       0.02
## odegree18   1   0  0.04   1       0.08
## odegree19   0   0  0.04   1       1.00
## odegree21   1   0  0.00   0       0.00
## odegree22   2   0  0.00   0       0.00
## odegree25   1   0  0.00   0       0.00
## odegree27   1   0  0.00   0       0.00
## odegree31   2   0  0.00   0       0.00
## odegree45   1   0  0.00   0       0.00
## odegree63   1   0  0.00   0       0.00
## odegree67   1   0  0.00   0       0.00
##
## Goodness-of-fit for edgewise shared partner
##
##       obs min    mean  max MC p-value
## esp0  463 995 1044.41 1115          0
## esp1  339 113  172.35  231          0
## esp2  208   8   18.34   34          0
## esp3  111   0    1.42    6          0
## esp4   57   0    0.12    1          0
## esp5   30   0    0.01    1          0
## esp6    8   0    0.00    0          0
## esp7   14   0    0.00    0          0
## esp8    6   0    0.00    0          0
## esp10   2   0    0.00    0          0
## esp12   2   0    0.00    0          0
##
## Goodness-of-fit for minimum geodesic distance
##
##       obs   min     mean   max MC p-value
```

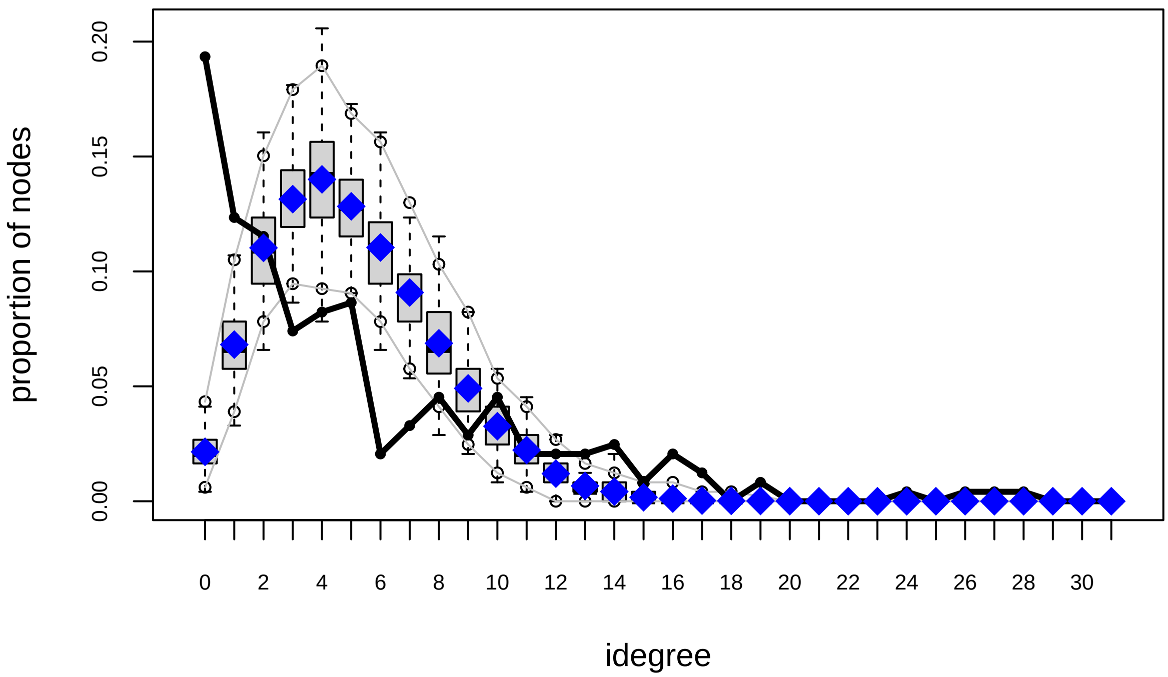
```
## 1    1240  1154  1236.65  1350       1.00
## 2    4316  5392  6010.29  7077       0.00
## 3    8643 15793 17711.74 20620       0.00
## 4    8555 18142 19738.69 21467       0.00
## 5    4416  6168  8526.66 10449       0.00
## 6    1423   900  2007.11  3307       0.30
## 7     279    63   340.93   871       0.76
## 8      37     0    50.67   333       0.84
## 9       2     0     6.86   112       1.00
## 10      0     0     0.65    16       1.00
## 11      0     0     0.03     2       1.00
## Inf 29895   967  3175.72  5695       0.00
##
## Goodness-of-fit for model statistics
##
##                          obs       min      mean       max MC p-value
## edges              1240.0000 1154.0000 1236.6500 1350.0000       1.00
## mix.ideology.2.1     42.0000   26.0000   41.8900   58.0000       0.98
## mix.ideology.3.1     26.0000   13.0000   26.3300   43.0000       1.00
## mix.ideology.4.1     16.0000    5.0000   15.6400   31.0000       1.00
## mix.ideology.5.1      6.0000    0.0000    6.3900   13.0000       1.00
## mix.ideology.1.2     67.0000   47.0000   67.2700   87.0000       0.92
## mix.ideology.2.2     65.0000   41.0000   64.8900   85.0000       1.00
## mix.ideology.3.2     27.0000   13.0000   26.0200   36.0000       0.94
## mix.ideology.4.2     26.0000   16.0000   25.9400   40.0000       1.00
## mix.ideology.5.2      4.0000    0.0000    3.6800   10.0000       1.00
## mix.ideology.1.3     41.0000   28.0000   41.2700   55.0000       0.96
## mix.ideology.2.3     29.0000   16.0000   29.0300   43.0000       1.00
## mix.ideology.3.3     23.0000   13.0000   23.5100   36.0000       1.00
## mix.ideology.4.3     41.0000   22.0000   40.6900   58.0000       1.00
## mix.ideology.5.3      4.0000    1.0000    4.1600   10.0000       1.00
## mix.ideology.1.4    137.0000  113.0000  136.1100  178.0000       0.92
## mix.ideology.2.4     49.0000   31.0000   49.3000   68.0000       0.98
## mix.ideology.3.4     38.0000   26.0000   38.5300   55.0000       0.96
## mix.ideology.4.4    180.0000  148.0000  179.2700  217.0000       1.00
## mix.ideology.5.4     16.0000    7.0000   16.3200   28.0000       1.00
## mix.ideology.1.5     55.0000   39.0000   54.8100   76.0000       1.00
## mix.ideology.2.5     11.0000    3.0000   10.5200   21.0000       1.00
## mix.ideology.3.5     20.0000    9.0000   20.3400   36.0000       1.00
## mix.ideology.4.5     21.0000    8.0000   20.8400   29.0000       1.00
## mix.ideology.5.5     35.0000   22.0000   34.4200   52.0000       1.00
## nodecov.bias_ratio  701.5113  645.3831  699.3484  772.5174       0.86
## mutual               88.0000   66.0000   85.8000  112.0000       0.74
```

```
plot(gof(mod_2))
```

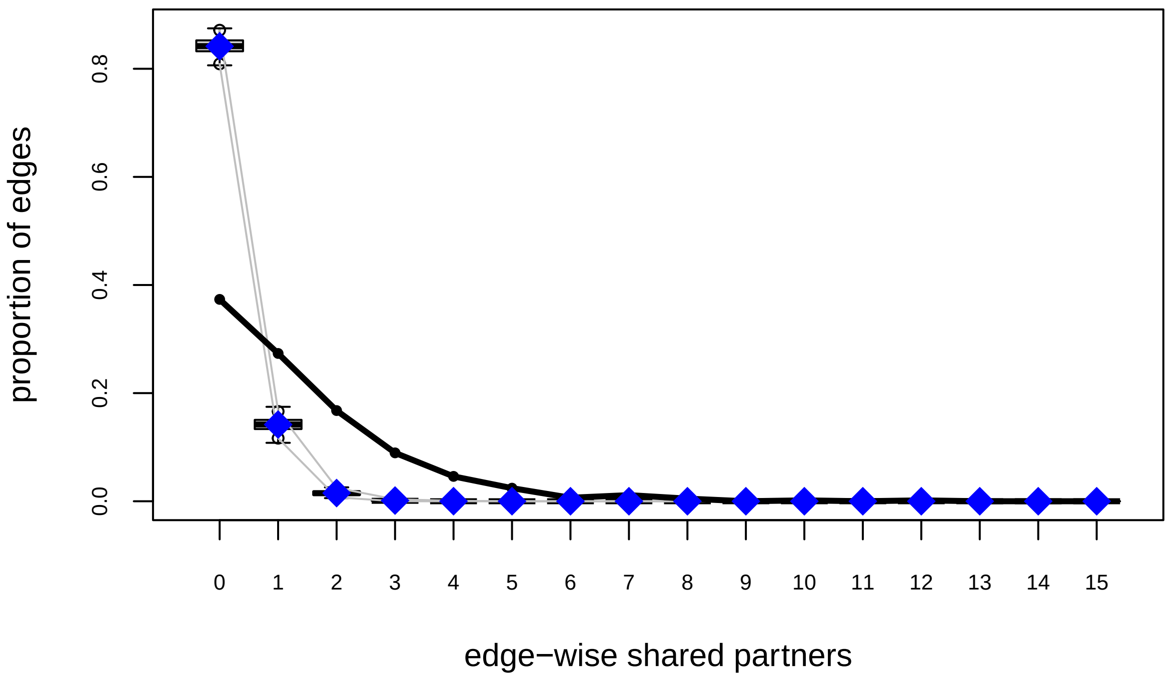


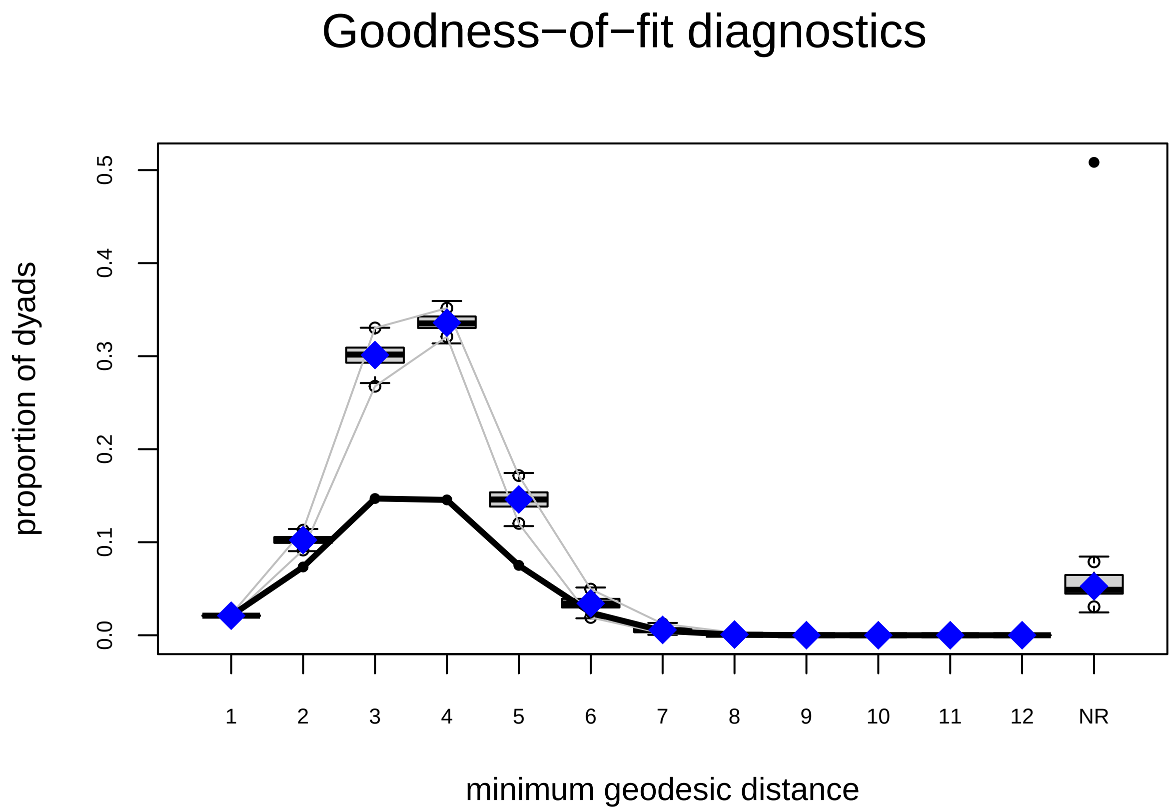
proportion of statistics
0.0
0.2
0.4
0.6
0.8
1.0
edges
mix.ideology.5.1
mix.ideology.5.2
mix.ideology.5.3
mix.ideology.5.4
mix.ideology.5.5
model statistics


proportion of nodes
0.00
0.05
0.10
0.15
0.20
0.25
0 3 6 9 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68
odegree



0.00
0.05
0.10
0.15
0.20
proportion of nodes
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
idegree


proportion of edges
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
edge−wise shared partners


Goodness−of−fit diagnostics
proportion of dyads
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
7
8
9
10
11
12
NR
minimum geodesic distance