Replication Document

Francesco Bailo

19 October 2018

Contents

Paper details **2**

Packages **2**

Functions **2**

Network and diffusion models **3**
Network setup functions . . . 3
Network initialisation . . . 6
Figure 1 . . . 7
Figure 2 . . . 7
Network simulation . . . 8
Figure 3 . . . 11

Meetup **13**
Meetup datasets . . . 13
Data collection range . . . 13
Number of records . . . 13
Users' joining date . . . 14
Number of members indicating a Facebook account for which a joining date is available . . . 14

Facebook **14**

Facebook friendship graph of meetup members **15**
Create and save anonymised version . . . 17

Recruitment network (based on at least 90-day Facebook relationship) **17**
Figure 4 . . . 20
Figure 5 . . . 22

Data analysis **24**
Spatial polygons of Italian regions . . . 24
ISTAT Survey . . . 24
Figure 6 . . . 30
Figure 12 . . . 31
Table 2 . . . 32
Logistic regression prediction: MALE . . . 34
Logistic regression prediction: FEMALE . . . 35
Diffusion . . . 36
Create geographic grid . . . 36
Number of Meetup members by grid cells . . . 37
Figure 7 . . . 39
Figure 11 . . . 42
Analysis of recruitment and Post-recruitment networks . . . 45
Figure 8 . . . 51
Exponential Random Graph Models . . . 53

Create control variable `pre_active_buff_20km` 53
Create control variable `gender` 55
ERGM 57
Logistic regression predicting at least one recruiting tie 59
Table 4 64

Paper details

Packages

```
library(igraph)
library(truncnorm)
library(openssl)
library(data.table)
library(parallel)
library(knitr)
library(ggplot2)
library(scales)
library(DBI)
library(RSQLite)
library(rjson)
library(geosphere)
library(rgdal)
library(sp)
library(sf)
library(rgeos)
library(dplyr)
library(gridExtra)
library(scales)
library(egg)
library(srvyr)
library(stargazer)
library(reshape2)
library(intergraph)
library(statnet)
library(ergm)
library(rgeos)
library(stringr)
library(texreg)
library(RColorBrewer)
library(knitr)
```

Functions

```
sqliteGetTable <- function(database, table) {
  require(DBI)
  require(RSQLite)
  con <- dbConnect(RSQLite::SQLite(), dbname = database)
  query <- dbSendQuery(con, paste("SELECT * FROM ", table, ";", sep=""))
```

```
  result <- fetch(query, n = -1)
  dbClearResult(query)
  dbDisconnect(con)
  return(result)
}

addDegreeToVertices <- function (g) {
  require(igraph)
  try(if(!(is.igraph(g))) stop("Not an igraph graph"))
  if (is_directed(g)) {
    V(g)$indegree <- igraph::degree(g, V(g), mode = 'in')
    V(g)$outdegree <- igraph::degree(g, V(g), mode = 'out')
  } else {
    try(if('degree' %in% igraph::list.vertex.attributes(g))
      stop("Vertex attribure degree already exists"))
    V(g)$degree <- igraph::degree(g, V(g))
  }
  return(g)
}

vertexAttributesAsDataFrame <- function(g) {
  try(if(!(is.igraph(g))) stop("Not an igraph graph"))
  vertex_attributes <- igraph::list.vertex.attributes(g)
  df <- data.frame(matrix(NA, nrow = vcount(g), ncol = 0))
  for (a in vertex_attributes) {
    df <- cbind(df, data.frame(igraph::get.vertex.attribute(g, a)))
  }
  names(df) <- igraph::list.vertex.attributes(g)
  return(df)
}
```

Network and diffusion models

Network setup functions

```
read_chunk('replication_code_01_simulation_functions.R')
```

```
truncatedProp <- function(n, mean, sd = .25) {
  require(truncnorm)
  return(rtruncnorm(n=n, a=0, b=1, mean=mean, sd = sd))
}

randomSocialNetwork <- function(n_communities, g_size, min_comm_size,
                                within_p, between_p) {

  if(n_communities>20) stop("Up to 20 communities")

  library(igraph)
  library(openssl)

  g_list <- list()
```

```
  default_comm_size <- g_size / n_communities
  comm_sizes <- rep(default_comm_size, n_communities)

  # Randomise community size
  isOdd <- function(x) {x %% 2 != 0}
  for (i in 1:n_communities) {
    if (isOdd(i)) {
      if (i == n_communities) next
      this_change <-
        sample(-abs(default_comm_size-min_comm_size):abs(default_comm_size-min_comm_size),
               1)
      comm_sizes[i] <- comm_sizes[i] + this_change
      last_change <- -this_change
    } else {
      comm_sizes[i] <- comm_sizes[i] + last_change
    }
  }

  for (i in 1:n_communities) {
    # print(paste0(i, " ", comm_sizes[i]))
    g_list[[i]] <- random.graph.game(comm_sizes[i], within_p)
    V(g_list[[i]])$community <- LETTERS[i]
    V(g_list[[i]])$name <- paste0(LETTERS[i], 1:vcount(g_list[[i]]))
  }

  g <- do.call(disjoint_union, g_list)
  v_communities <- V(g)$community
  g_mat <- as_adj(g)
  from_is <- c(1,cumsum(comm_sizes)+1)
  from_is <- from_is[1:n_communities]
  to_is <- cumsum(comm_sizes)
  is <- mapply(seq, from_is, to_is, by = 1)
  for (i in 1:n_communities) {
    g_mat[is[[i]], unlist(is[which(1:n_communities != i)])] <-
      rbinom(n=length(as.numeric(g_mat[is[[i]],
                                       unlist(is[which(1:n_communities != i)])])),
             size=1, prob=between_p)
  }
  g <- graph.adjacency(g_mat, mode = 'undirected')
  V(g)$community <- v_communities
  return(g)
}

activationProb <- function(political_distress,
                           internal_drive, relational_drive, universal_drive,
                           neighborhood, i) {
  p <-
    1/2 * (political_distress * internal_drive) +
    1/2 * (political_distress * (relational_drive ^ (1/neighborhood)))
  return(p)
}

addSCNet <- function(g, prob = .3, mean = 1, sd = .25) {
```

```
  nodes <- V(g)$name[V(g)$internal_drive == 1]
  comb_nodes <- combn(nodes, 2)
  comb_nodes <- comb_nodes[,as.logical(rbinom(ncol(comb_nodes), size = 1, prob))]
  g <- g + edges(as.character(comb_nodes))
  g <- simplify(g)
  edge_attr(g, 'weight', get.edge.ids(g, as.character(comb_nodes))) <-
    trunctedProp(ncol(comb_nodes), mean=1, sd = .1)
  return(g)
}

getWeightedNeigh <- function(neigh, g) {
  len <- length(neigh)
  if (len == 1) return(0)
  ego <- neigh$name[1]
  neighbors <- neigh$name[2:len]
  active <- neigh$active[2:len]
  ids <- get.edge.ids(g, as.vector(rbind(rep(ego, len-1), neighbors)))
  weight <- edge_attr(g, name = 'weight', index = ids)
  return(weighted.mean(active, weight))
}

updateNetwork <- function(g, i, stay_active = TRUE, reset_each = NULL) {
  neigh_list <- neighborhood(g, order = 1)
  neigh <- unlist(lapply(neigh_list, getWeightedNeigh, g))
  neigh[is.na(neigh)] <- 0
  prob <- activationProb(V(g)$political_distress,
                         V(g)$internal_drive,
                         V(g)$relational_drive,
                         V(g)$universal_drive,
                         neigh, i)
  active <- as.logical(rbinom(vcount(g), size = 1, p = prob))
  if (stay_active == TRUE) {
    active[V(g)$active == TRUE] <- TRUE
  } else {
    active[V(g)$active == TRUE & V(g)$past < reset_each] <- TRUE
  }
  past <- ifelse(active == FALSE, 0, V(g)$past + active)
  return(list('active' = active, 'past' = past))
}

doNetSim <- function(i, g, path = NULL, print = FALSE) {
  if(i > 1) {
    res <- updateNetwork(g, i, stay_active = T)
    V(g)$active <- res[['active']]
    V(g)$past <- res[['past']]
  }
  if (print == TRUE) {
    jpeg(file = paste0(path, sprintf('%03d.jpg', i)),
         width = 2500, height = 2500)
    par(mar=c(0,0,4,0))
    plot(g, vertex.label = NA, vertex.size = 4, layout = g_lo,
         vertex.color = ifelse(V(g)$active, "black", "white"))
    title(main = paste0("Time: ", i,
```

```
                        "     Activation: ",
                        round(sum(V(g)$active) / vcount(g) *100, 2),
                        "%"),
          cex.main = 6)
    dev.off()
  }
  return(list(graph = g, activation = sum(V(g)$active) / vcount(g)))
}
```

Network initialisation

```
set.seed(28100)

g <- randomSocialNetwork(n_communities = 20, g_size = 490, min_comm_size = 15,
                         within_p = .115, between_p = 0.00005)
g_lo <- layout.fruchterman.reingold(g)

# Node attributes
V(g)$political_distress <- tructedProp(vcount(g), mean=.5, sd = .2)
V(g)$internal_drive_wtout_internet <- rbinom(vcount(g), 1, p = 0.1)
V(g)$internal_drive_wt_internet <- rbinom(vcount(g), 1, p = 0.5)
V(g)$relational_drive <- tructedProp(vcount(g), mean=.2, sd = .2)

# Edge attributes
E(g)$weight_sc <- tructedProp(ecount(g), mean=.6, sd = .15)
E(g)$weight_nosc <- tructedProp(ecount(g), mean=.2, sd = .15)

# Penalise cross-community edges
es <- as_edgelist(g)
es[,1] <- gsub("\\d", "", es[,1])
es[,2] <- gsub("\\d", "", es[,2])
is_same_comm <- es[,1] == es[,2]

E(g)$weight_nosc[!is_same_comm] <-
  tructedProp(sum(!is_same_comm), mean=0, sd = .15)

# Initialise
V(g)$active <- FALSE

# Create high social capital network
V(g)$internal_drive <- V(g)$internal_drive_wtout_internet
g_wt_sc_net <- addSCNet(g)

# Number of nodes
vcount(g)
```

```
## [1] 490
```

```
# Numebr of Edges
ecount(g)
```

```
## [1] 684
```

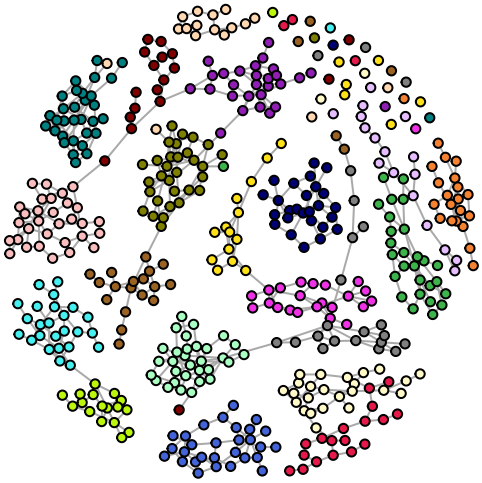
```r
# Percentage of edges running within a predefined social knot
el <- as_edgelist(g)
sum(gsub("\\d","", el[,1]) == gsub("\\d","", el[,2])) / nrow(el)
```

```
## [1] 0.9780702
```

```r
save(g, g_lo, file = "replication_file_01_simulated_net.RData")
```

Figure 1

```r
# Model network
colpal <- c('#e6194b', '#3cb44b', '#ffe119', '#4363d8', '#f58231',
            '#911eb4', '#46f0f0', '#f032e6', '#bcf60c', '#fabebe',
            '#008080', '#e6beff', '#9a6324', '#fffac8', '#800000',
            '#aaffc3', '#808000', '#ffd8b1', '#000075', '#808080',
            '#ffffff', '#000000')

plot(g, vertex.label = NA, vertex.size = 4, layout = g_lo,
     vertex.color = colpal[as.numeric(as.factor(V(g)$community))])
```

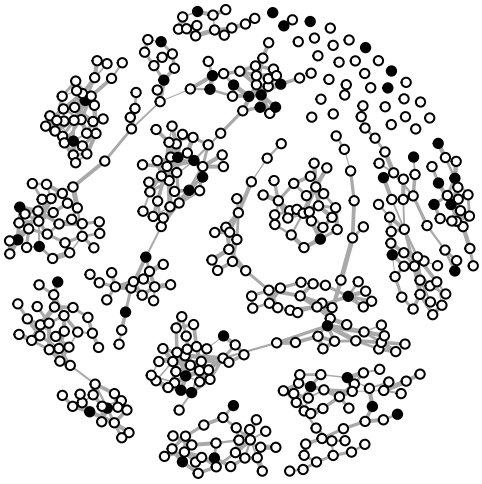


```r
pdf(file = "figure/network-model.pdf")
par(mar=c(0,0,0,0))
plot(g, vertex.label = NA, vertex.size = 4, layout = g_lo,
     vertex.color = colpal[as.numeric(as.factor(V(g)$community))])
dev.off()
```

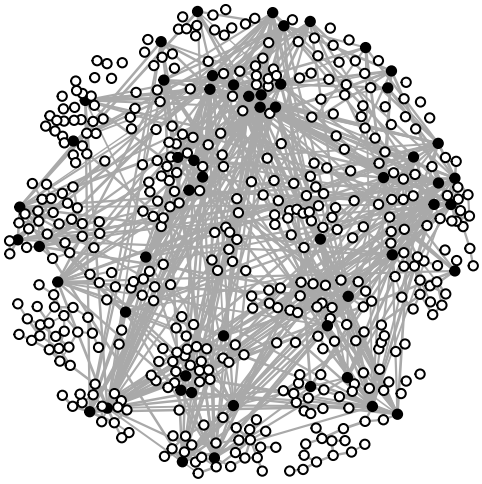
```
## pdf
##   2
```

Figure 2

```r
plot(g, vertex.label = NA, vertex.size = 4, layout = g_lo,
     vertex.color = ifelse(V(g)$internal_drive == 1, "black", "white"),
     edge.width = sqrt(E(g)$weight_nosc)*3.5)
```



```
plot(g_wt_sc_net, vertex.label = NA, vertex.size = 4, layout = g_lo,
     vertex.color = ifelse(V(g)$internal_drive == 1, "black", "white"),
     edge.width = sqrt(E(g)$weight_sc)*1.5)
```



```
pdf(file = "figure/network-model-low-high-capital.pdf",
    width  = 16, height = 7)
par(mar=c(0,0,0,0), mfrow=c(1,2))
plot(g, vertex.label = NA, vertex.size = 4, layout = g_lo,
     vertex.color = ifelse(V(g)$internal_drive == 1, "black", "white"),
     edge.width = sqrt(E(g)$weight_nosc)*3.5)
plot(g_wt_sc_net, vertex.label = NA, vertex.size = 4, layout = g_lo,
     vertex.color = ifelse(V(g)$internal_drive == 1, "black", "white"),
     edge.width = sqrt(E(g)$weight_sc)*1.5)
dev.off()
```

```
## pdf
##   2
```

Network simulation

```
read_chunk('replication_code_02_simulate_diffusion.R')
```

```r
library(parallel)
library(igraph)
library(data.table)

load("replication_file_01_simulated_net.RData")

no_cores <- detectCores() - 1
cl <- makeCluster(no_cores)

fun4Parallel <- function(j, sc_bin, internet_bin, save_plots = FALSE) {

  library(igraph)
  library(data.table)

  g <- randomSocialNetwork(n_communities = 20, g_size = 490, min_comm_size = 15,
                           within_p = .115, between_p = 0.00005)
  g_lo <- layout.fruchterman.reingold(g)

  # Node attributes
  V(g)$political_distress <- trunctedProp(vcount(g), mean=.5, sd = .2)
  V(g)$internal_drive_wtout_internet <- rbinom(vcount(g), 1, p = 0.1)
  V(g)$internal_drive_wt_internet <- rbinom(vcount(g), 1, p = 0.5)
  V(g)$relational_drive <- trunctedProp(vcount(g), mean=.2, sd = .2)

  # Edge attributes
  E(g)$weight_sc <- trunctedProp(ecount(g), mean=.6, sd = .15)
  E(g)$weight_nosc <- trunctedProp(ecount(g), mean=.2, sd = .15)

  # Penalise cross-community edges
  es <- as_edgelist(g)
  es[,1] <- gsub("\\d", "", es[,1])
  es[,2] <- gsub("\\d", "", es[,2])
  is_same_comm <- es[,1] == es[,2]

  E(g)$weight_nosc[!is_same_comm] <-
    trunctedProp(sum(!is_same_comm), mean=0, sd = .15)

  # Initialise
  V(g)$active <- FALSE

  V(g)$active <- FALSE

  # The network structure (edges) is determined by `internal_drive_wtout_internet` also
  # for the Internet scenario.
  V(g)$internal_drive <- V(g)$internal_drive_wtout_internet

  if (sc_bin == TRUE) {
    E(g)$weight <- E(g)$weight_sc
    g <- addSCNet(g)
  } else {
    E(g)$weight <- E(g)$weight_nosc
  }
```

```
  if (internet_bin == TRUE) {
    V(g)$internal_drive <- V(g)$internal_drive_wt_internet
  } else {
    V(g)$internal_drive <- V(g)$internal_drive_wtout_internet
  }

  what <- ifelse(sc_bin, "wt sc", "wtout sc")
  path <- ifelse(sc_bin, "sim_wt_sc", "sim_wtout_sc")
  if (internet_bin == TRUE) {
    what <- paste0(what, " wt internet")
    path <- paste0(path, "_wt_internet/")
  } else {
    what <- paste0(what, " wtout internet")
    path <- paste0(path, "_wtout_internet/")
  }
  this_dt <- data.table()
  for (i in 1:1000) {
    res <- doNetSim(i, g)
    g <- res$graph
    if (save_plots == TRUE) {
      jpeg(file = paste0(path, sprintf('%03d.jpg', i)),
           width = 2500, height = 2500)
      par(mar=c(0,0,4,0))
      plot(g, vertex.label = NA, vertex.size = 4, layout = g_lo,
           vertex.color = ifelse(V(g)$active, "black", "white"))
      title(main = paste0("Time: ", i,
                          "     Activation: ",
                          round(sum(V(g)$active) / vcount(g) *100, 2),
                          "%"),
            cex.main = 6)
      dev.off()
    }
    this_dt <- rbind(this_dt,
                     data.table(activation = res$activation,
                                t = i,
                                j,
                                what = what))
  }
  return(this_dt)
}
X <- lsf.str()
clusterExport(cl=cl, varlist=c(as.vector(X)), envir=environment())

print('active_nodes_wt_sc_wtout_internet')
active_nodes_wt_sc_wtout_internet <-
  rbindlist(parLapply(cl, 1:100, fun4Parallel, sc_bin = T,
                      internet_bin = F, save_plots = FALSE))
# ffmpeg -framerate 10 -i %03d.jpg -c:v libx264 -profile:v high -crf 20 -pix_fmt yuv420p
# sim_g_wt_sc_wtout_internet.mp4

print('active_nodes_wtout_sc_wtout_internet')
active_nodes_wtout_sc_wtout_internet <-
  rbindlist(parLapply(cl, 1:100, fun4Parallel, sc_bin = F,
```

```
                      internet_bin = F, save_plots = FALSE))
# ffmpeg -framerate 10 -i %03d.jpg -c:v libx264 -profile:v high -crf 20 -pix_fmt yuv420p
# sim_g_wtout_sc_wtout_internet.mp4

print('active_nodes_wt_sc_wt_internet')
active_nodes_wt_sc_wt_internet <-
  rbindlist(parLapply(cl, 1:100, fun4Parallel, sc_bin = T,
                      internet_bin = T, save_plots = FALSE))

print('active_nodes_wtout_sc_wt_internet')
active_nodes_wtout_sc_wt_internet <-
  rbindlist(parLapply(cl, 1:100, fun4Parallel, sc_bin = F,
                      internet_bin = T, save_plots = FALSE))

stopCluster(cl)

save(active_nodes_wt_sc_wtout_internet, active_nodes_wtout_sc_wtout_internet,
     active_nodes_wt_sc_wt_internet, active_nodes_wtout_sc_wt_internet,
     file = 'replication_file_02_active_nodes.RData')
```

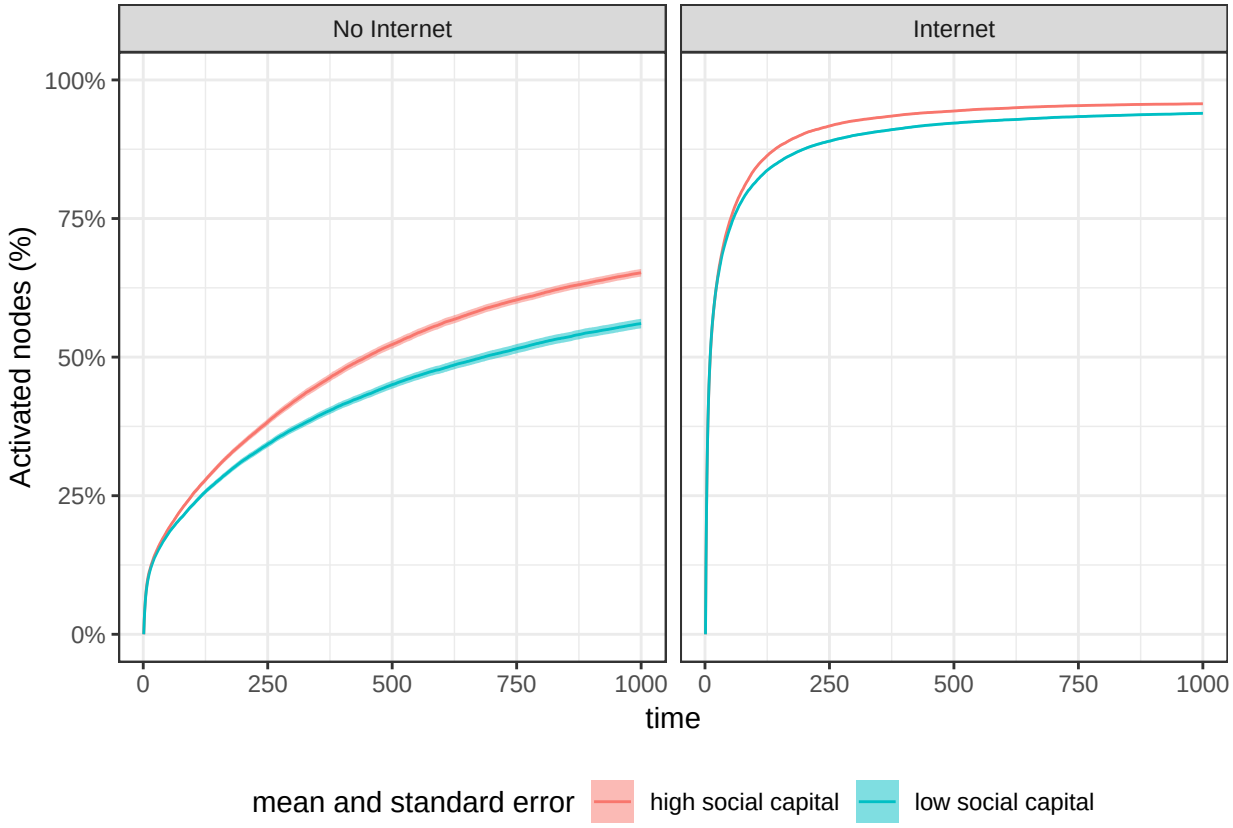
Simulations runs in parallel without any `set.seed()`. To run the simulation execute:

```
R -e 'setwd("/path/to/directory"); source("replication_code_01_simulation_functions.R"); source("replica
```

Figure 3

```
load("replication_file_02_active_nodes.RData")

active_nodes_df <-
  rbind(active_nodes_wt_sc_wtout_internet, active_nodes_wtout_sc_wt_internet,
        active_nodes_wtout_sc_wtout_internet, active_nodes_wt_sc_wt_internet)
active_nodes_df$group <- paste0(active_nodes_df$j, "-", active_nodes_df$what)

active_nodes_df <-
  active_nodes_df %>%
  dplyr::group_by(t, what) %>%
  dplyr::summarize(mean = mean(activation),
                   se = sqrt(var(activation)/length(activation)))

active_nodes_df$internet <-
  grepl("wt internet", active_nodes_df$what)
active_nodes_df$internet <-
  ifelse(active_nodes_df$internet, "Internet", "No Internet")
active_nodes_df$internet <-
  factor(active_nodes_df$internet, levels = c("No Internet", "Internet"))
active_nodes_df$label <-
  gsub(" (wt|wtout) internet", "", active_nodes_df$what)
active_nodes_df$label <-
  gsub("wt sc", "high social capital", active_nodes_df$label)
active_nodes_df$label <-
  gsub("wtout sc", "low social capital", active_nodes_df$label)
```

```
active_nodes_df$`mean and standard error` <- active_nodes_df$label

ggplot(active_nodes_df, aes(x=t, colour = `mean and standard error`)) +
  geom_line(aes(y=mean)) +
  geom_ribbon(aes(ymin = mean - se, ymax = mean + se, fill = `mean and standard error`),
              colour = NA, alpha = 0.5) +
  facet_wrap(~internet) +
  scale_y_continuous(label = percent, limits = c(0,1)) + theme_bw() +
  labs(x='time', y="Activated nodes (%)") + theme(legend.position = 'bottom')
```




```
pdf(file = "figure/network-model-diffusion.pdf",
    width  = 8, height = 5)
ggplot(active_nodes_df, aes(x=t, colour = `mean and standard error`)) +
  geom_line(aes(y=mean)) +
  geom_ribbon(aes(ymin = mean - se, ymax = mean + se, fill = `mean and standard error`),
              colour = NA, alpha = 0.5) +
  facet_wrap(~internet) +
  scale_y_continuous(label = percent, limits = c(0,1)) + theme_bw() +
  labs(x='time', y="Activated nodes (%)") + theme(legend.position = 'bottom') +
  theme(text = element_text(family = "Palatino"))
dev.off()
```

```
## pdf
##   2
```

Meetup

Meetup datasets

The following chunks are not replicable if you don't have access to `meetup_aug14.sqlite` and `meetup_mar15.sqlite`, which are password protected since they contain users' personal information.

The 2015 dataset was collected to correct the joining date of the 2014 dataset (`meetup_members_2014$joined`,), which was not correctly parse from the API response.

```
meetup_members_2014 <- sqliteGetTable('meetup_aug14.sqlite', 'member')
meetup_groups_2014 <- sqliteGetTable('meetup_aug14.sqlite', '[group]')
meetup_events_2014 <- sqliteGetTable('meetup_aug14.sqlite', 'event')
meetup_members_2015 <- sqliteGetTable('meetup_mar15.sqlite', 'member')
meetup_groups_2015 <- sqliteGetTable('meetup_mar15.sqlite', '[group]')
meetup_events_2015 <- sqliteGetTable('meetup_mar15.sqlite', 'event')
```

Data collection range

```
min(meetup_members_2014$timestamp)
```

```
## [1] "2014-08-01 09:56:21"
```

```
max(meetup_members_2014$timestamp)
```

```
## [1] "2014-08-02 09:52:33"
```

```
min(meetup_members_2015$timestamp)
```

```
## [1] "2015-03-02 08:17:01"
```

```
max(meetup_members_2015$timestamp)
```

```
## [1] "2015-03-02 09:15:48"
```

Number of records

```
length(unique(c(meetup_members_2015$member_id, meetup_members_2014$member_id)))
```

```
## [1] 104630
```

```
nrow(meetup_members_2014)
```

```
## [1] 97808
```

```
nrow(meetup_groups_2014)
```

```
## [1] 1883
```

```
nrow(meetup_events_2014)
```

```
## [1] 115335
```

```
nrow(meetup_members_2015)
```

```
## [1] 92486
```

```
nrow(meetup_groups_2015)
```

```
## [1] 1701
```

```
nrow(meetup_events_2015)
```

```
## [1] 128694
```

Users' joining date

```
meetup_members_2014$joined <-
  meetup_members_2015$joined[match(meetup_members_2014$member_id,
                                   meetup_members_2015$member_id)]

sum(!is.na(meetup_members_2014$joined))
```

```
## [1] 85664
```

```
min(meetup_members_2014$joined, na.rm = T)
```

```
## [1] "2005-07-11 15:07:47"
```

```
max(meetup_members_2014$joined, na.rm = T)
```

```
## [1] "2015-03-01 19:05:53"
```

Number of members indicating a Facebook account for which a joining date is available

```
sum(!is.na(meetup_members_2014$facebook) & !is.na(meetup_members_2014$joined))
```

```
## [1] 13350
```

Facebook

The following chunks are not fully replicable if you don't have access to ", which are password protected since they contain users' personal information.

To collect information about existing Facebook relations among Meetup users, I proceeded as following:

1. I scraped public information for each of the 15033 Facebook accounts indicated by Meetup members as of August 2014 with the Python script `replication_code_03_scrape_facebook.py`;
2. I scraped public information available on the Facebook "Friends" page of each of the Facebook users identified in the previous step with with the Python script `replication_code_04_scrape_facebook.py`;
3. I scraped public information available on the Facebook "Friendship" page corresponding to each Facebook relationship identified in the previous step with the Python script `replication_code_05_scrape_facebook.py`.

```
fb_user_timeline <-
  sqliteGetTable('facebook_user_timeline.sqlite', 'fbUsersTimeline')
fb_friendship_edgelist <-
  sqliteGetTable('facebook_friendship.sqlite', 'fb_friendship_edgelist')
```

The `fb_user_timeline` dataset contains the Facebook joining date of NA Facebook users. The `fb_friendship_edgelist` dataset contains the creation date of 165452 Facebook relationships.

The data collection process from Facebook can be summarised as following:

1. 15033 is the number of Meetup users indicating a Facebook account in August 2014;
2. 14381 is the number of Facebook accounts found publicly accessible;
3. 5387 is the number of Facebook accounts with a publicly accessible "Friends" page;
4. 9018 is the number of Facebook accounts indicated by Meetup users for which at least one Facebook friendship with another Meetup user is mapped (note: 3631 Meetup/Facebook users with no publicly accessible Facebook "Friends" page appeared in their friends' "Friends" pages).

Facebook friendship graph of meetup members

```
users <- fb_user_timeline
edgelist <- fb_friendship_edgelist

# Finding when user created their Facebook account (exclude all dates before 2007)
whenJoinedFB <- function(json) {
  if (length(json)==1 & json == 'null' | json == 'NA') return(NA)
  # print(json)
  # Tests
  # json1 = '[2013]'
  # json2 = '[2013, 2012, 1974]'
  # json3 = '[2013, 2012, 2009, 1943]'
  # json4 = '[2014, 1943]'
  require(rjson)
  fbActivity <- fromJSON(json)
  joined <- min(fbActivity[fbActivity>=2007])
  while(TRUE) {
    # Check whether there's continuity between supposed joining date and activity
    # (Possible the event in the timeline doesn't correspond to date of joining Facebook)
    if (joined == max(fbActivity)) {
      return(joined)
    } else if (((joined + 1) %in% fbActivity) & (joined %in% fbActivity))  {
      return(joined)
    } else {
      joined = joined + 1
    }
  }
  return(joined)
}

# Manual replace of missing data for
# V(g)[is.na(V(g)$joined)]
missing_data <- read.csv("replication_file_03_manually_coded.csv", header = FALSE)
users$joined[users$fb_id %in% missing_data$V1] <- missing_data$V2
users$joined_fb <- sapply(users$active_years, whenJoinedFB)

# Checks
unique_id <- unique(c(edgelist$from, edgelist$to))
fb_id <- users$fb_id
sum(unique_id %in% fb_id)

# Create graph
g <-
  graph.data.frame(edgelist[,2:8], directed=FALSE, vertices=users[fb_id %in% unique_id,])
```

```r
vcount(g)
ecount(g)

# Remove edges with error status (generated when requesting the friendship page)
g <- g - E(g)[E(g)$status == 2]
vcount(g)
ecount(g)

# Remove edges among users who are not friends
g <- g - E(g)[E(g)$friends == 0]
vcount(g)
ecount(g)

# There are 20 multiedges
g <- igraph::simplify(g, edge.attr.comb="first")
g_fship <- g


# Add Meetup joining date (note: collected only in 2015)
meetup_members_tmp <- meetup_members_2015
meetup_members_tmp$facebook <- NULL
meetup_members_tmp <-
  merge(meetup_members_tmp,
        meetup_members_2014[,c("member_id","facebook")],
        all.x = T, all.y = F)

V(g)$joined_mu <-
  meetup_members_tmp$joined[match(V(g)$name, meetup_members_tmp$facebook)]
V(g)$member_id <-
  meetup_members_tmp$member_id[match(V(g)$name, meetup_members_tmp$facebook)]

V(g)$lon <-
  meetup_members_2014$lon[match(V(g)$member_id, meetup_members_2014$member_id)]
V(g)$lat <-
  meetup_members_2014$lat[match(V(g)$member_id, meetup_members_2014$member_id)]

# Nodes with no joining date details
sum(is.na(V(g)$joined_mu))

## Removing them
g <- g - V(g)[is.na(V(g)$joined_mu)]
```

```r
# Counting potential recruiting ties
social_recruitment <- function(vindex, g, days) {
  # print(vindex)
  if(is.na(V(g)$joined_fb[vindex])) {
    return(NA)
  } else if(as.Date(V(g)$joined_mu[vindex]) <
            as.Date(paste0(V(g)$joined_fb[vindex], "-01-01"),
                    format="%Y-%m-%d", origin='1970-01-01')) {
    return(NA)
  } else {
    n <-
      sum(as.Date(paste0("01 ", E(g)[from(vindex)]$since), format=" %d %B %Y") + days
```

```
          < as.Date(V(g)$joined_mu[vindex]))
    # n <- vcount(neigh)
    return(n)
  }
}

# Computationally intense
V(g)$recruting_friends_90 <- sapply(1:vcount(g), social_recruitment, g, 90)

save(g, file="replication_file_04_friendship_graph.RData")
```

Create and save anonymised version

```
load("replication_file_04_friendship_graph.RData")

g_anonymised <- g
g_anonymised <-
  delete_vertex_attr(g_anonymised, "name")
g_anonymised <-
  delete_vertex_attr(g_anonymised, "member_id")
g_anonymised <-
  delete_vertex_attr(g_anonymised, "lon")
g_anonymised <-
  delete_vertex_attr(g_anonymised, "lat")
g_anonymised <-
  delete_edge_attr(g_anonymised, "grew_up")
g_anonymised <-
  delete_edge_attr(g_anonymised, "living")


save(g_anonymised, file="replication_file_05_friendship_graph_anonymised.RData")
```

Recruitment network (based on at least 90-day Facebook relationship)

```
load("replication_file_04_friendship_graph.RData")

friendship_edgelist <- as.data.frame(get.edgelist(g))
friendship_edgelist$friends <- E(g)$friends
friendship_edgelist$since <- E(g)$since
friendship_edgelist$grew_up <- E(g)$grew_up
friendship_edgelist$living <- E(g)$living

friendship_edgelist$joined_mu_A <-
  V(g)$joined_mu[match(friendship_edgelist$V1, V(g)$name)]
friendship_edgelist$joined_mu_B <-
  V(g)$joined_mu[match(friendship_edgelist$V2, V(g)$name)]
friendship_edgelist$A_joined_first <-
  friendship_edgelist$joined_mu_A < friendship_edgelist$joined_mu_B
```

```
returnV1 <- function(a, b, bool) {if (bool==TRUE) {return(a)} else {return(b)}}
returnV2 <- function(a, b, bool) {if (bool==TRUE) {return(b)} else{return(a)}}

recruitment_edgelist <- data.frame(V1=rep(NA, nrow(friendship_edgelist)),
                                   V2=rep(NA, nrow(friendship_edgelist)),
                                   recruiting=rep(NA, nrow(friendship_edgelist)),
                                   when=rep(NA, nrow(friendship_edgelist)))

recruitment_edgelist$V1[friendship_edgelist$A_joined_first==TRUE] <-
  as.character(friendship_edgelist$V1[friendship_edgelist$A_joined_first==TRUE])
recruitment_edgelist$V1[friendship_edgelist$A_joined_first==FALSE] <-
  as.character(friendship_edgelist$V2[friendship_edgelist$A_joined_first==FALSE])

recruitment_edgelist$V2[friendship_edgelist$A_joined_first==TRUE] <-
  as.character(friendship_edgelist$V2[friendship_edgelist$A_joined_first==TRUE])
recruitment_edgelist$V2[friendship_edgelist$A_joined_first==FALSE] <-
  as.character(friendship_edgelist$V1[friendship_edgelist$A_joined_first==FALSE])

recruitment_edgelist$when[friendship_edgelist$A_joined_first==TRUE] <-
  friendship_edgelist$joined_mu_B[friendship_edgelist$A_joined_first==TRUE]
recruitment_edgelist$when[friendship_edgelist$A_joined_first==FALSE] <-
  friendship_edgelist$joined_mu_A[friendship_edgelist$A_joined_first==FALSE]

recruitment_edgelist$recruiting <-
  (as.Date(paste0("01 ",
                  friendship_edgelist$since), format=" %d %B %Y") + 90) <
  as.Date(recruitment_edgelist$when)

recruitment_edgelist <- subset(recruitment_edgelist, recruiting==TRUE)

members <- meetup_members_2014

# Geo
recruitment_edgelist$lonA <-
  members$lon[match(recruitment_edgelist$V1, members$facebook)]
recruitment_edgelist$latA <-
  members$lat[match(recruitment_edgelist$V1, members$facebook)]
recruitment_edgelist$lonB <-
  members$lon[match(recruitment_edgelist$V2, members$facebook)]
recruitment_edgelist$latB <-
  members$lat[match(recruitment_edgelist$V2, members$facebook)]

require(geosphere)
recruitment_edgelist$distance <-
  distVincentySphere(
    cbind(recruitment_edgelist$lonA, recruitment_edgelist$latA),
    cbind(recruitment_edgelist$lonB, recruitment_edgelist$latB))

g_recruitment <-
  graph.data.frame(recruitment_edgelist[,c("V1","V2","when","distance")],
                   directed=TRUE,
                   vertices=unique(data.frame(name = c(recruitment_edgelist$V1,
```

```
                                                    recruitment_edgelist$V2),
                                          lon = c(recruitment_edgelist$lonA,
                                                  recruitment_edgelist$lonB),
                                          lat = c(recruitment_edgelist$latA,
                                                  recruitment_edgelist$latB))
                  ))

V(g_recruitment)$joined_mu <- V(g)$joined_mu[match(V(g_recruitment)$name, V(g)$name)]
V(g_recruitment)$joined_fb <- V(g)$joined_fb[match(V(g_recruitment)$name, V(g)$name)]

# Adding vertices removed from the recruitment network (because of no recruiting tie)
missing_vertices <- vertexAttributesAsDataFrame(g)
missing_vertices <- missing_vertices[!missing_vertices$name %in% V(g_recruitment)$name,]
# THIS MUST BE TRUE: nrow(missing_vertices) + vcount(g_recruitment) == vcount(g)
g_recruitment <-
  g_recruitment %>%
  add_vertices(nrow(missing_vertices),
               attr = list(name = as.character(missing_vertices$name),
                           lon = missing_vertices$lon,
                           lat = missing_vertices$lat,
                           joined_mu = as.character(missing_vertices$joined_mu),
                           joined_fb = as.character(missing_vertices$joined_fb)))

# Nodes
vcount(g_recruitment)
```

```
## [1] 8132
```

```
# Nodes with at least one recruiting ties
sum(igraph::degree(g_recruitment, mode="in") > 0)
```

```
## [1] 3804
```

```
sum(igraph::degree(g_recruitment, mode="in") > 0) / vcount(g_recruitment)
```

```
## [1] 0.4677816
```

```
# Nodes that joined Meetup.com after joining Facebook.com
sum(as.Date(V(g_recruitment)$joined_mu) >=
      as.Date(paste0(V(g_recruitment)$joined_fb, "-01-01"),
              format="%Y-%m-%d", origin='1970-01-01'), na.rm=TRUE)
```

```
## [1] 7874
```

```
sum(as.Date(V(g_recruitment)$joined_mu) >=
      as.Date(paste0(V(g_recruitment)$joined_fb, "-01-01"),
              format="%Y-%m-%d", origin='1970-01-01'), na.rm=TRUE) /
  vcount(g_recruitment)
```

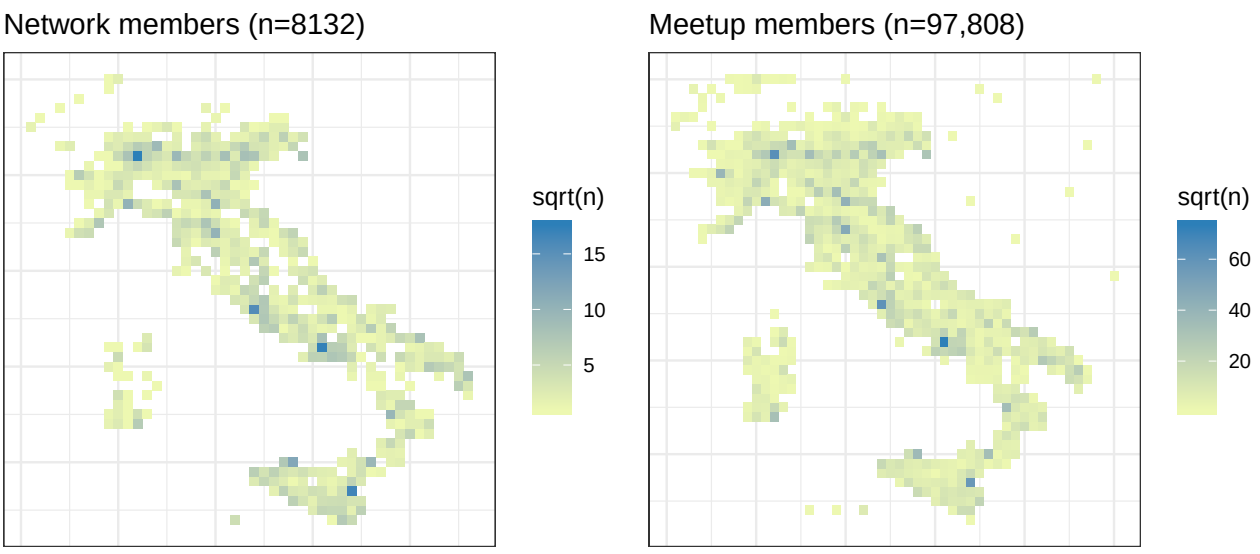
```
## [1] 0.9682735
```

```
save(g_recruitment, file = 'replication_file_06_recruitment_graph.RData')
```

Figure 4

```
load("replication_file_06_recruitment_graph.RData")

italy_box_y <- c(36, 47.5)
italy_box_x <- c(6.1, 19.5)

italy_seq_x <- seq(italy_box_x[1], italy_box_x[2], length.out = 51)
italy_seq_y <- seq(italy_box_y[1], italy_box_y[2], length.out = 51)

which_g <-
  V(g)$lon >= italy_box_x[1] & V(g)$lon <= italy_box_x[2] &
  V(g)$lat >= italy_box_y[1] & V(g)$lat <= italy_box_y[2]
lon_g <- V(g)$lon[which_g]
lat_g <- V(g)$lat[which_g]

which_mu <-
  meetup_members_2014$lon >=
  italy_box_x[1] & meetup_members_2014$lon <= italy_box_x[2] &
  meetup_members_2014$lat >=
  italy_box_y[1] & meetup_members_2014$lat <= italy_box_y[2]
lon_mu <- meetup_members_2014$lon[which_mu]
lat_mu <- meetup_members_2014$lat[which_mu]

lon_mu <-
  cut(lon_mu, breaks = italy_seq_x, labels = italy_seq_x[1:50])
lat_mu <-
  cut(lat_mu, breaks = italy_seq_y, labels = italy_seq_y[1:50])

lon_g <-
  cut(lon_g, breaks = italy_seq_x, labels = italy_seq_x[1:50])
lat_g <-
  cut(lat_g, breaks = italy_seq_y, labels = italy_seq_y[1:50])

g_lonlat_df <-
  data.frame(lon_g, lat_g) %>%
  group_by(lon_g, lat_g) %>%
  summarize(n = n())
mu_lonlat_df <-
  data.frame(lon_mu, lat_mu) %>%
  group_by(lon_mu, lat_mu) %>%
  summarize(n = n())

g_mu_lonlat_df <-
  merge(g_lonlat_df, mu_lonlat_df,
        by.x = c('lon_g','lat_g'),
        by.y = c('lon_mu','lat_mu'))

g_mu_lonlat_df$n_perc.x <-
  g_mu_lonlat_df$n.x / sum(g_mu_lonlat_df$n.x)
g_mu_lonlat_df$n_perc.y <-
  g_mu_lonlat_df$n.y / sum(g_mu_lonlat_df$n.y)
```

```
g_mu_lonlat_df$diff <- with(g_mu_lonlat_df, n_perc.x - n_perc.y)

grid.arrange(
  ggplot(g_lonlat_df,
         aes(x = as.numeric(lon_g), y = as.numeric(lat_g),
             fill = sqrt(n))) + geom_tile() +
    scale_fill_gradient(low = '#edf8b1', high = '#2c7fb8', na.value = 'white') +
    theme_bw() +
    labs(x=NULL, y=NULL, title = "Network members (n=8132)") +
    theme(axis.title=element_blank(),
          axis.text=element_blank(),
          axis.ticks=element_blank()),

  ggplot(mu_lonlat_df,
         aes(x = as.numeric(lon_mu),
             y = as.numeric(lat_mu),
             fill = sqrt(n))) +
    geom_tile() +
    scale_fill_gradient(low = '#edf8b1', high = '#2c7fb8', na.value = 'white') +
    theme_bw() +
    labs(x=NULL, y=NULL, title = "Meetup members (n=97,808)") +
    theme(axis.title=element_blank(),
          axis.text=element_blank(),
          axis.ticks=element_blank()),
  ncol = 2)
```




```
pdf(file = "figure/heatmap-dist-mu-g.pdf",
    width  = 8, height = 3.5)
grid.arrange(
  ggplot(g_lonlat_df,
         aes(x = as.numeric(lon_g),
             y = as.numeric(lat_g),
             fill = sqrt(n))) +
    geom_tile() +
    scale_fill_gradient(low = '#edf8b1', high = '#2c7fb8', na.value = 'white') +
    theme_bw() +
```

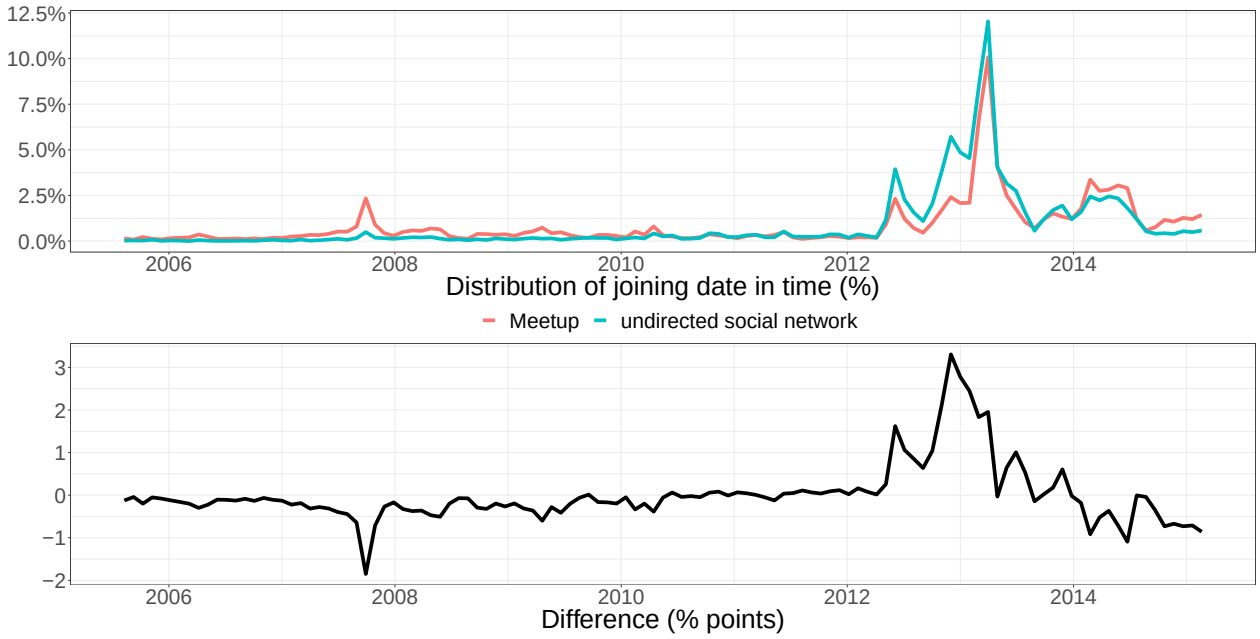
```
    labs(x=NULL, y=NULL, title = "Network members (n=8132)") +
    theme(axis.title=element_blank(),
          axis.text=element_blank(),
          axis.ticks=element_blank(),
          text = element_text(family = "Palatino")),

  ggplot(mu_lonlat_df,
         aes(x = as.numeric(lon_mu),
             y = as.numeric(lat_mu),
             fill = sqrt(n))) +
    geom_tile() +
    scale_fill_gradient(low = '#edf8b1', high = '#2c7fb8', na.value = 'white') +
    theme_bw() +
    labs(x=NULL, y=NULL, title = "Meetup members (n=97,808)") +
    theme(axis.title=element_blank(),
          axis.text=element_blank(),
          axis.ticks=element_blank(),
          text = element_text(family = "Palatino")),
  ncol = 2)
dev.off()
```

```
## pdf
##   2
```

Figure 5

```
member15 <- meetup_members_2015
member15$joined <- as.Date(member15$joined)

g_el <- as_edgelist(g)
g_el_lonlat_df <- data.frame(lon.x = V(g)$lon[match(g_el[,1], V(g)$name)],
                             lat.x = V(g)$lat[match(g_el[,1], V(g)$name)],
                             lon.y = V(g)$lon[match(g_el[,2], V(g)$name)],
                             lat.y = V(g)$lat[match(g_el[,2], V(g)$name)])

g_el_lonlat_df$dist <- distVincentySphere(g_el_lonlat_df[,1:2], g_el_lonlat_df[,3:4])
E(g)$distance <- g_el_lonlat_df$dist

sequence <- seq(from = min(member15$joined), max(member15$joined), by = 30)
ts_meetup_df <- as.data.frame(table(cut(member15$joined, breaks = sequence,
                                        labels = sequence[2:length(sequence)])))
ts_meetup_df$Prop <- ts_meetup_df$Freq / sum(ts_meetup_df$Freq)
ts_meetup_df$cumsum <- cumsum(ts_meetup_df$Prop)
ts_meetup_df$what <- "Meetup"

ts_g_df <- as.data.frame(table(cut(as.Date(V(g)$joined_mu), breaks = sequence,
                                   labels = sequence[2:length(sequence)])))

ts_g_df$Prop <- ts_g_df$Freq / sum(ts_g_df$Freq)
ts_g_df$cumsum <- cumsum(ts_g_df$Prop)
ts_g_df$what <- "undirected social network"
```

```
ts_df <- rbind(ts_meetup_df, ts_g_df)
ts_diff_df <- merge(ts_meetup_df, ts_g_df, by = "Var1")
ts_diff_df$diff <- ts_diff_df$Prop.y - ts_diff_df$Prop.x

p1 <-
  ggplot(ts_df, aes(x=as.Date(Var1), y=Prop, colour=what)) +
  geom_line(size = 2) + scale_y_continuous(label = percent) + theme_bw() +
  labs(y=NULL, colour=NULL, x="Distribution of joining date in time (%)") +
  theme(legend.position = 'bottom', text = element_text(size=30))

p2 <-
  ggplot(ts_diff_df, aes(x=as.Date(Var1), y=diff*100)) +
  geom_line(size = 2) + theme_bw() + labs(y=NULL, x="Difference (% points)") +
  theme(text = element_text(size=30))

ggarrange(p1, p2, ncol = 1)
```




```
p1 <-
  ggplot(ts_df, aes(x=as.Date(Var1), y=Prop, colour=what)) +
  geom_line(size = 2) + scale_y_continuous(label = percent) + theme_bw() +
  labs(y=NULL, colour=NULL, x="Distribution of joining date in time (%)") +
  theme(legend.position = 'bottom',
        text = element_text(family = "Palatino"))

p2 <-
  ggplot(ts_diff_df, aes(x=as.Date(Var1), y=diff*100)) +
  geom_line(size = 2) + theme_bw() + labs(y=NULL, x="Difference (% points)") +
  theme(text = element_text(family = "Palatino"))

pdf(file = "figure/ts-mu-g-joining-date.pdf",
    width  = 10, height = 5)
ggarrange(p1, p2, ncol = 1)
dev.off()
```

```
## pdf
##   2
```

Data analysis

Spatial polygons of Italian regions

Shapefiles are produced by Istat and also available here: https://www.istat.it/it/archivio/124086

```
region_2013.sp <- readOGR('.', 'Reg01012013_g_WGS84')
```

```
## OGR data source with driver: ESRI Shapefile
## Source: ".", layer: "Reg01012013_g_WGS84"
## with 20 features
## It has 5 fields
```

```
region_2013.sp <-
  spTransform(region_2013.sp,
              "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
region_2013.sp$COD_REG <- as.numeric(as.character(region_2013.sp$COD_REG))
```

ISTAT Survey

The following chunks are not replicable if you don't have access to `DF_AVQ_A2013.RData`, which is password protected in this repository (as third party data). You can get download the data package from the Istat website at this page: www.istat.it/en/archivio/129959 (accessed October 2018).

```
load("DF_AVQ_A2013.RData")

asIndexComp <- function(x) {
  x <- as.numeric(x)
  x[is.na(x)] <- 0
  x
}

reverse <- function(x) {
  if (is.na(x)) {
    res <- NA
  } else if (x == 1) {
    res <- 3
  } else if (x == 2) {
    res <- 2
  } else if (x == 3) {
    res <- 1
  } else if (x == 4) {
    res <- 0
  } else {
    res <- NA
  }
  return(res)
}

nrow(DF_AVQ_A2013)
```

```
## [1] 20275
```

```
# Demographics
DF_AVQ_A2013$male <- DF_AVQ_A2013$sesso == 1
DF_AVQ_A2013$age <- rescale(DF_AVQ_A2013$eta, from = c(1,18), to = c(0,1))
DF_AVQ_A2013$married <- DF_AVQ_A2013$stciv == 2
DF_AVQ_A2013$unemployed <-  DF_AVQ_A2013$cond %in% 2:3
DF_AVQ_A2013$income <- as.factor(DF_AVQ_A2013$redpr)
levels(DF_AVQ_A2013$income) <- c('emp','selfemp', 'pens', 'soc', 'rev', 'fam')
DF_AVQ_A2013$tv_hours <- DF_AVQ_A2013$hhtel
DF_AVQ_A2013$tv_hours[DF_AVQ_A2013$tv_hours == 99] <- NA
DF_AVQ_A2013$edu <- NA
DF_AVQ_A2013$edu[DF_AVQ_A2013$istr %in% 10:11] <- 1
DF_AVQ_A2013$edu[DF_AVQ_A2013$istr==9] <- 2
DF_AVQ_A2013$edu[DF_AVQ_A2013$istr==7] <- 3
DF_AVQ_A2013$edu[DF_AVQ_A2013$istr %in% 1:2] <- 4

# Geography
region_labels <-
  c("Piemonte", "Valle d'Aosta/Vallée d'Aoste", "Lombardia", "Bolzano", "Trento", "Veneto",
    "Friuli-Venezia Giulia", "Liguria", "Emilia-Romagna", "Toscana", "Umbria", "Marche",
    "Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia",
    "Sardegna", "Trentino-Alto Adige/Südtirol")
names(region_labels) <-
  c("010", "020", "030", "041", "042", "050",
    "060", "070", "080", "090",
    "100", "110", "120", "130",
    "140", "150", "160", "170",
    "180", "190", "200", "040")
DF_AVQ_A2013$reg[DF_AVQ_A2013$reg %in% c(41,42)] <- 040
DF_AVQ_A2013$COD_REG <- as.numeric(gsub("0$", "", DF_AVQ_A2013$reg))
DF_AVQ_A2013$DEN_REG <-
  region_labels[match(sprintf("%03d", DF_AVQ_A2013$reg), names(region_labels))]
DF_AVQ_A2013$DEN_REG <-
  factor(DF_AVQ_A2013$DEN_REG,
         levels = region_labels[order(names(region_labels))])
DF_AVQ_A2013$north <-
  (!DF_AVQ_A2013$DEN_REG %in%
     c("Abruzzo", "Puglia", "Basilicata",
       "Campania", "Calabria", "Molise", "Sicilia", "Sardegna"))
DF_AVQ_A2013$north[is.na(DF_AVQ_A2013$DEN_REG)] <- NA

DF_AVQ_A2013$north_south <- ifelse(DF_AVQ_A2013$north, "North", "South")
prop.table(table(DF_AVQ_A2013$north_south))
```

```
##
##     North     South
## 0.5871763 0.4128237
```

```
# Social capital: networking
## `sc_net` without political participation
DF_AVQ_A2013$sc_net <-
  as.logical(as.numeric(with(DF_AVQ_A2013,
                             asIndexComp(volon==4) +
                               asIndexComp(psind == 4) +
```

```
                              asIndexComp(pgrvo == 6) +
                              asIndexComp(paeco == 2) +
                              asIndexComp(pcult == 4) +
                              asIndexComp(paspro == 6) > 0)))
summary(DF_AVQ_A2013$sc_net)
```

```
##    Mode   FALSE    TRUE
## logical   16327    3948
```

```
prop.table(table(DF_AVQ_A2013$sc_net))
```

```
##
##     FALSE      TRUE
## 0.8052774 0.1947226
```

```
## Political participation (active)
DF_AVQ_A2013$pol_part_act <-
  as.numeric(with(DF_AVQ_A2013,
                  asIndexComp(ppapo == 2) +
                    asIndexComp(volpa == 8))) > 0
summary(DF_AVQ_A2013$pol_part_act)
```

```
##    Mode   FALSE    TRUE
## logical   19573     702
```

```
prop.table(table(DF_AVQ_A2013$pol_part_act))
```

```
##
##      FALSE       TRUE
## 0.96537608 0.03462392
```

```
## `sc_net` now includes also political participation
DF_AVQ_A2013$sc_net <-
  DF_AVQ_A2013$sc_net == TRUE | DF_AVQ_A2013$pol_part_act == TRUE
summary(DF_AVQ_A2013$sc_net)
```

```
##    Mode   FALSE    TRUE
## logical   16085    4190
```

```
prop.table(table(DF_AVQ_A2013$sc_net))
```

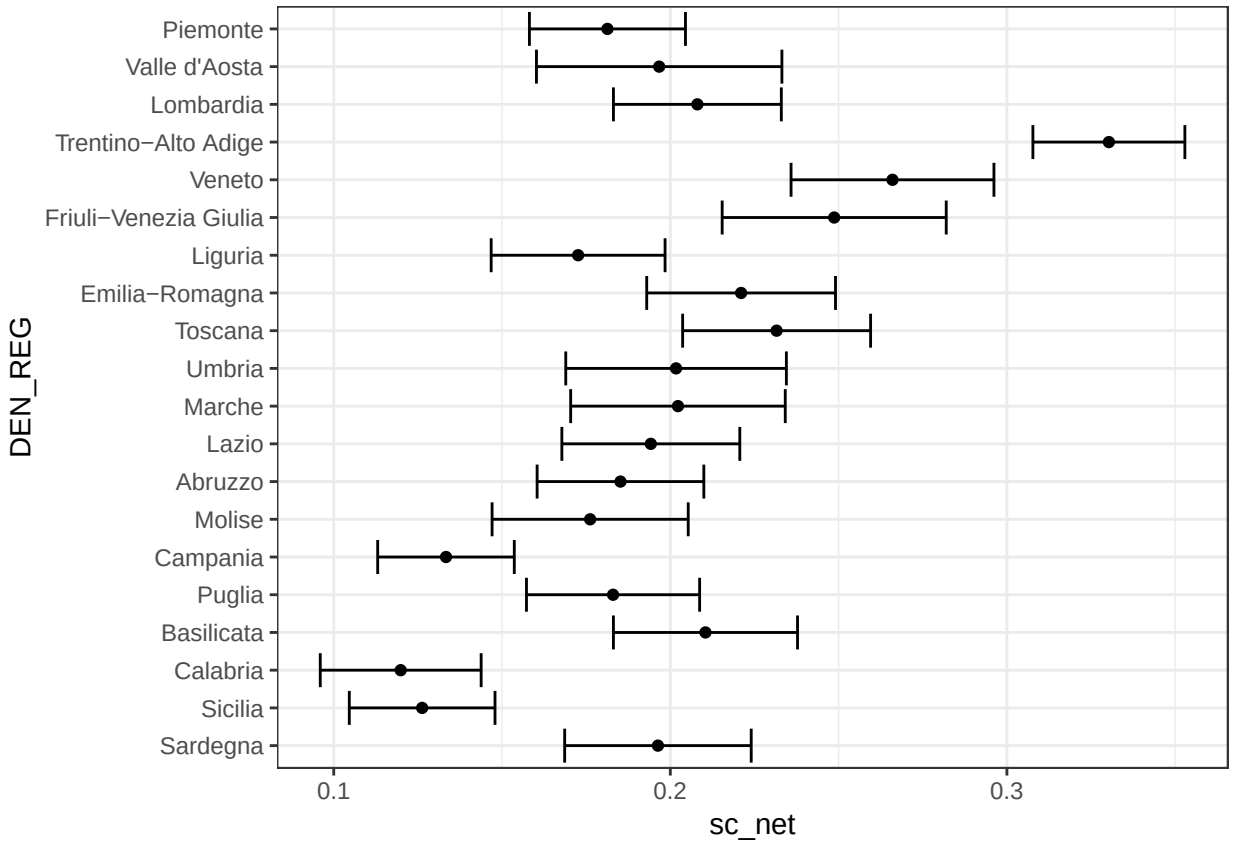
```
##
##     FALSE      TRUE
## 0.7933416 0.2066584
```

```
# `coemicro` is the weighting variable for regional-level averages
DF_AVQ_A2013$coemicro <-
  as.numeric(as.character(DF_AVQ_A2013$coemicro))

# Social capital: Trust.
DF_AVQ_A2013$sc_trust <-
  rescale(
    with(DF_AVQ_A2013,sapply(fidu3, reverse)),
    to = c(0, 1),
    from = c(0, 3))
summary(DF_AVQ_A2013$sc_trust)
```

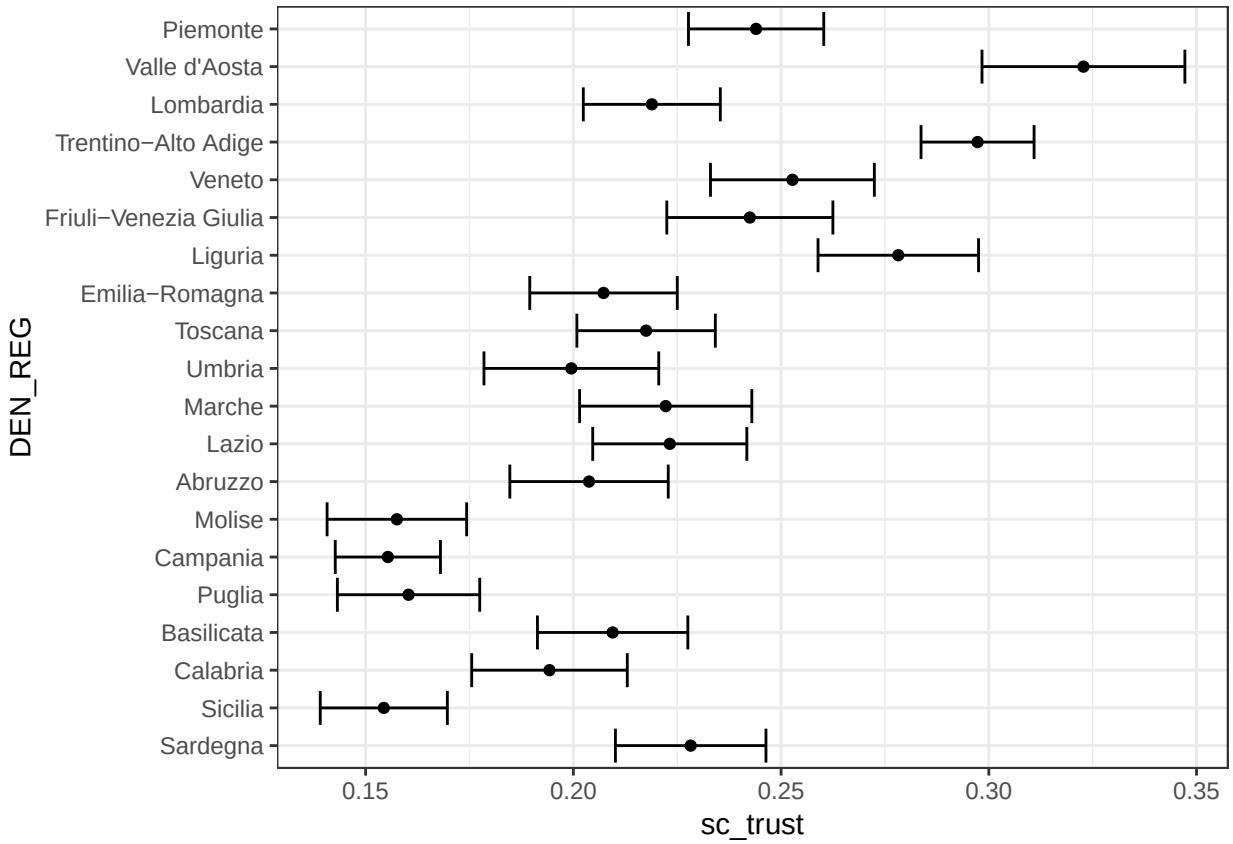
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
```

```
##  0.0000  0.0000  0.3333  0.2198  0.3333  1.0000    3132
```

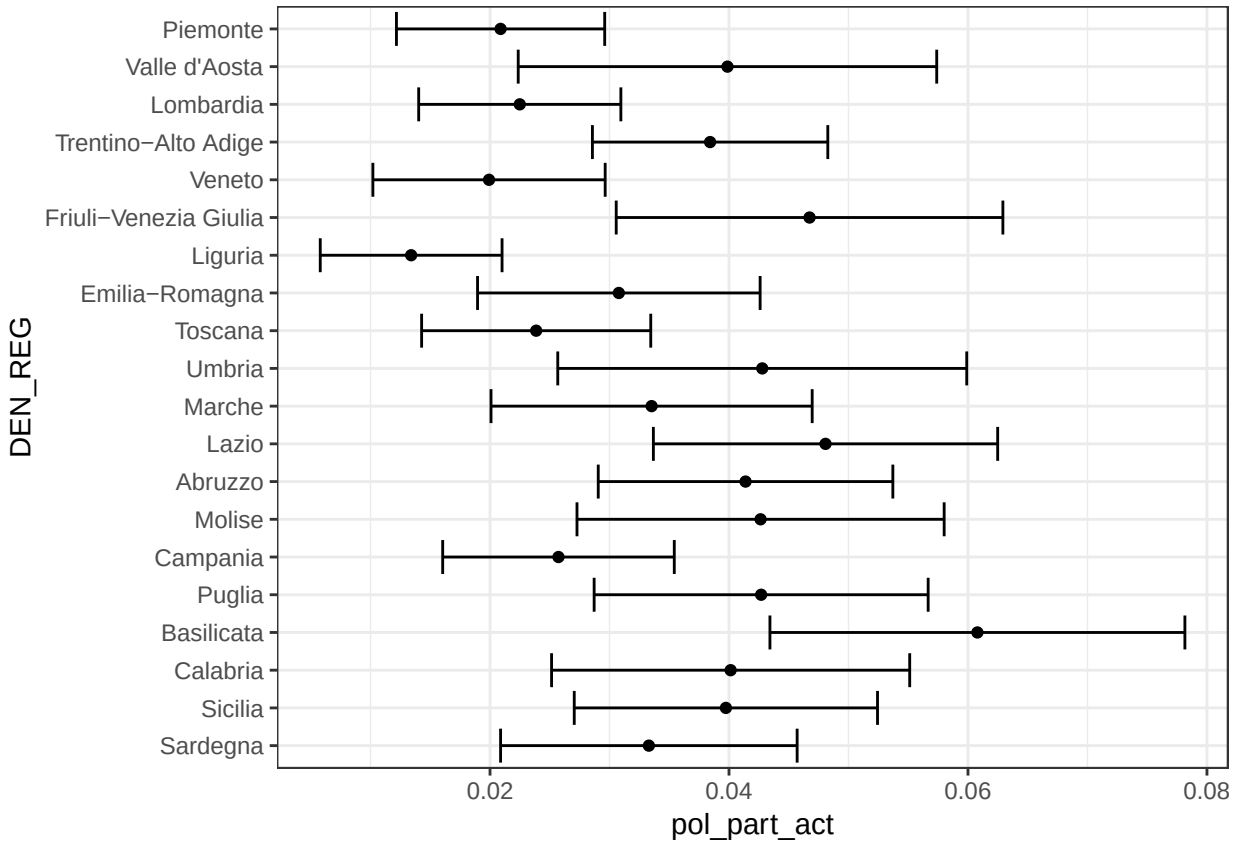
```
# Logistic regression
log_sc_net <-
  glm(sc_net ~ age + male + married + edu + unemployed + tv_hours + income + north_south,
      data = DF_AVQ_A2013,
      family = 'binomial')

lm_sc_trust <-
  glm(sc_trust ~ sc_net + age + male + married + edu + unemployed + tv_hours + income + north_south,
      data = DF_AVQ_A2013,
      family = 'binomial')

# Survey responses averaged at regional level
survey_reg_avg <-
  srvyr::as_survey_design(DF_AVQ_A2013, strata = COD_REG, weights = coemicro)

survey_reg_avg <-
  survey_reg_avg %>%
  group_by(COD_REG) %>%
  summarize_at(vars(sc_net, sc_trust, pol_part_act),
               survey_mean,
               vartype = "ci",
               na.rm = TRUE)

survey_reg_avg$DEN_REG <-
  region_2013.sp@data$DEN_REG[match(survey_reg_avg$COD_REG,
                                    region_2013.sp@data$COD_REG)]
survey_reg_avg$DEN_REG <- gsub("/(.*)$", "", survey_reg_avg$DEN_REG)
survey_reg_avg$DEN_REG <- factor(survey_reg_avg$DEN_REG, levels = rev(survey_reg_avg$DEN_REG))

ggplot(survey_reg_avg) +
  geom_point(aes(y=sc_net, x=DEN_REG)) +
  geom_errorbar(aes(ymin = sc_net_low, ymax = sc_net_upp, x=DEN_REG)) +
  coord_flip() +
  theme_bw()
```




```
ggplot(survey_reg_avg) +
  geom_point(aes(y=sc_trust, x=DEN_REG)) +
  geom_errorbar(aes(ymin = sc_trust_low, ymax = sc_trust_upp, x=DEN_REG)) +
  coord_flip() +
  theme_bw()
```



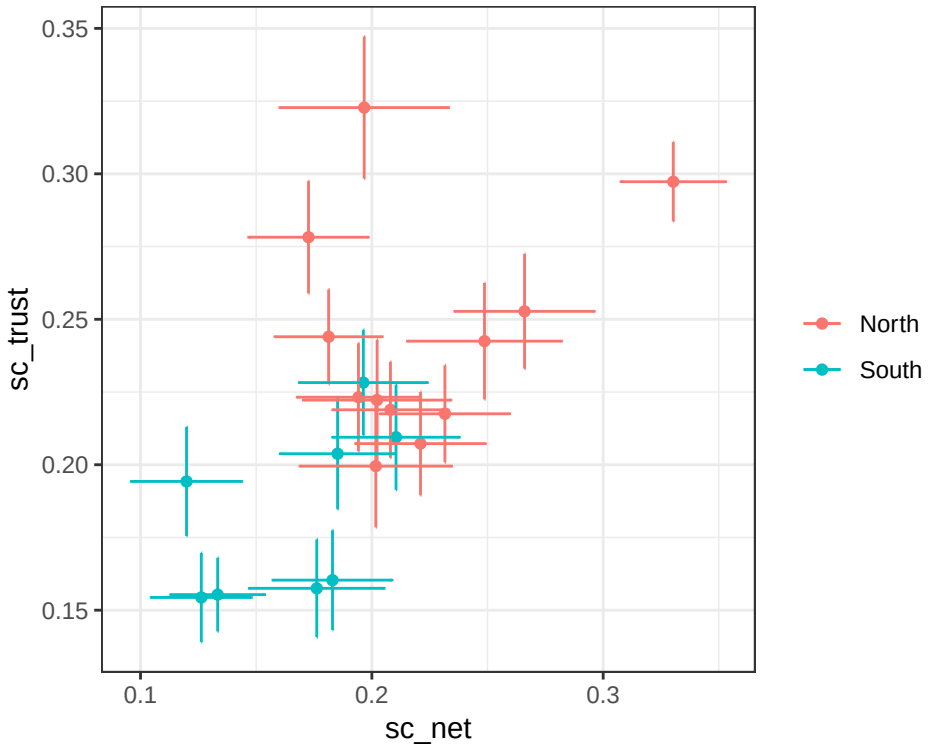

```
ggplot(survey_reg_avg) +
  geom_point(aes(y=pol_part_act, x=DEN_REG)) +
  geom_errorbar(aes(ymin = pol_part_act_low, ymax = pol_part_act_upp, x=DEN_REG)) +
  coord_flip() +
  theme_bw()
```




```
survey_reg_avg$north_south <- ifelse(survey_reg_avg$COD_REG < 13, 'North', 'South')
```

Figure 6

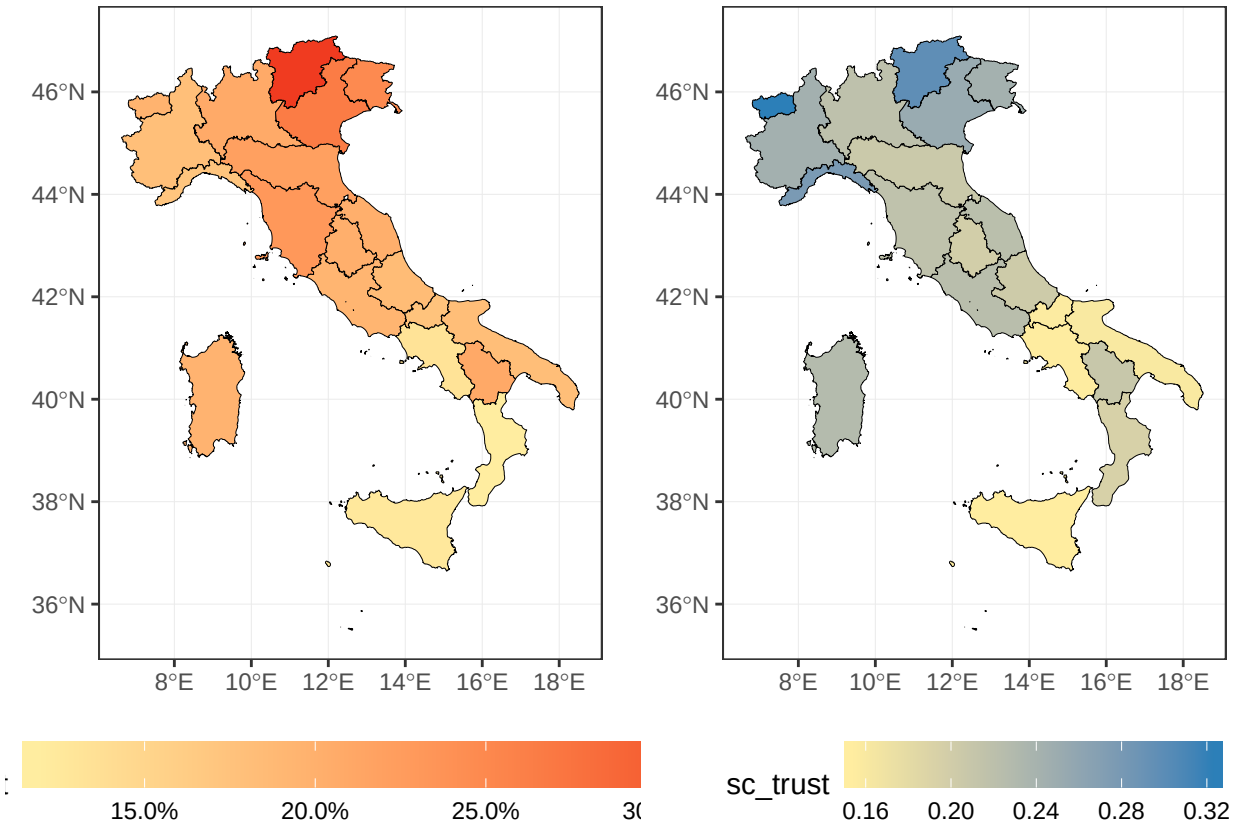
```
ggplot(survey_reg_avg) +
  geom_point(aes(x=sc_net, y=sc_trust, colour = north_south )) +
  geom_errorbar(aes(x=sc_net, ymin=sc_trust_low, ymax=sc_trust_upp, colour = north_south),
                width = 0) +
  geom_errorbarh(aes(y=sc_trust, xmin=sc_net_low, xmax=sc_net_upp, colour = north_south)) +
  theme_bw() +
  labs(colour=NULL)
```




```
ggsave(filename = "figure/sc-ita-regions.pdf", dpi = 300,
       width = 5, height = 4,
       ggplot(survey_reg_avg) +
         geom_point(aes(x=sc_net, y=sc_trust, colour = north_south )) +
         geom_errorbar(aes(x=sc_net, ymin=sc_trust_low, ymax=sc_trust_upp,
                           colour = north_south),
                       width = 0) +
         geom_errorbarh(aes(y=sc_trust, xmin=sc_net_low, xmax=sc_net_upp,
                            colour = north_south)) +
         theme_bw() + labs(colour=NULL))
```

Figure 12

```
region_2013.sf <- st_as_sf(region_2013.sp)

region_2013.sf$sc_trust <-
  survey_reg_avg$sc_trust[match(region_2013.sf$COD_REG, survey_reg_avg$COD_REG)]
region_2013.sf$sc_net <-
  survey_reg_avg$sc_net[match(region_2013.sf$COD_REG, survey_reg_avg$COD_REG)]

grid.arrange(
  ggplot(region_2013.sf) + geom_sf(aes(fill = sc_net), colour = 'black', size = .05) +
    scale_fill_gradient(low = '#ffeda0', high = '#f03b20', label = percent) + theme_bw() +
    theme(legend.position = 'bottom', legend.key.width=unit(2,"cm")) ,
  ggplot(region_2013.sf) + geom_sf(aes(fill = sc_trust), colour = 'black', size = .05) +
    scale_fill_gradient(low = '#ffeda0', high = '#2c7fb8')  + theme_bw() +
    theme(legend.position = 'bottom', legend.key.width=unit(1,"cm")),
  ncol=2)
```




```
pdf(file = "figure/map-ita-reg-sc.pdf", width  = 9, height = 4.5)
grid.arrange(
  ggplot(region_2013.sf) + geom_sf(aes(fill = sc_net), colour = 'black', size = .05) +
    scale_fill_gradient(low = '#ffeda0', high = '#f03b20', label = percent) + theme_bw() +
    theme(legend.position = 'bottom', legend.key.width=unit(1.4,"cm"),
          text = element_text(family = "Palatino")) ,
  ggplot(region_2013.sf) + geom_sf(aes(fill = sc_trust), colour = 'black', size = .05) +
    scale_fill_gradient(low = '#ffeda0', high = '#2c7fb8')  + theme_bw() +
    theme(legend.position = 'bottom', legend.key.width=unit(1.4,"cm"),
          text = element_text(family = "Palatino")),
  ncol=2)
```

Table 2

```
stargazer(log_sc_net, lm_sc_trust)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Fri, Oct 19, 2018 - 03:50:11

```
stargazer(log_sc_net, lm_sc_trust,
          type = 'latex', out = 'table/sc-net-survey-mod.tex',
          report = "vc*",
          float = FALSE,
          style = 'ajps')
stargazer(log_sc_net, lm_sc_trust,
          type = 'html', out = 'table/sc-net-survey-mod.html',
          report = "vc*",
```

Table 1:

	Dependent variable:	
	sc_net	sc_trust
	(1)	(2)
sc_net		0.274***
		(0.051)
age	0.121	0.157
	(0.178)	(0.175)
male	0.421***	−0.031
	(0.046)	(0.047)
married	0.114**	0.056
	(0.049)	(0.049)
edu	0.604***	0.116***
	(0.027)	(0.028)
unemployed	−0.241**	−0.198**
	(0.094)	(0.094)
tv_hours	−0.089***	−0.021
	(0.015)	(0.013)
incomeselfemp	0.106	0.104
	(0.074)	(0.079)
incomepens	−0.259***	0.013
	(0.079)	(0.079)
incomesoc	−0.173	0.123
	(0.159)	(0.154)
incomerev	−0.839**	0.002
	(0.340)	(0.290)
incomefam	−0.224***	0.001
	(0.069)	(0.069)
north_southSouth	−0.353***	−0.318***
	(0.047)	(0.047)
Constant	−2.450***	−1.527***
	(0.149)	(0.149)
Observations	12,103	11,911
Log Likelihood	−6,253.930	−4,823.210
Akaike Inf. Crit.	12,533.860	9,674.419
Note:	*p<0.1; **p<0.05; ***p<0.01	

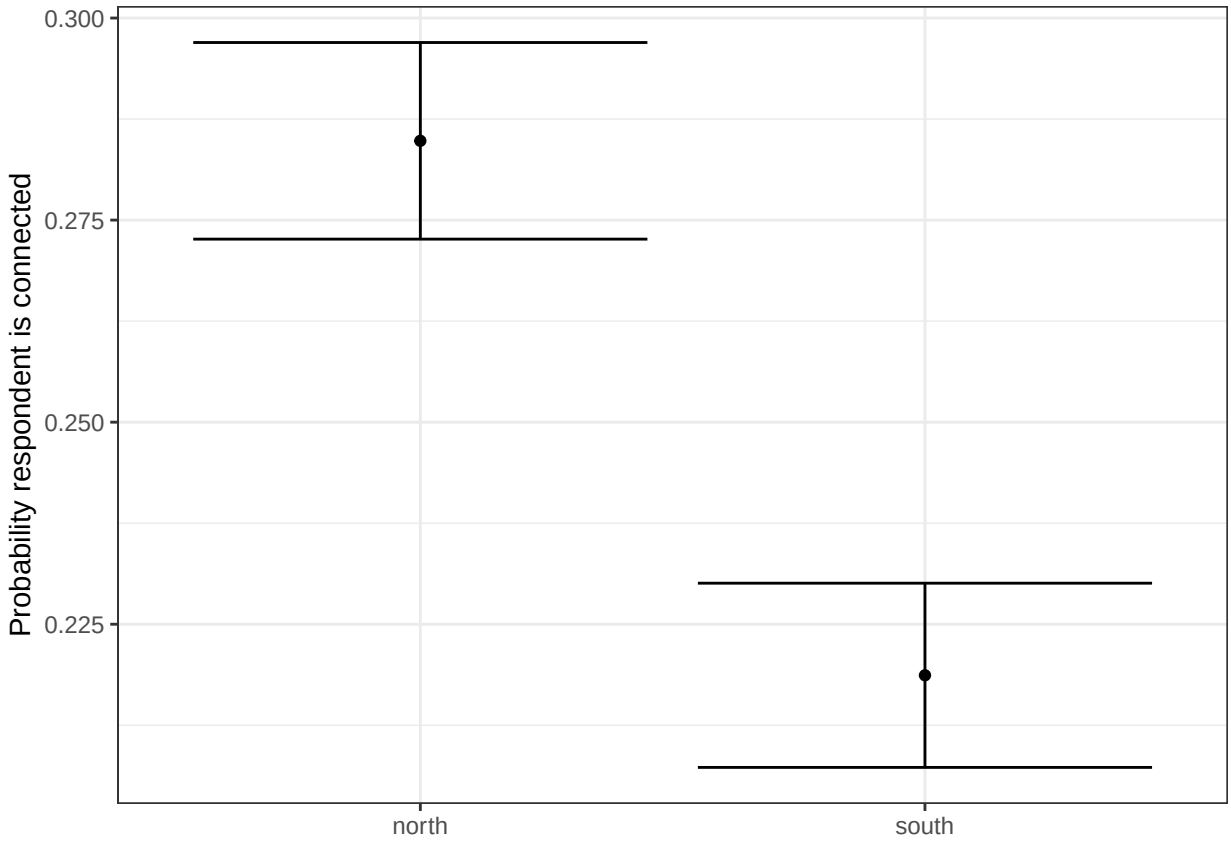
```
          float = FALSE,
          style = 'ajps')
```

Logistic regression prediction: MALE

```
pr <-
  predict(log_sc_net,
          newdata = data.frame(age = 0.6470588,
                               male = TRUE,
                               married = FALSE,
                               edu = 2,
                               unemployed = FALSE,
                               tv_hours = 2,
                               income = 'emp',
                               north_south = c('North', 'South')),
          type="response", se.fit = TRUE)
pr
```

```
## $fit
##         1         2
## 0.2848085 0.2186827
##
## $se.fit
##          1          2
## 0.01216219 0.01139579
##
## $residual.scale
## [1] 1
```

```
pr_df <- data.frame(where = c('north','south'),
                    fit = pr$fit,
                    se.fit = pr$se.fit)
ggplot(pr_df, aes(x = where, y = fit)) +
  geom_point() +
  geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit)) +
  theme_bw() +
  labs(x = NULL, y = "Probability respondent is connected")
```

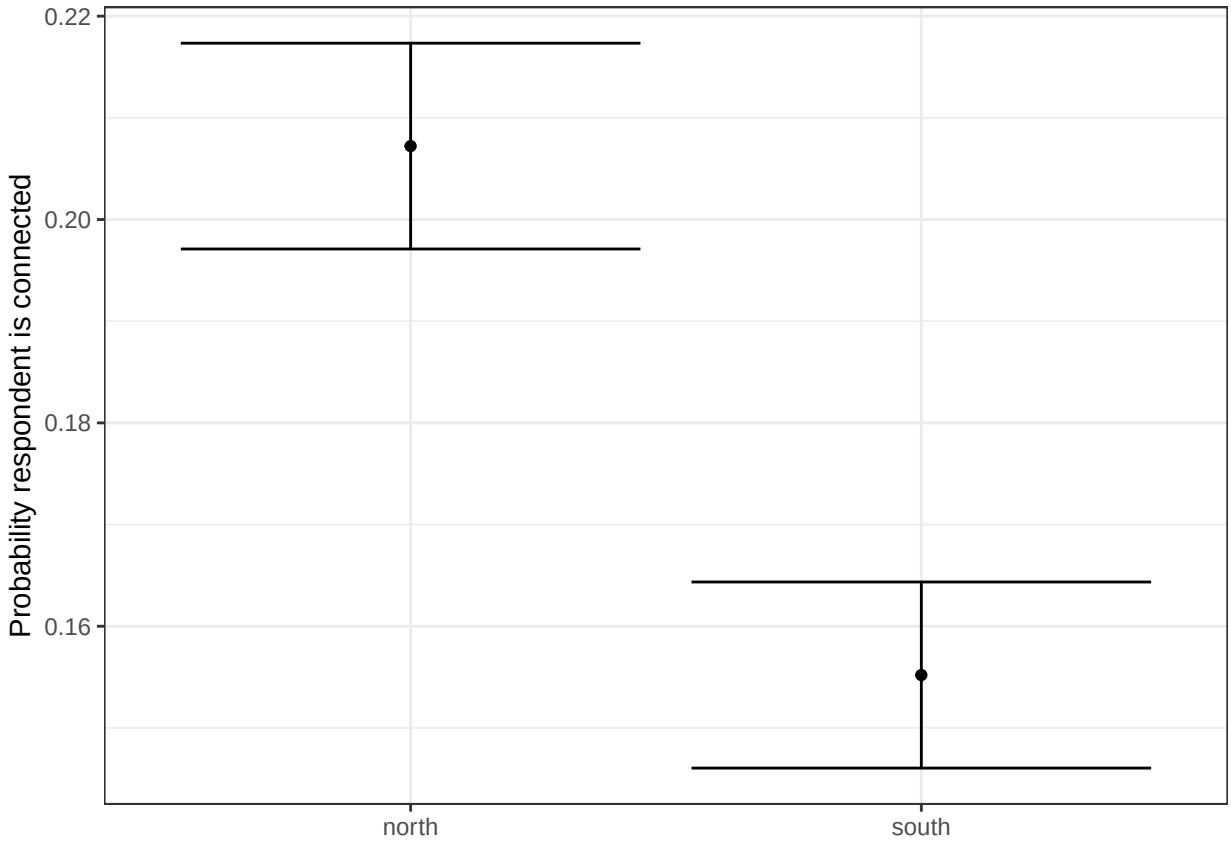


Logistic regression prediction: FEMALE

```
pr <-
  predict(log_sc_net,
          newdata = data.frame(age = 0.6470588,
                               male = FALSE,
                               married = FALSE,
                               edu = 2,
                               unemployed = FALSE,
                               tv_hours = 2,
                               income = 'emp',
                               north_south = c('North', 'South')),
          type="response", se.fit = TRUE)
pr
```

```
## $fit
##         1         2
## 0.2072220 0.1552009
##
## $se.fit
##           1           2
## 0.010119259 0.009150905
##
## $residual.scale
## [1] 1
```

```
pr_df <- data.frame(where = c('north','south'),
                    fit = pr$fit,
                    se.fit = pr$se.fit)
ggplot(pr_df, aes(x = where, y = fit)) +
  geom_point() +
  geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit)) +
  theme_bw() +
  labs(x = NULL, y = "Probability respondent is connected")
```



Diffusion

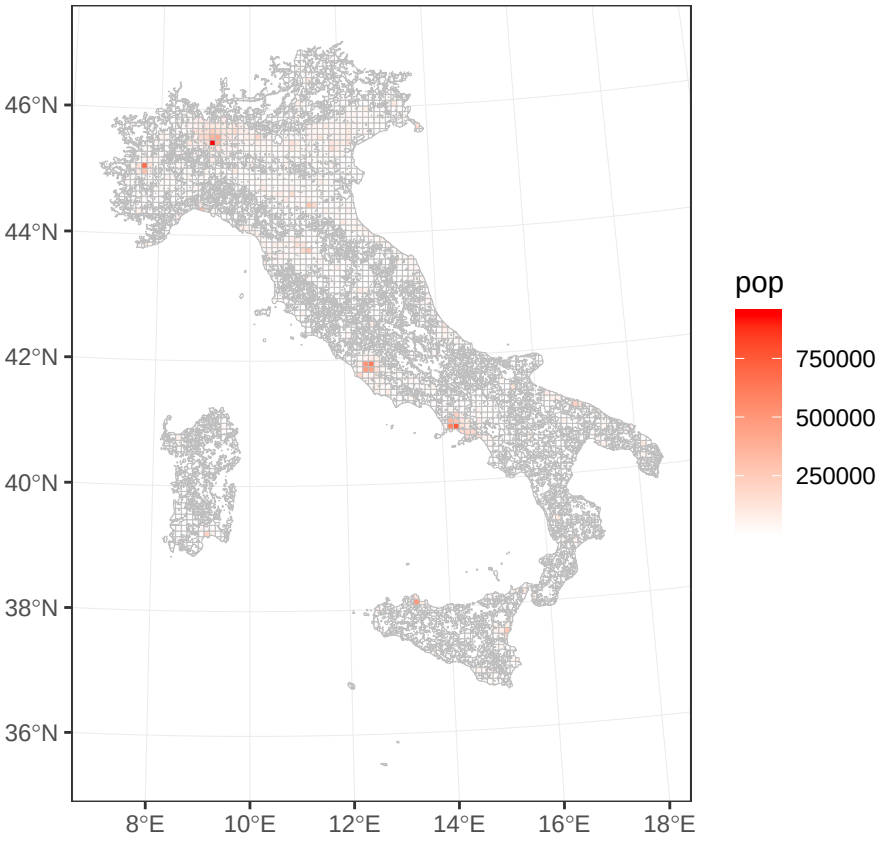
Create geographic grid

The geographic grid is a simplified version of the GEOSTAT 2011. The file is also available from https://www.istat.it/it/archivio/155162.

```
# See https://www.istat.it/it/archivio/155162
geostat_grid <- readOGR(".",'GEOSTAT_grid_POP_1K_IT_2011')

geostat_grid_simp.sf <- st_as_sf(geostat_grid)
geostat_grid_simp.sf <- geostat_grid_simp.sf %>% filter(POP > 0)
geostat_grid_simp.sf$GRID_ID_simp <- gsub("\\dE", "E", geostat_grid_simp.sf$GRID_ID)
geostat_grid_simp.sf$GRID_ID_simp <- gsub("\\d$", "", geostat_grid_simp.sf$GRID_ID_simp)

geostat_grid_simp.sf <-
  geostat_grid_simp.sf %>%
```

```
  group_by(GRID_ID_simp) %>%
  summarize(pop = sum(POP))

geostat_grid_simp.sf <-
  geostat_grid_simp.sf %>%
  filter(pop > 0)

save(geostat_grid_simp.sf, file = 'replication_file_07_geostat_grid.RData')
```

```
load('replication_file_07_geostat_grid.RData')
ggplot(geostat_grid_simp.sf) +
  geom_sf(aes(fill=pop), color = 'grey', size = 0.05) +
  scale_fill_gradient(low = "white", high = "red") +
  theme_bw()
```




```
# Number of cells
nrow(geostat_grid_simp.sf)
```

```
## [1] 3396
```

```
# Mean area
mean(st_area(geostat_grid_simp.sf))
```

```
## 60015607 m^2
```

Number of Meetup members by grid cells

To replicate the following chunks:

```
R -e 'setwd("/path/to/directory");
source("replication_code_06_measure_geographic_diffusion.R");'
```

```
read_chunk('replication_code_06_measure_geographic_diffusion.R')
```

```
# Packages
library(data.table)
library(parallel)
library(sp)
library(sf)
library(rgdal)

# Functions
sqliteGetTable <- function(database, table) {
  require(DBI)
  require(RSQLite)
  con <- dbConnect(RSQLite::SQLite(), dbname = database)
  query <- dbSendQuery(con, paste("SELECT * FROM ", table, ";", sep=""))
  result <- fetch(query, n = -1)
  dbClearResult(query)
  dbDisconnect(con)
  return(result)
}

countMembersByGrid <- function(i) {
  print(i)
  this_meetup_members.sp <- meetup_members.sp[meetup_members.sp$joined <= sequence[i],]
  res <- over(this_meetup_members.sp, geostat_grid.sp)
  res <- as.data.frame(table(res$GRID_ID_simp))
  res$member_density <- res$member_density <- res$Freq / sum(res$Freq)
  ita_popgrid$these_members <- res$Freq[match(ita_popgrid$GRID_ID_simp, res$Var1)]
  ita_popgrid$these_members[is.na(ita_popgrid$these_members)] <- 0
  return(data.frame(date = sequence[i],
                    members = ita_popgrid$these_members,
                    GRID_ID_simp = ita_popgrid$GRID_ID_simp,
                    COD_REG = ita_popgrid$COD_REG))
}

load('replication_file_07_geostat_grid.RData')

region_2013.sp <- readOGR('.', 'Reg01012013_g_WGS84')
region_2013.sp <-
  spTransform(region_2013.sp, "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
region_2013.sp$COD_REG <- as.numeric(as.character(region_2013.sp$COD_REG))

geostat_grid.sp <- as(geostat_grid_simp.sf,'Spatial')
geostat_grid.sp <-
  spTransform(geostat_grid.sp,
              '+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0')
res <- over(geostat_grid.sp, region_2013.sp)
geostat_grid.sp$COD_REG <- res$COD_REG
geostat_grid.sp$DEN_REG <- as.character(res$DEN_REG)
ita_popgrid <- geostat_grid.sp[,c('GRID_ID_simp','pop','COD_REG')]
```

```
# Calculate density from entire meetup dataset
meetup_members <- sqliteGetTable('meetup_mar15.sqlite', 'member')
meetup_members <- meetup_members[meetup_members$country %in% c("it"),]
meetup_members$joined <- as.Date(meetup_members$joined)
meetup_members.sp <-
  SpatialPointsDataFrame(meetup_members[,c('lon','lat')],
                         data = meetup_members,
                         proj4string = CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))

sequence <-
  seq(from = as.Date('2005-07-23'),
      to = as.Date('2015-03-01'),
      by = 7)

no_cores <- detectCores() - 1
cl <- makeCluster(no_cores, type="FORK")
members_pop_grid_dt <- rbindlist(parLapply(cl, 1:length(sequence), countMembersByGrid))
stopCluster(cl)
members_pop_grid_dt$north_south <- ifelse(members_pop_grid_dt$COD_REG < 13, "North", "South")

members_pop_grid_dt$pop <-
  ita_popgrid$pop[match(members_pop_grid_dt$GRID_ID_simp, ita_popgrid$GRID_ID_simp)]

save(members_pop_grid_dt, file = "replication_file_08_geographic_diffusion.RData")
```

Figure 7

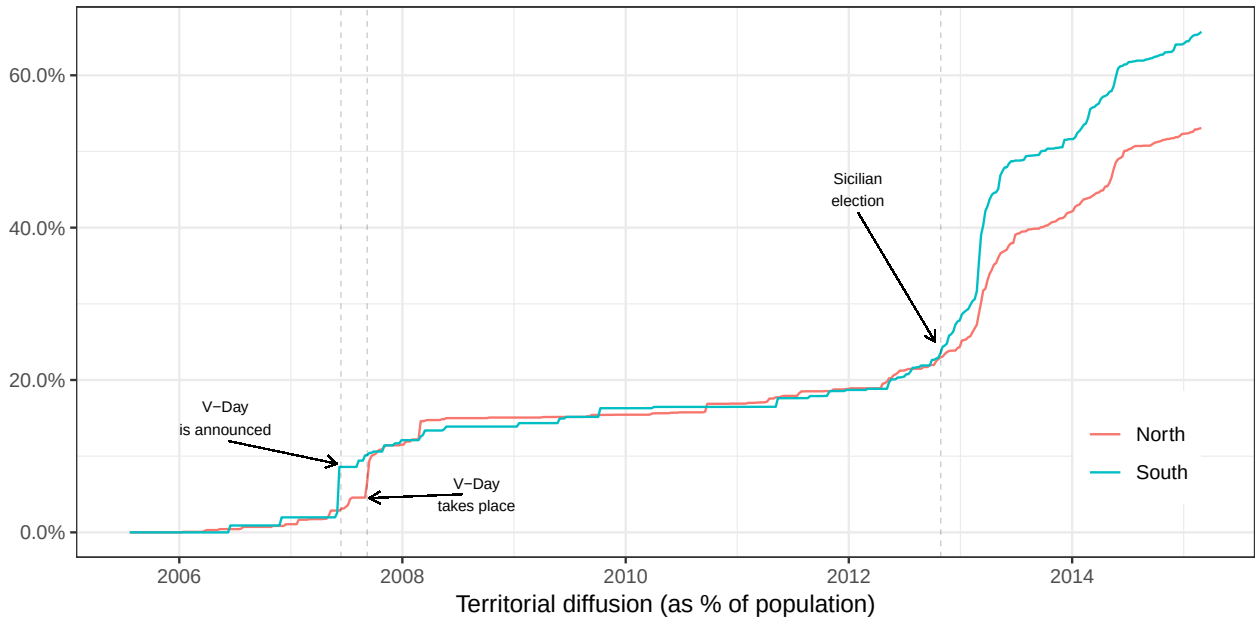
```
load("replication_file_08_geographic_diffusion.RData")

# Average number of residents for each Meetup member
1 /
  mean(members_pop_grid_dt$members[
    members_pop_grid_dt$date == max(members_pop_grid_dt$date) &
      members_pop_grid_dt$pop > 0] /
      members_pop_grid_dt$pop[
        members_pop_grid_dt$date == max(members_pop_grid_dt$date) &
          members_pop_grid_dt$pop > 0])
```

```
## [1] 1436.43
```

```
# Calculate density from entire meetup dataset
meetup_members <- meetup_members_2015
meetup_members <- meetup_members[meetup_members$country %in% c("it"),]
meetup_members$joined <- as.Date(meetup_members$joined)
meetup_members.sp <-
  SpatialPointsDataFrame(meetup_members[,c('lon','lat')],
                         data =
                           meetup_members,
                         proj4string =
                           CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))

geostat_grid.sp <- as(geostat_grid_simp.sf,'Spatial')
```

```
geostat_grid.sp <-
  spTransform(geostat_grid.sp,
              '+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0')
res <- over(geostat_grid.sp, region_2013.sp)
geostat_grid.sp$COD_REG <- res$COD_REG
geostat_grid.sp$DEN_REG <- as.character(res$DEN_REG)

ita_popgrid <- geostat_grid.sp[,c('GRID_ID_simp','pop','COD_REG')]
ita_cells <- ita_popgrid$GRID_ID_simp[!is.na(ita_popgrid$COD_REG)]
north_cells <- ita_popgrid$GRID_ID_simp[ita_popgrid$COD_REG < 13]
south_cells <- ita_popgrid$GRID_ID_simp[ita_popgrid$COD_REG >= 13]
ita_pop <- sum(ita_popgrid$pop)
north_pop <- sum(ita_popgrid$pop[ita_popgrid$COD_REG < 13], na.rm = T)
south_pop <- sum(ita_popgrid$pop[ita_popgrid$COD_REG >= 13], na.rm = T)

# A cell is activated if a person every 2000 is a member (1*0.0005)
stats_dt <-
  members_pop_grid_dt[!is.na(members_pop_grid_dt$COD_REG)] %>%
  group_by(date) %>%
  summarize(ita_pop = sum(pop[GRID_ID_simp %in% ita_cells & members > pop*0.0005]) /
              ita_pop,
            north_pop = sum(pop[GRID_ID_simp %in% north_cells & members > pop*0.0005]) /
              north_pop,
            south_pop = sum(pop[GRID_ID_simp %in% south_cells & members > pop*0.0005]) /
              south_pop)

stats_melted_dt <- melt(stats_dt, id.vars = c('date'))
stats_melted_dt <- stats_melted_dt[stats_melted_dt$variable != 'ita_pop',]
stats_melted_dt$what <- ifelse(stats_melted_dt$variable == "north_pop", "North", "South")

ggplot(stats_melted_dt, aes(x=date, y=value, colour = what)) +
  geom_line() +
  scale_y_continuous(label = percent) +
  labs(x="Territorial diffusion (as % of population)", y=NULL, colour=NULL) +
  geom_vline(xintercept = as.Date('2007-06-14'),
             size = .3, linetype = 2, alpha  = .2) +
  geom_vline(xintercept = as.Date('2007-09-08'),
             size = .3, linetype = 2, alpha  = .2) +
  geom_vline(xintercept = as.Date('2012-10-28'),
             size = .3, linetype = 2, alpha  = .2) +
  annotate("text", x = as.Date("2006-06-01"),
           y = 0.15, label = "V-Day\nis announced",
           size = 2.5) +
  geom_segment(aes(x=as.Date("2006-06-10"), xend=as.Date('2007-06-01'),
                   y=0.12, yend=0.09),
               size = .3, arrow = arrow(length = unit(0.2, "cm")),
               colour = 'black') +
  annotate("text", x = as.Date("2008-08-31"), y = 0.05,
           label = "V-Day\ntakes place",
           size = 2.5) +
  geom_segment(aes(x=as.Date("2008-07-15"), xend=as.Date('2007-09-15'),
                   y=0.05, yend=0.045),
               size = .3, arrow = arrow(length = unit(0.2, "cm")),
```

```
              colour = 'black') +
  annotate("text", x = as.Date("2012-01-31"), y = 0.45,
           label = "Sicilian\nelection",
           size = 2.5) +
  geom_segment(aes(x=as.Date("2012-01-31"), xend=as.Date('2012-10-10'),
                   y=0.42, yend=0.25),
               size = .3, arrow = arrow(length = unit(0.2, "cm")), colour = 'black') +
  theme_bw() + theme(legend.position = c(0.9, 0.2))
```



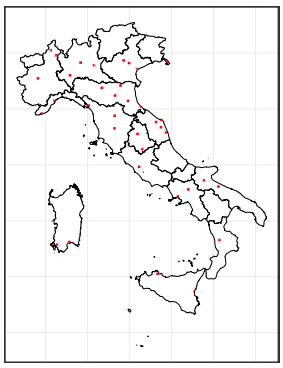

```
ggsave(filename = "figure/ts-meetup-diffusion.pdf", dpi = 300, width = 8, height = 4,
       ggplot(stats_melted_dt, aes(x=date, y=value, colour = what)) +
         geom_line() +
         scale_y_continuous(label = percent) +
         labs(x="Territorial diffusion (as % of population)", y=NULL, colour=NULL) +
         geom_vline(xintercept = as.Date('2007-06-14'),
                    size = .3, linetype = 2, alpha  = .2) +
         geom_vline(xintercept = as.Date('2007-09-08'),
                    size = .3, linetype = 2, alpha  = .2) +
         geom_vline(xintercept = as.Date('2012-10-28'),
                    size = .3, linetype = 2, alpha  = .2) +
         annotate("text", x = as.Date("2006-06-01"),
                  y = 0.15, label = "V-Day\nis announced",
                  size = 3) +
         geom_segment(aes(x=as.Date("2006-06-10"), xend=as.Date('2007-06-01'),
                          y=0.12, yend=0.09),
                      size = .3, arrow = arrow(length = unit(0.2, "cm")),
                      colour = 'black') +
         annotate("text", x = as.Date("2008-08-31"), y = 0.05,
                  label = "V-Day\ntakes place",
                  size = 3) +
         geom_segment(aes(x=as.Date("2008-07-15"), xend=as.Date('2007-09-15'),
                          y=0.05, yend=0.045),
                      size = .3, arrow = arrow(length = unit(0.2, "cm")),
                      colour = 'black') +
```

```
        annotate("text", x = as.Date("2012-01-31"), y = 0.45,
                 label = "Sicilian\nelection",
                 size = 3) +
        geom_segment(aes(x=as.Date("2012-01-31"), xend=as.Date('2012-10-10'),
                         y=0.42, yend=0.25),
                     size = .3, arrow = arrow(length = unit(0.2, "cm")),
                     colour = 'black') +
        theme_bw() + theme(legend.position = c(0.9, 0.2))
)
```

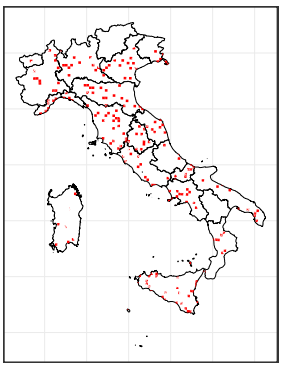
Figure 11

```
geostat_grid.sf <- geostat_grid_simp.sf

ita_grid_plot1 <- members_pop_grid_dt[date == '2007-09-15']
ita_grid_plot1 <- merge(geostat_grid.sf,ita_grid_plot1, by = 'GRID_ID_simp')
ita_grid_plot1$act <- ita_grid_plot1$members > (ita_grid_plot1$pop.x*0.0005)

ita_grid_plot2 <- members_pop_grid_dt[date == '2012-11-03']
ita_grid_plot2 <- merge(geostat_grid.sf, ita_grid_plot2, by = 'GRID_ID_simp')
ita_grid_plot2$act <- ita_grid_plot2$members > (ita_grid_plot2$pop.x*0.0005)

ita_grid_plot3 <- members_pop_grid_dt[date == '2013-03-02']
ita_grid_plot3 <- merge(geostat_grid.sf,ita_grid_plot3, by = 'GRID_ID_simp')
ita_grid_plot3$act <- ita_grid_plot3$members > (ita_grid_plot3$pop.x*0.0005)

ita_grid_plot4 <- members_pop_grid_dt[date == '2015-02-28']
ita_grid_plot4 <- merge(geostat_grid.sf,ita_grid_plot4, by = 'GRID_ID_simp')
ita_grid_plot4$act <- ita_grid_plot4$members > (ita_grid_plot4$pop.x*0.0005)

geostat_grid.sf$cut <-
  cut(geostat_grid.sf$pop,
      breaks = quantile(geostat_grid.sf$pop, probs = seq(0, 1, 0.1)),
      include.lowest = TRUE)

grid.arrange(
  ggplot() +
    geom_sf(data=region_2013.sf, colour = 'black', size = .05,
            fill = 'white') +
    labs(caption = '2007-09-15') +
    geom_sf(data=ita_grid_plot1[ita_grid_plot1$act == TRUE,],
            fill='red', colour = NA, show.legend=F) +
    theme_bw() + theme(axis.text.x=element_blank(),
                       axis.text.y=element_blank(),
                       axis.ticks=element_blank(),
                       axis.title.x=element_blank(),
                       axis.title.y=element_blank()),
  ggplot() +
    geom_sf(data=region_2013.sf, colour = 'black', size = .05,
            fill = 'white') +
    labs(caption = '2012-11-03') +
    geom_sf(data=ita_grid_plot2[ita_grid_plot2$act == TRUE,],
```

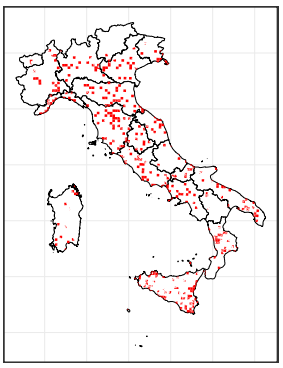
```
          fill='red', colour = NA, show.legend=F) +
  theme_bw() + theme(axis.text.x=element_blank(),
                     axis.text.y=element_blank(),
                     axis.ticks=element_blank(),
                     axis.title.x=element_blank(),
                     axis.title.y=element_blank()),
ggplot() +
  geom_sf(data=region_2013.sf, colour = 'black', size = .05,
          fill = 'white') +
  labs(caption = '2013-03-02') +
  geom_sf(data=ita_grid_plot3[ita_grid_plot3$act == TRUE,],
          fill='red', colour = NA, show.legend=F) +
  theme_bw() + theme(axis.text.x=element_blank(),
                     axis.text.y=element_blank(),
                     axis.ticks=element_blank(),
                     axis.title.x=element_blank(),
                     axis.title.y=element_blank()),
ggplot() +
  geom_sf(data=region_2013.sf, colour = 'black', size = .1,
          fill = 'white') +
  labs(caption = '2015-02-28') +
  geom_sf(data=ita_grid_plot4[ita_grid_plot3$act == TRUE,],
          fill='red', colour = NA, show.legend=F) +
  theme_bw() + theme(axis.text.x=element_blank(),
                     axis.text.y=element_blank(),
                     axis.ticks=element_blank(),
                     axis.title.x=element_blank(),
                     axis.title.y=element_blank()),
ggplot() +
  labs(caption = 'Population density (2011 census)') +
  geom_sf(data=geostat_grid.sf, aes(fill=cut), colour = NA, show.legend=F) +
  scale_fill_manual(values = c('white',brewer.pal(9, "YlGn"))) +
  theme_bw() + theme(axis.text.x=element_blank(),
                     axis.text.y=element_blank(),
                     axis.ticks=element_blank(),
                     axis.title.x=element_blank(),
                     axis.title.y=element_blank()),
ncol = 3)
```



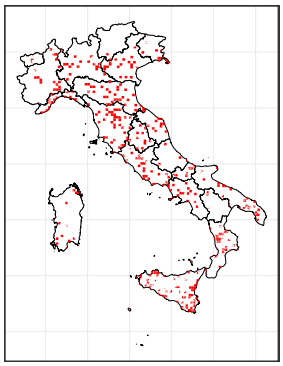
2007−09−15



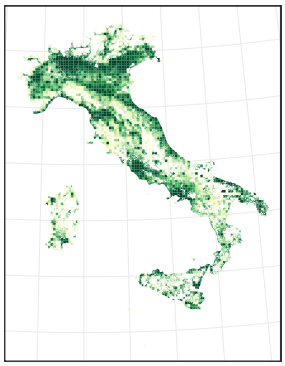
2012−11−03



2013−03−02



2015−02−28



Population density (2011 census)

```
pdf(file = "figure/map-ita-pop-act.pdf", width = 10, height = 20)
grid.arrange(
  ggplot() +
    geom_sf(data=region_2013.sf, colour = 'black', size = .05, fill = 'white') +
    labs(caption = '2007-09-15') +
    geom_sf(data=ita_grid_plot1[ita_grid_plot1$act == TRUE,],
            fill='red', colour = NA, show.legend=F) +
    theme_bw() + theme(axis.text.x=element_blank(),
                       axis.text.y=element_blank(),
                       axis.ticks=element_blank(),
                       axis.title.x=element_blank(),
                       axis.title.y=element_blank(),
                       text = element_text(family = "Palatino")),
  ggplot() +
    geom_sf(data=region_2013.sf, colour = 'black', size = .05, fill = 'white') +
    labs(caption = '2012-11-03') +
    geom_sf(data=ita_grid_plot2[ita_grid_plot2$act == TRUE,],
            fill='red', colour = NA, show.legend=F) +
    theme_bw() + theme(axis.text.x=element_blank(),
                       axis.text.y=element_blank(),
                       axis.ticks=element_blank(),
                       axis.title.x=element_blank(),
                       axis.title.y=element_blank(),
                       text = element_text(family = "Palatino")),
  ggplot() +
    geom_sf(data=region_2013.sf, colour = 'black', size = .05, fill = 'white') +
    labs(caption = '2013-03-02') +
```

```
    geom_sf(data=ita_grid_plot3[ita_grid_plot3$act == TRUE,],
            fill='red', colour = NA, show.legend=F) +
    theme_bw() + theme(axis.text.x=element_blank(),
                       axis.text.y=element_blank(),
                       axis.ticks=element_blank(),
                       axis.title.x=element_blank(),
                       axis.title.y=element_blank(),
                       text = element_text(family = "Palatino")),
  ggplot() +
    geom_sf(data=region_2013.sf, colour = 'black', size = .1, fill = 'white') +
    labs(caption = '2015-02-28') +
    geom_sf(data=ita_grid_plot4[ita_grid_plot3$act == TRUE,],
            fill='red', colour = NA, show.legend=F) +
    theme_bw() + theme(axis.text.x=element_blank(),
                       axis.text.y=element_blank(),
                       axis.ticks=element_blank(),
                       axis.title.x=element_blank(),
                       axis.title.y=element_blank(),
                       text = element_text(family = "Palatino")),
  ggplot() +
    labs(caption = 'Population density (2011 census)') +
    geom_sf(data=geostat_grid.sf, aes(fill=cut), colour = NA, show.legend=F) +
    scale_fill_manual(values = c('white',brewer.pal(9, "YlGn"))) +
    theme_bw() + theme(axis.text.x=element_blank(),
                       axis.text.y=element_blank(),
                       axis.ticks=element_blank(),
                       axis.title.x=element_blank(),
                       axis.title.y=element_blank(),
                       text = element_text(family = "Palatino")),
  ncol = 2)
dev.off()
```

```
## pdf
##   2
```

Analysis of recruitment and Post-recruitment networks

To replicate the following chunks:

```
R -e 'setwd("/path/to/directory");
source("replication_code_06_measure_geographic_diffusion.R");'
```

```
read_chunk('replication_code_07_network_density.R')
```

```
load("replication_file_04_friendship_graph.RData")
load("replication_file_06_recruitment_graph.RData")
load('replication_file_07_geostat_grid.RData')

library(data.table)
library(parallel)
library(igraph)
library(sp)
library(sf)
library(rgdal)
```

```r
library(dplyr)

region_2013.sp <- readOGR('.', 'Reg01012013_g_WGS84')
region_2013.sp <-
  spTransform(region_2013.sp, "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
region_2013.sp$COD_REG <- as.numeric(as.character(region_2013.sp$COD_REG))

region_2013.sf <- st_as_sf(region_2013.sp)
region_south_2013.sf <- region_2013.sf %>% filter(COD_REG >= 13)

lonlat_node_df <- data.frame(lon = V(g_recruitment)$lon,
                             lat = V(g_recruitment)$lat)
lonlat_node.sp <-
  SpatialPoints(lonlat_node_df,
                proj4string=CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))
res <- over(lonlat_node.sp, region_2013.sp)

V(g_recruitment)$DEN_REG <- as.character(res$DEN_REG)
V(g_recruitment)$COD_REG <- res$COD_REG
V(g_recruitment)$north_south <-
  ifelse(V(g_recruitment)$COD_REG < 13, "North", "South")

sequence <-
  seq(from = min(as.Date(V(g_recruitment)$joined_mu)), t
      o = as.Date(max(V(g_recruitment)$joined_mu)),
      by = 7)

geostat_grid.sp <-
  as(geostat_grid_simp.sf,'Spatial')
geostat_grid.sp <-
  spTransform(geostat_grid.sp,
              '+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0')

# Italy only
g_recruitment <-
  g_recruitment - V(g_recruitment)[is.na(V(g_recruitment)$north_south)]
lonlat_node_df <-
  data.frame(lon = V(g_recruitment)$lon,
             lat = V(g_recruitment)$lat)
lonlat_node.sp <-
  SpatialPoints(lonlat_node_df,
                proj4string=CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))

lonlat_north_node.df <-
  data.frame(lon = V(g_recruitment)$lon[V(g_recruitment)$north_south == "North"],
             lat = V(g_recruitment)$lat[V(g_recruitment)$north_south == "North"])
lonlat_north_node.sp <-
  SpatialPoints(lonlat_north_node.df,
                proj4string=CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))

lonlat_south_node.df <-
  data.frame(lon = V(g_recruitment)$lon[V(g_recruitment)$north_south == "South"],
             lat = V(g_recruitment)$lat[V(g_recruitment)$north_south == "South"])
```

```
lonlat_south_node.sp <-
  SpatialPoints(as.matrix(lonlat_south_node.df),
                proj4string=CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))

geostat_grid.sp <-
  as(geostat_grid_simp.sf,'Spatial')
geostat_grid.sp <-
  spTransform(geostat_grid.sp,
              '+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0')

res <- over(geostat_grid.sp, region_2013.sp)
geostat_grid.sp$COD_REG <- res$COD_REG
geostat_grid.sp$DEN_REG <- as.character(res$DEN_REG)

tot_pop <- sum(geostat_grid.sp$pop)
tot_north_pop <- sum(geostat_grid.sp$pop[geostat_grid.sp$COD_REG < 13], na.rm = T)
tot_south_pop <- sum(geostat_grid.sp$pop[geostat_grid.sp$COD_REG >= 13], na.rm = T)

## Tot
res <- over(lonlat_node.sp, geostat_grid.sp)
res <- as.data.frame(table(res$GRID_ID_simp))
geostat_grid.sp$n_nodes <- res$Freq[match(geostat_grid.sp$GRID_ID_simp, res$Var1)]
geostat_grid.sp$n_nodes[is.na(geostat_grid.sp$n_nodes)] <- 0

## North
res <- over(lonlat_north_node.sp, geostat_grid.sp)
res <- as.data.frame(table(res$GRID_ID_simp))
geostat_grid.sp$n_north_nodes <- res$Freq[match(geostat_grid.sp$GRID_ID_simp, res$Var1)]
geostat_grid.sp$n_north_nodes[is.na(geostat_grid.sp$n_north_nodes)] <- 0

## South
res <- over(lonlat_south_node.sp, geostat_grid.sp)
res <- as.data.frame(table(res$GRID_ID_simp))
geostat_grid.sp$n_south_nodes <- res$Freq[match(geostat_grid.sp$GRID_ID_simp, res$Var1)]
geostat_grid.sp$n_south_nodes[is.na(geostat_grid.sp$n_south_nodes)] <- 0

# Tot node
tot_max_act_cells <- sum(geostat_grid.sp$n_nodes > 0)
north_max_act_cells <- sum(geostat_grid.sp$n_north_nodes > 0)
south_max_act_cells <- sum(geostat_grid.sp$n_south_nodes > 0)

getPnt <- function(g) {
  require(igraph)
  lonlat_node.df <-
    data.frame(lon = V(g)$lon,
               lat = V(g)$lat)
  lonlat_node.sp <-
    SpatialPoints(lonlat_node.df,
                  proj4string=CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))
  return(lonlat_node.sp)
}

computeNetStats <- function(i) {
```

```
require(igraph)
require(sp)
print(i)
this_g <-
  g_recruitment - V(g_recruitment)[as.Date(V(g_recruitment)$joined_mu) >= sequence[i]]

this_g <- this_g -  V(this_g)[is.na(V(this_g)$north_south)]
this_g_north <- this_g - V(this_g)[V(this_g)$north_south == 'South']
this_g_south <- this_g - V(this_g)[V(this_g)$north_south == 'North']

res <- over(getPnt(this_g), geostat_grid.sp)
res <- unique(res[,c('GRID_ID_simp','pop')])
this_act_cells <- nrow(res)
this_act_pop <- sum(res$pop, na.rm = T)

if (vcount(this_g_north) > 0) {
  res <- over(getPnt(this_g_north), geostat_grid.sp)
  res <- unique(res[,c('GRID_ID_simp','pop')])
  this_north_act_cells <- nrow(res)
  this_north_act_pop <- sum(res$pop, na.rm = T)
} else {
  this_north_act_cells <- 0
  this_north_act_pop <- 0
}

res <- over(getPnt(this_g_south), geostat_grid.sp)
res <- unique(res[,c('GRID_ID_simp','pop')])
this_south_act_cells <- nrow(res)
this_south_act_pop <- sum(res$pop, na.rm = T)

density_cum_df <-
  data.frame(time = sequence[i],
             density = edge_density(this_g),
             n = vcount(this_g),
             act_cells =  this_act_cells / tot_max_act_cells,
             act_pop = this_act_pop / tot_pop,
             scope = 'all')
density_cum_df <-
  rbind(density_cum_df,
        data.frame(time = sequence[i],
                   density = edge_density(this_g_north),
                   n = vcount(this_g_north),
                   act_cells = this_north_act_cells / north_max_act_cells,
                   act_pop = this_north_act_pop / tot_north_pop,
                   scope = 'north'))
density_cum_df <-
  rbind(density_cum_df,
        data.frame(time = sequence[i],
                   density = edge_density(this_g_south),
                   n = vcount(this_g_south),
                   act_cells = this_south_act_cells / south_max_act_cells,
                   act_pop = this_south_act_pop / tot_south_pop,
                   scope = 'south'))
```

```
  return(density_cum_df)
}

no_cores <- detectCores() - 1
cl <- makeCluster(no_cores, type="FORK")

density_cum_dt <- rbindlist(parLapply(cl, 2:length(sequence), computeNetStats))
stopCluster(cl)

save(density_cum_dt, file = "replication_file_09_network_density.RData")
```

```
load("replication_file_04_friendship_graph.RData")
load("replication_file_06_recruitment_graph.RData")

region_south_2013.sf <- region_2013.sf %>% filter(COD_REG >= 13)

lonlat_node_df <- data.frame(lon = V(g_recruitment)$lon,
                             lat = V(g_recruitment)$lat)
lonlat_node.sp <-
  SpatialPoints(lonlat_node_df,
                proj4string=CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))
res <- over(lonlat_node.sp, region_2013.sp)

V(g_recruitment)$DEN_REG <- as.character(res$DEN_REG)
V(g_recruitment)$COD_REG <- res$COD_REG
V(g_recruitment)$north_south <-
  ifelse(V(g_recruitment)$COD_REG < 13, "North", "South")
```

Recruitment density

```
indegree_north <-
  igraph::degree(g_recruitment -
                   V(g_recruitment)[as.Date(V(g_recruitment)$joined_mu) >
                                      as.Date('2012-10-28') |
                                      V(g_recruitment)$north_south %in% "South"],
                 mode = 'in')
indegree_south <-
  igraph::degree(g_recruitment -
                   V(g_recruitment)[as.Date(V(g_recruitment)$joined_mu) >
                                      as.Date('2012-10-28') |
                                      V(g_recruitment)$north_south %in% "North"],
                 mode = 'in')
indegree_test_df <- data.frame(degree = c(indegree_north, indegree_south),
                               where = c(rep('North',length(indegree_north)),
                                         rep('South',length(indegree_south))))

t.test(degree ~ where, data = indegree_test_df, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  degree by where
## t = 1.827, df = 2347.4, p-value = 0.06783
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.01436314  0.40596479
## sample estimates:
## mean in group North mean in group South
##           0.8032305           0.6074297
```

Post-recruitment density

```
postrecruitment_el <- as.data.frame(get.edgelist(g))
postrecruitment_el$since <- as.Date(paste0("01 ", E(g)$since), "%d %B %Y")
postrecruitment_el$joined_A <- as.Date(V(g)$joined_mu[match(postrecruitment_el$V1, V(g)$name)])
postrecruitment_el$joined_B <- as.Date(V(g)$joined_mu[match(postrecruitment_el$V2, V(g)$name)])
postrecruitment_el <- subset(postrecruitment_el, since > joined_A & since > joined_B)
g_postrecruitment <- graph_from_data_frame(postrecruitment_el[,1:3], directed = F)
missing_v <- V(g)$name[! V(g)$name %in% V(g_postrecruitment)$name]
g_postrecruitment <-
  g_postrecruitment %>% add_vertices(nv = length(missing_v), attr = list(name = missing_v))
V(g_postrecruitment)$joined_mu <- V(g)$joined_mu[match(V(g_postrecruitment)$name, V(g)$name)]
V(g_postrecruitment)$COD_REG <-
  V(g_recruitment)$COD_REG[match(V(g_postrecruitment)$name, V(g_recruitment)$name)]

postrecruiment_density_df <- data.frame()
sequence <-
  seq(from = min(as.Date(meetup_members_2015$joined)),
      max(as.Date(meetup_members_2015$joined)), by = 30)
for (i in 1:length(sequence)) {

  this_g <-
    g_postrecruitment -
    V(g_postrecruitment)[as.Date(V(g_postrecruitment)$joined_mu) >= sequence[i]]

  this_g_north <- this_g - V(this_g)[V(this_g)$COD_REG %in% 13:20]
  this_g_south <- this_g - V(this_g)[V(this_g)$COD_REG %in% 1:12]

  postrecruiment_density_df <-
    rbind(postrecruiment_density_df, data.frame(time = sequence[i],
                                                density = c(edge_density(this_g_north),
                                                            edge_density(this_g_south)),
                                                where = c("North", "South")))
}


degree_north <-
  igraph::degree(g_postrecruitment -
                   V(g_postrecruitment)[as.Date(V(g_postrecruitment)$joined_mu) >
                                          as.Date('2012-10-28') |
                                          V(g_postrecruitment)$COD_REG %in% 13:20],
                 mode = 'all')
degree_south <-
  igraph::degree(g_postrecruitment -
                   V(g_postrecruitment)[as.Date(V(g_postrecruitment)$joined_mu) >
                                          as.Date('2012-10-28') |
                                          V(g_postrecruitment)$COD_REG %in% 1:12],
                 mode = 'all')
```

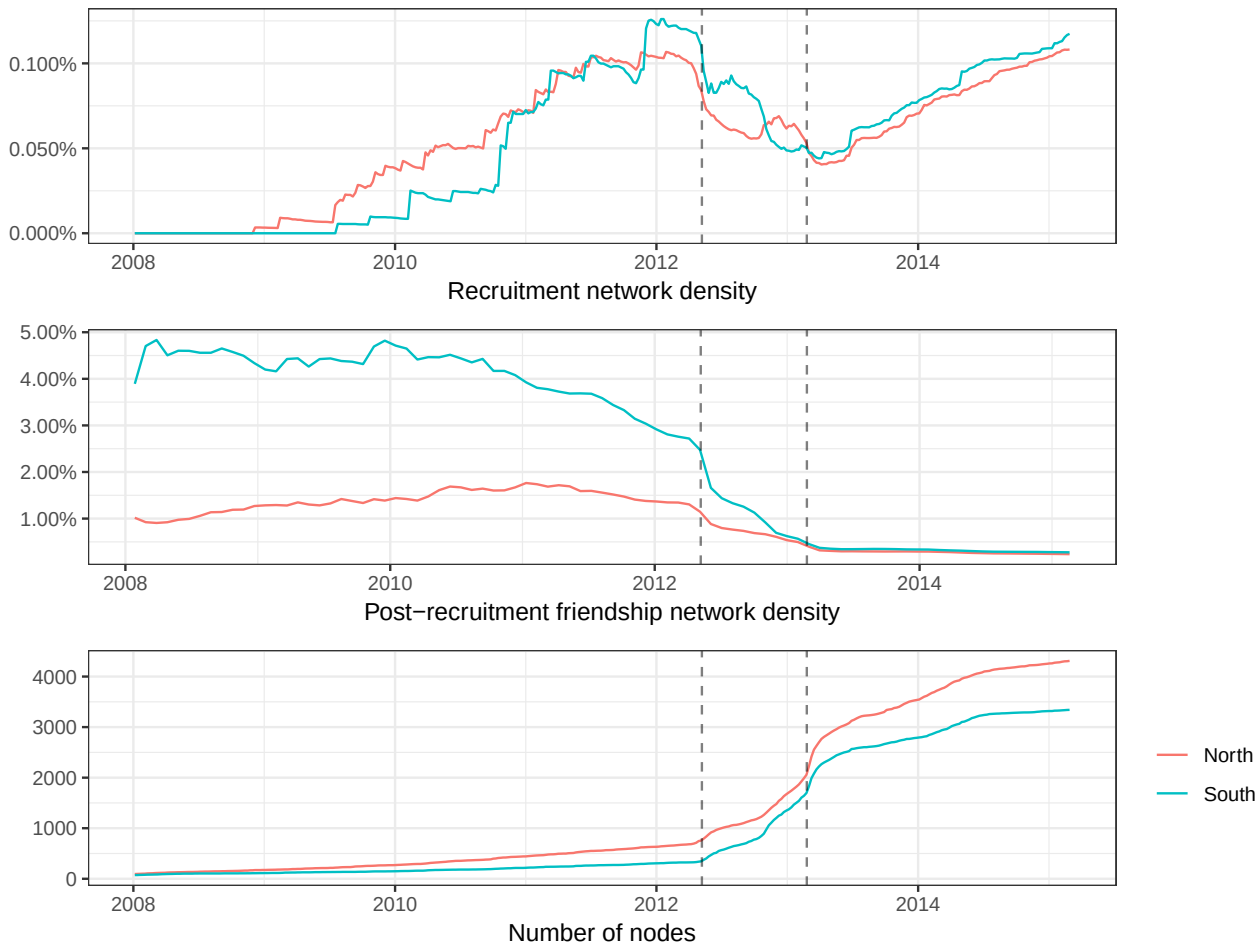
```r
degree_test_df <- data.frame(degree = c(degree_north, degree_south),
                             where = c(rep('North',length(degree_north)),
                                       rep('South',length(degree_south))))

t.test(degree ~ where, data = degree_test_df, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  degree by where
## t = -1.0779, df = 2081.7, p-value = 0.2812
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.1332163  0.6199386
## sample estimates:
## mean in group North mean in group South
##            9.004405            9.761044
```

Figure 8

```r
load("replication_file_09_network_density.RData")

p1 <-
  ggplot(density_cum_dt[density_cum_dt$scope!='all' &
                          density_cum_dt$time > as.Date("2008-01-01"),],
         aes(x=time, y=density, colour=scope)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date(c("2012-05-07", "2013-02-24"))),
             linetype = 2, alpha = .5) +
  labs(y = NULL, x="Recruitment network density") +
  scale_y_continuous(labels = percent) + guides(colour=FALSE) + theme_bw()

p2 <-
  ggplot(postrecruiment_density_df[postrecruiment_density_df$time >
                                     as.Date("2008-01-01"),],
         aes(x=time, y=density, colour=where)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date(c("2012-05-07", "2013-02-24"))),
             linetype = 2, alpha = .5) +
  labs(y = NULL, x="Post-recruitment friendship network density") +
  scale_y_continuous(labels = percent) + guides(colour=FALSE) + theme_bw()

density_cum_dt$label <- ifelse(density_cum_dt$scope == "north", "North", "South")

p3 <-
  ggplot(density_cum_dt[density_cum_dt$scope!='all' &
                          density_cum_dt$time > as.Date("2008-01-01"),],
         aes(x=time, y=n, colour=label)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date(c("2012-05-07", "2013-02-24"))),
             linetype = 2, alpha = .5) +
  labs(y = NULL, x="Number of nodes", colour = NULL) +
```

```
  theme(legend.position = 'bottom') + theme_bw()

egg::ggarrange(p1, p2, p3, ncol = 1)
```




```
p1 <-
  ggplot(density_cum_dt[density_cum_dt$scope!='all' &
                          density_cum_dt$time > as.Date("2008-01-01"),],
         aes(x=time, y=density, colour=scope)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date(c("2012-05-07", "2013-02-24"))),
             linetype = 2, alpha = .5) +
  labs(y = NULL, x="Recruitment network density") +
  scale_y_continuous(labels = percent) + guides(colour=FALSE) +
  theme_bw() +
  theme(legend.position = 'bottom', text = element_text(family = "Palatino"))

p2 <-
  ggplot(postrecruiment_density_df[postrecruiment_density_df$time >
                                     as.Date("2008-01-01"),],
         aes(x=time, y=density, colour=where)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date(c("2012-05-07", "2013-02-24"))),
             linetype = 2, alpha = .5) +
```

```
  labs(y = NULL, x="Post-recruitment friendship network density") +
  scale_y_continuous(labels = percent) + guides(colour=FALSE) +
  theme_bw() +
  theme(legend.position = 'bottom', text = element_text(family = "Palatino"))

density_cum_dt$label <- ifelse(density_cum_dt$scope == "north", "North", "South")

p3 <-
  ggplot(density_cum_dt[density_cum_dt$scope!='all' & density_cum_dt$time >
                          as.Date("2008-01-01"),],
         aes(x=time, y=n, colour=label)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date(c("2012-05-07", "2013-02-24"))),
             linetype = 2, alpha = .5) +
  labs(y = NULL, x="Number of nodes", colour = NULL) +
  theme_bw() +
  theme(legend.position = 'bottom', text = element_text(family = "Palatino"))

pdf(file = "figure/ts-net-density.pdf", width = 8, height = 6)
egg::ggarrange(p1, p2, p3, ncol = 1)
dev.off()
```

```
## pdf
##   2
```

Exponential Random Graph Models

Create control variable `pre_active_buff_20km`

```
read_chunk('replication_code_08_buffer_density.R')
```

```
load("replication_file_04_friendship_graph.RData")
load("replication_file_06_recruitment_graph.RData")

library(parallel)
library(data.table)
library(igraph)
library(sp)
library(sf)
library(rgdal)
library(dplyr)
library(DBI)
library(RSQLite)

# Functions
sqliteGetTable <- function(database, table) {
  require(DBI)
  require(RSQLite)
  con <- dbConnect(RSQLite::SQLite(), dbname = database)
  query <- dbSendQuery(con, paste("SELECT * FROM ", table, ";", sep=""))
  result <- fetch(query, n = -1)
  dbClearResult(query)
  dbDisconnect(con)
```

```
  return(result)
}

region_2013.sp <- readOGR('.', 'Reg01012013_g_WGS84')
region_2013.sp <-
  spTransform(region_2013.sp, "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
region_2013.sp$COD_REG <- as.numeric(as.character(region_2013.sp$COD_REG))

region_2013.sf <- st_as_sf(region_2013.sp)
region_south_2013.sf <- region_2013.sf %>% filter(COD_REG >= 13)

lonlat_node_df <- data.frame(lon = V(g_recruitment)$lon,
                             lat = V(g_recruitment)$lat)
lonlat_node.sp <-
  SpatialPoints(lonlat_node_df,
                proj4string=CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))
res <- over(lonlat_node.sp, region_2013.sp)

V(g_recruitment)$DEN_REG <- as.character(res$DEN_REG)
V(g_recruitment)$COD_REG <- res$COD_REG
V(g_recruitment)$north_south <- ifelse(V(g_recruitment)$COD_REG < 13, "North", "South")

# Italy only
g_recruitment <-
  g_recruitment - V(g_recruitment)[is.na(V(g_recruitment)$north_south)]

g_pnts <-
  SpatialPointsDataFrame(cbind(V(g_recruitment)$lon,
                               V(g_recruitment)$lat),
                         proj4string = CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'),
                         data = data.frame(joined_mu = V(g_recruitment)$joined_mu,
                                           name = as.character(V(g_recruitment)$name)))
g_pnts <- spTransform(g_pnts, CRS("+proj=utm +zone=32 +datum=WGS84 +units=m +no_defs"))

meetup_members <- sqliteGetTable('meetup_mar15.sqlite', 'member')
italy_box_y <- c(36, 47.5)
italy_box_x <- c(6.1, 19.5)
meetup_members <- meetup_members[meetup_members$lon >= italy_box_x[1] &
                                 meetup_members$lon <= italy_box_x[2] &
                                 meetup_members$lat >= italy_box_y[1] &
                                 meetup_members$lat <= italy_box_y[2],]
mu_pnts <-
  SpatialPoints(cbind(meetup_members$lon,
                      meetup_members$lat),
                proj4string = CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))
mu_pnts <-
  spTransform(mu_pnts, CRS("+proj=utm +zone=32 +datum=WGS84 +units=m +no_defs"))

countGeoActive <- function(i) {
  this_buff <- rgeos::gBuffer(g_pnts[i,], width = 20000)
  these_pre_pnts <-
    g_pnts[as.POSIXct(g_pnts$joined_mu) < as.POSIXct(g_pnts$joined_mu[i]),]
  if (nrow(these_pre_pnts) == 0) {
```

```
    pre_active_buff_20km <- 0
  } else {
    pre_res <- over(these_pre_pnts, this_buff)
    pre_active_buff_20km <- sum(!is.na(pre_res))
  }
  tot_res <- over(mu_pnts, this_buff)
  return(list(pre_active_buff_20km = pre_active_buff_20km,
              tot_active_buff_20km = sum(!is.na(tot_res)),
              name = g_pnts$name[i]))
}

no_cores <- detectCores() - 1
cl <- makeCluster(no_cores, type="FORK")

res <- parLapply(cl, 1:vcount(g_recruitment), countGeoActive)
net_active_buff_20km_df <-
  data.frame(name = sapply(res, FUN = function(x) x$name),
             pre_active_buff_20km = sapply(res, FUN = function(x) x$pre_active_buff_20km),
             tot_active_buff_20km = sapply(res, FUN = function(x) x$tot_active_buff_20km))

save(net_active_buff_20km_df,
     file  = "replication_file_10_net_active_buff_20km.RData")
```

Create control variable gender

```
read_chunk('replication_code_09_gender_attribution.R')
```

```
italian_names_by_gender <- read.csv("replication_file_11_italian_names_by_gender.csv", header=FALSE)
italian_names_by_gender <- unique(italian_names_by_gender)
italian_names_by_gender <- italian_names_by_gender[
  (!duplicated(italian_names_by_gender$V1) &
     !duplicated(italian_names_by_gender$V1, fromLast = TRUE)),]

sqliteGetTable <- function(database, table) {
  require(DBI)
  require(RSQLite)
  con <- dbConnect(RSQLite::SQLite(), dbname = database)
  query <- dbSendQuery(con, paste("SELECT * FROM ", table, ";", sep=""))
  result <- fetch(query, n = -1)
  dbClearResult(query)
  dbDisconnect(con)
  return(result)
}
meetup_members_2014 <- sqliteGetTable('meetup_aug14.sqlite', 'member')

# Functions
simpleCap <- function(string) {
  s <- strsplit(string, " ")[[1]]
  paste(toupper(substring(s, 1, 1)), substring(s, 2), sep = "", collapse = " ")
}

firstnamesFirst <- function (name, name_dictionary) {
```

```
  name_dictionary <- tolower(as.character(name_dictionary[[1]]))
  name <- as.character(name)
  name <- tolower(name)
  name <- unlist(strsplit(name, " "))

  for (char in name) {
    if (char %in% name_dictionary) {
      return(char)
    }

  }
  return(NA)
}

genderAttribution <- function(string, dictionary) {
  require(stringr)

  # First name first
  string <- firstnamesFirst(string, dictionary)

  # Substring first word (first name)
  string <- stringr::word(string, 1)

  # Remove white spaces
  string <- gsub("-", " ", string)

  # Capitalise first letter to match dictionary
  string <- simpleCap(string)

  # Create vector with matched gender
  return(as.character(dictionary[[2]][match(string, dictionary[[1]])]))
}

require(parallel)
no_cores <- detectCores() - 1
cl <- makeCluster(no_cores)
X <- lsf.str()
clusterExport(cl=cl, varlist=c(as.vector(X)), envir=environment())

assigned_gender_vec <-
  unlist(parLapply(cl, meetup_members_2014$name, genderAttribution, italian_names_by_gender))

stopCluster(cl)

meetup_members_2014_gender <-
  data.frame(member_id = meetup_members_2014$member_id,
             gender = assigned_gender_vec, stringsAsFactors = FALSE)
save(meetup_members_2014_gender,
     file = 'replication_file_12_meetup_members_2014_gender.RData')
```

```
load('replication_file_12_meetup_members_2014_gender.RData')
meetup_members_2014$gender <-
  meetup_members_2014_gender$gender[match(meetup_members_2014$member_id,
                                          meetup_members_2014_gender$member_id)]
```

```
prop.table(table(meetup_members_2014$gender, useNA = 'always'))
```

```
##
##       female         male    undefined         <NA>
## 0.1888393587 0.6724807787 0.0001329135 0.1385469491
```

ERGM

```
load("replication_file_10_net_active_buff_20km.RData")

region_2013.sp$sc_net <-
  survey_reg_avg$sc_net[match(region_2013.sp$COD_REG, survey_reg_avg$COD_REG)]
region_2013.sp$sc_trust <-
  survey_reg_avg$sc_trust[match(region_2013.sp$COD_REG, survey_reg_avg$COD_REG)]

g_recruitment_df <-
  vertexAttributesAsDataFrame(addDegreeToVertices(g_recruitment))
g_recruitment_df$joined_mu_year <-
  format(as.Date(g_recruitment_df$joined_mu), "%Y")
g_recruitment.sp <-
  SpatialPoints(g_recruitment_df[,c('lon','lat')],
                proj4string =
                  CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))

res <-
  over(g_recruitment.sp,
       spTransform(region_2013.sp,
                   '+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs'))
g_recruitment_df <-
  cbind(g_recruitment_df,
        res[,c("sc_net", "sc_trust" )])
g_recruitment_df$pre_active_buff_20km <-
  net_active_buff_20km_df$pre_active_buff_20km[
    match(g_recruitment_df$name, net_active_buff_20km_df$name)]

g_recruitment_df$joined_mu <-
  as.Date(as.character(g_recruitment_df$joined_mu))
g_recruitment_df$joined_mu_num <-
  as.numeric(g_recruitment_df$joined_mu)
g_recruitment_df$gender <-
  meetup_members_2014$gender[match(g_recruitment_df$name,
                                   meetup_members_2014$facebook)]

## Add to network
V(g_recruitment)$sc_net <-
  g_recruitment_df$sc_net[match(g_recruitment_df$name,
                                V(g_recruitment)$name)]
V(g_recruitment)$sc_trust <-
  g_recruitment_df$sc_trust[match(g_recruitment_df$name,
                                  V(g_recruitment)$name)]
V(g_recruitment)$pre_active_buff_20km <-
  g_recruitment_df$pre_active_buff_20km[
```

```
    match(g_recruitment_df$name, V(g_recruitment)$name)]
V(g_recruitment)$gender <-
  g_recruitment_df$gender[match(g_recruitment_df$name,
                                V(g_recruitment)$name)]

# Exclude nodes out of Italy
net_recruitment <-
  asNetwork(g_recruitment -
              V(g_recruitment)[is.na(V(g_recruitment)$COD_REG)])

mod_egrm_recruitment <-
  ergm(net_recruitment ~ edges +
         nodeicov("pre_active_buff_20km") +
         nodeifactor("north_south", base = 1) +
         nodeifactor("gender", base = 1) +
         nodeofactor("gender", base = 1) +
         nodematch("gender") +
         nodeicov("sc_net") + nodeicov("sc_trust")
  )
```

```
## Evaluating log-likelihood at the estimate.
```

```
net_recruitment
```

```
##  Network attributes:
##   vertices = 7654
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 44790
##     missing edges= 0
##     non-missing edges= 44790
##
##  Vertex attribute names:
##     COD_REG DEN_REG gender joined_fb joined_mu lat lon north_south pre_active_buff_20km sc_net sc_tru
##
##  Edge attribute names not shown
```

```
V(g_postrecruitment)$pre_active_buff_20km <-
  V(g_recruitment)$pre_active_buff_20km[
    match(V(g_postrecruitment)$name, V(g_recruitment)$name)]
V(g_postrecruitment)$north_south <-
  V(g_recruitment)$north_south[
    match(V(g_postrecruitment)$name, V(g_recruitment)$name)]
V(g_postrecruitment)$gender <-
  V(g_recruitment)$gender[
    match(V(g_postrecruitment)$name, V(g_recruitment)$name)]
V(g_postrecruitment)$sc_net <-
  V(g_recruitment)$sc_net[
    match(V(g_postrecruitment)$name, V(g_recruitment)$name)]
V(g_postrecruitment)$sc_trust <-
  V(g_recruitment)$sc_trust[
    match(V(g_postrecruitment)$name, V(g_recruitment)$name)]
```

```r
# Exclude nodes out of Italy
net_postrecruitment <- asNetwork(g_postrecruitment - V(g_postrecruitment)[is.na(V(g_postrecruitment)$COD

mod_egrm_postrecruitment <-
  ergm(net_postrecruitment ~ edges +
         nodecov("pre_active_buff_20km") +
         nodefactor("north_south", base = 1) +
         nodefactor("gender") +
         nodematch("gender") +
         nodecov("sc_net") + nodecov("sc_trust")
  )
```

```
## Evaluating log-likelihood at the estimate.
```

```r
net_postrecruitment
```

```
##  Network attributes:
##   vertices = 7654
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 57875
##     missing edges= 0
##     non-missing edges= 57875
##
##  Vertex attribute names:
##     COD_REG gender joined_mu north_south pre_active_buff_20km sc_net sc_trust vertex.names
##
##  Edge attribute names not shown
```

```r
save(g_recruitment, g_postrecruitment, file = 'debug_replication.RData')
```

Table 3

```r
texreg(list(mod_egrm_recruitment, mod_egrm_postrecruitment))
```

```r
texreg(list(mod_egrm_recruitment, mod_egrm_postrecruitment),
       file = "table/ergm-coef.tex",
       custom.model.names = c("Recruitment network", "Post-recruitment network"))
htmlreg(list(mod_egrm_recruitment, mod_egrm_postrecruitment),
        file = "table/ergm-coef.html",
        custom.model.names = c("Recruitment network", "Post-recruitment network"))
```

Logistic regression predicting at least one recruiting tie

```r
g_recruitment_df$joined_mu <- as.Date(g_recruitment_df$joined_mu)
g_recruitment_df$at_least_one_indegree <- g_recruitment_df$indegree > 0

# First phase
prop.table(table(g_recruitment_df[g_recruitment_df$joined_mu <
                                    as.Date("2012-10-28"), ]$at_least_one_indegree,
```

	Model 1	Model 2
edges	-8.17^{***}	-4.48^{***}
	(0.06)	(0.08)
nodeicov.pre_active_buff_20km	0.00^{***}	
	(0.00)	
nodeifactor.north_south.South	0.17^{***}	
	(0.02)	
nodeifactor.gender.male	-0.05^{***}	
	(0.02)	
nodeofactor.gender.male	-0.06^{***}	
	(0.02)	
nodematch.gender	-0.18^{***}	-0.22^{***}
	(0.02)	(0.01)
nodeicov.sc_net	0.80^{***}	
	(0.18)	
nodeicov.sc_trust	3.23^{***}	
	(0.24)	
nodecov.pre_active_buff_20km		-0.00^{***}
		(0.00)
nodefactor.north_south.South		-0.20^{***}
		(0.01)
nodefactor.gender.male		-0.05^{***}
		(0.01)
nodecov.sc_net		-0.24^{*}
		(0.11)
nodecov.sc_trust		-3.06^{***}
		(0.15)
AIC	724596.58	834853.44
BIC	724723.66	834959.79
Log Likelihood	-362290.29	-417419.72

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 2: Statistical models

```
                g_recruitment_df[g_recruitment_df$joined_mu <
                                   as.Date("2012-10-28"), ]$north_south),
          2)
```

```
##
##             North     South
##   FALSE 0.7705696 0.7552836
##   TRUE  0.2294304 0.2447164
```

```
mean(g_recruitment_df[g_recruitment_df$joined_mu <
                        as.Date("2012-10-28"), ]$at_least_one_indegree)
```

```
## [1] 0.2320319
```

```
# Second phase
prop.table(table(g_recruitment_df[g_recruitment_df$joined_mu >=
                                   as.Date("2012-10-28") &
                                   g_recruitment_df$joined_mu <
                                   as.Date("2013-02-24"), ]$at_least_one_indegree,
                g_recruitment_df[g_recruitment_df$joined_mu >=
                                   as.Date("2012-10-28") &
                                   g_recruitment_df$joined_mu <
                                   as.Date("2013-02-24"), ]$north_south),
          2)
```

```
##
##             North     South
##   FALSE 0.6567164 0.6394052
##   TRUE  0.3432836 0.3605948
```

```
mean(g_recruitment_df[g_recruitment_df$joined_mu >=
                        as.Date("2012-10-28") &
                        g_recruitment_df$joined_mu <
                        as.Date("2013-02-24"), ]$at_least_one_indegree)
```

```
## [1] 0.3459119
```

```
# Third phase
prop.table(table(g_recruitment_df[g_recruitment_df$joined_mu >=
                                   as.Date("2013-02-24"), ]$at_least_one_indegree,
                g_recruitment_df[g_recruitment_df$joined_mu >=
                                   as.Date("2013-02-24"), ]$north_south),
          2)
```

```
##
##             North     South
##   FALSE 0.3678110 0.3365913
##   TRUE  0.6321890 0.6634087
```

```
mean(g_recruitment_df[g_recruitment_df$joined_mu >=
                        as.Date("2013-02-24"), ]$at_least_one_indegree)
```

```
## [1] 0.6480988
```

```
log_mod1 <-
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south + gender + pre_active_buff_20km,
      data = g_recruitment_df[g_recruitment_df$joined_mu <
```

```
                                as.Date("2012-10-28"), ],
      family = 'binomial')

log_mod2 <-
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south + gender + pre_active_buff_20km,
      data = g_recruitment_df[g_recruitment_df$joined_mu >=
                                as.Date("2012-10-28") &
                                g_recruitment_df$joined_mu <
                                as.Date("2013-02-24"), ],
      family = 'binomial')

log_mod3 <-
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south + gender + pre_active_buff_20km,
      data = g_recruitment_df[g_recruitment_df$joined_mu >=
                                as.Date("2013-02-24"), ],
      family = 'binomial')

log_mod0 <-
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south + gender + pre_active_buff_20km,
      data = g_recruitment_df,
      family = 'binomial')
```

```
newdata <-
  data.frame(gender =
               'male',
             pre_active_buff_20km = mean(g_recruitment_df$pre_active_buff_20km, na.rm = T),
             sc_net =
               c(mean(g_recruitment_df$sc_net[
                 g_recruitment_df$DEN_REG %in% 'Veneto']),
                 mean(g_recruitment_df$sc_net[
                   g_recruitment_df$DEN_REG %in% 'Calabria'])),
             sc_trust =
               c(mean(g_recruitment_df$sc_trust[
                 g_recruitment_df$DEN_REG %in% 'Veneto']),
                 mean(g_recruitment_df$sc_trust[
                   g_recruitment_df$DEN_REG %in% 'Calabria'])),
             north_south = c("North", "South"))

pr1 <-
  predict(log_mod1,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr2 <-
  predict(log_mod2,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr3 <-
  predict(log_mod3,
          newdata = newdata,
```

```
          type="response", se.fit = TRUE)

pr0 <-
  predict(log_mod0,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr_df <-
  data.frame(where = rep(c('Veneto (North)','Calabria (South)'), 4),
             time = c(rep("2005-2012", 2),
                      rep("2012-2013", 2),
                      rep("2013-2015", 2),
                      rep("2005-2015", 2)),
             fit = c(pr1$fit, pr2$fit, pr3$fit, pr0$fit),
             se.fit = c(pr1$se.fit, pr2$se.fit, pr3$se.fit, pr0$se.fit),
             stringsAsFactors = F)

pr_df$time <-
  factor(pr_df$time,
         levels = c("2005-2012", "2012-2013", "2013-2015", "2005-2015"))

pr_df$panel <-
  factor(ifelse(pr_df$time == "2005-2015", "whole", "partial"),
         levels = c("partial","whole"))

ggplot(pr_df, aes(colour = where, y = fit, x = time)) +
  geom_point() +
  geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit), width = .3) +
  theme_bw() +
  labs(x = NULL, y = "Probability node has 1 recruiting tie or more") +
  facet_wrap(~panel, scales = "free_x") + scale_y_continuous(labels = percent)
```

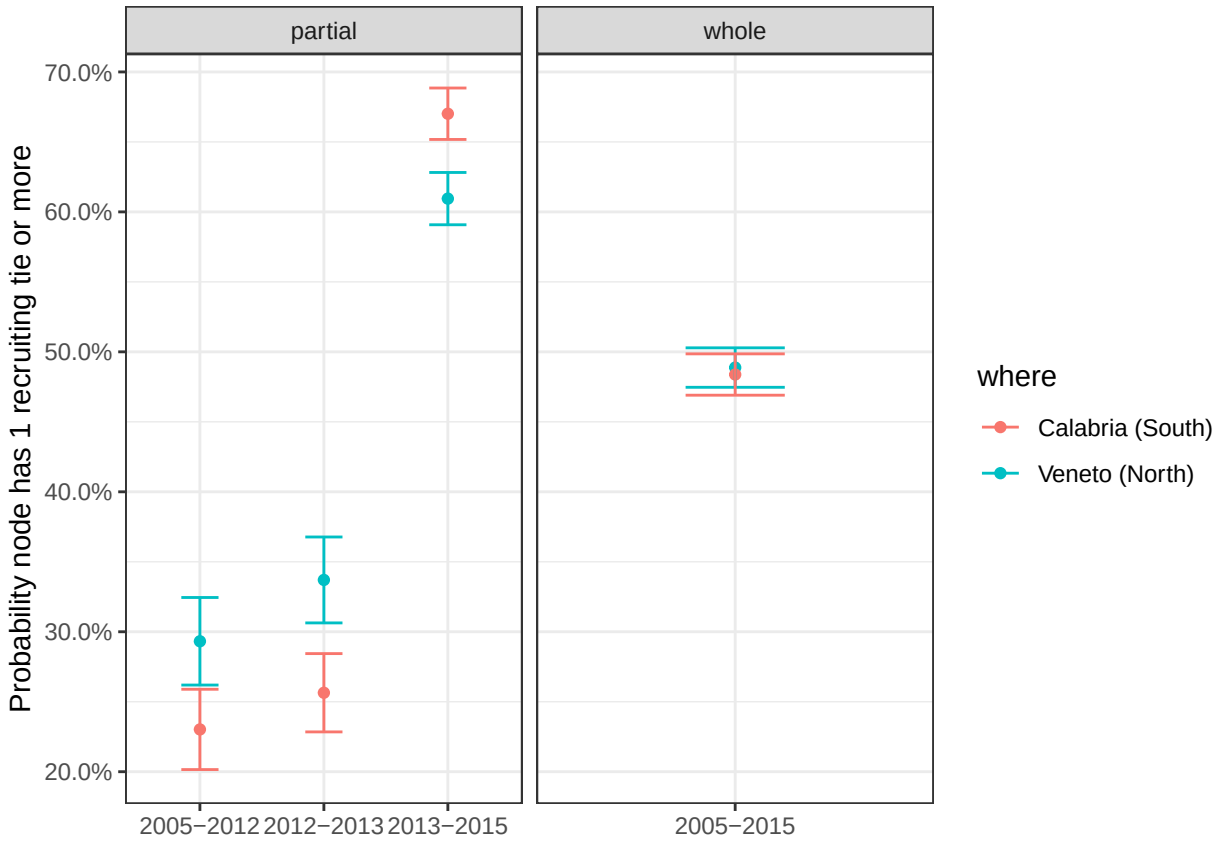


Table 4

```
stargazer(log_mod1, log_mod2, log_mod3, log_mod0,
          column.labels = c("2005-2012", "2012-2013", "2013-2015", "2005-2015"))
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Fri, Oct 19, 2018 - 03:56:34

```
stargazer(log_mod1, log_mod2, log_mod3, log_mod0,
          column.labels = c("2005-2012", "2012-2013", "2013-2015", "2005-2015"),
          type = 'latex', out = 'table/logistic-coef.tex',
          report = "vc*",
          float = FALSE,
          style = 'ajps')
stargazer(log_mod1, log_mod2, log_mod3, log_mod0,
          column.labels = c("2005-2012", "2012-2013", "2013-2015", "2005-2015"),
          type = 'html', out = 'table/logistic-coef.html',
          report = "vc*",
          float = FALSE,
          style = 'ajps')
```

Only sc_net varies

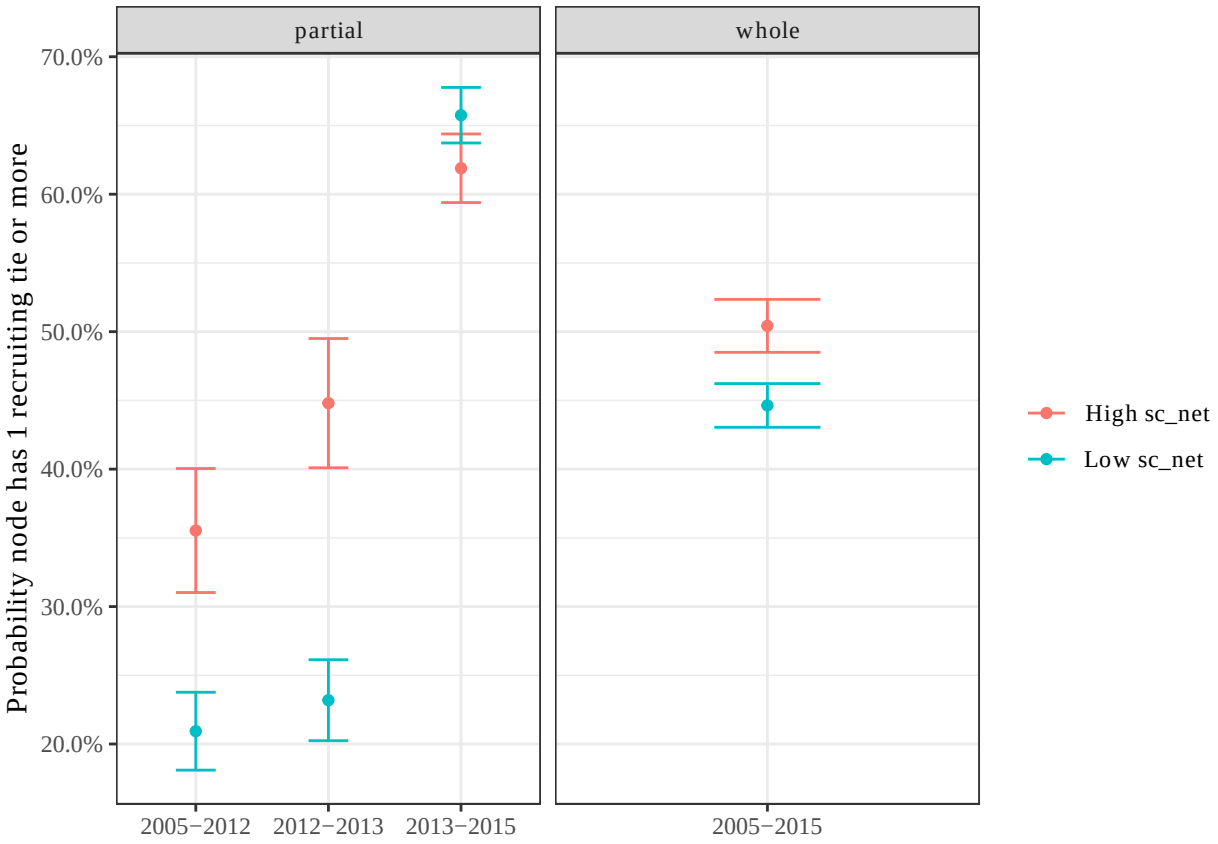
```
g_recruitment_df$north_south_num <- as.numeric(g_recruitment_df$north_south)

log_mod1 <-
```

Table 3:

	Dependent variable:			
	at_least_one_indegree			
	2005-2012	2012-2013	2013-2015	2005-2015
	(1)	(2)	(3)	(4)
sc_net	5.018**	6.769***	−1.145	1.591*
	(2.177)	(2.230)	(1.214)	(0.882)
sc_trust	−4.997	−10.310***	0.523	2.735**
	(3.091)	(3.104)	(1.663)	(1.197)
north_southSouth	0.114	−0.002	0.127	0.372***
	(0.238)	(0.232)	(0.131)	(0.096)
gendermale	−0.427***	−0.271*	−0.059	−0.211***
	(0.138)	(0.149)	(0.091)	(0.065)
pre_active_buff_20km	0.009***	0.0004	0.002***	0.006***
	(0.002)	(0.001)	(0.0005)	(0.0004)
Constant	−0.998	0.380	0.586	−1.249***
	(0.723)	(0.720)	(0.418)	(0.298)
Observations	1,962	1,495	3,640	7,097
Log Likelihood	−1,057.303	−955.112	−2,350.082	−4,779.037
Akaike Inf. Crit.	2,126.607	1,922.224	4,712.165	9,570.074
Note:				*p<0.1; **p<0.05; ***p<0.01

```
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south_num + gender + pre_active_buff_20km,
      data = g_recruitment_df[g_recruitment_df$joined_mu <
                                as.Date("2012-10-28"), ],
      family = 'binomial')

log_mod2 <-
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south_num + gender + pre_active_buff_20km,
      data = g_recruitment_df[g_recruitment_df$joined_mu >=
                                as.Date("2012-10-28") &
                                g_recruitment_df$joined_mu <
                                as.Date("2013-02-24"), ],
      family = 'binomial')

log_mod3 <-
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south_num + gender + pre_active_buff_20km,
      data = g_recruitment_df[g_recruitment_df$joined_mu >=
                                as.Date("2013-02-24"), ],
      family = 'binomial')

log_mod0 <-
  glm(at_least_one_indegree ~ sc_net + sc_trust +
        north_south_num + gender + pre_active_buff_20km,
      data = g_recruitment_df,
      family = 'binomial')

newdata <-
  data.frame(gender =
               'male',
             pre_active_buff_20km = mean(g_recruitment_df$pre_active_buff_20km, na.rm = T),
             sc_net =
               c(mean(g_recruitment_df$sc_net[
                 g_recruitment_df$DEN_REG %in% 'Veneto']),
                 mean(g_recruitment_df$sc_net[
                   g_recruitment_df$DEN_REG %in% 'Calabria'])),
             sc_trust =
               median(g_recruitment_df$sc_trust, na.rm = T),
             north_south_num = 1.5)

pr1 <-
  predict(log_mod1,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr2 <-
  predict(log_mod2,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr3 <-
  predict(log_mod3,
```

```
          newdata = newdata,
          type="response", se.fit = TRUE)

pr0 <-
  predict(log_mod0,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr_df <-
  data.frame(where = rep(c('High sc_net','Low sc_net'), 4),
             time = c(rep("2005-2012", 2),
                      rep("2012-2013", 2),
                      rep("2013-2015", 2),
                      rep("2005-2015", 2)),
             fit = c(pr1$fit, pr2$fit, pr3$fit, pr0$fit),
             se.fit = c(pr1$se.fit, pr2$se.fit, pr3$se.fit, pr0$se.fit),
             stringsAsFactors = F)

pr_df$time <-
  factor(pr_df$time,
         levels = c("2005-2012", "2012-2013", "2013-2015", "2005-2015"))

pr_df$panel <-
  factor(ifelse(pr_df$time == "2005-2015", "whole", "partial"),
         levels = c("partial","whole"))

ggplot(pr_df, aes(colour = where, y = fit, x = time)) +
  geom_point() +
  geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit), width = .3) +
  theme_bw() +
  labs(x = NULL, y = "Probability node has 1 recruiting tie or more", colour = NULL) +
  facet_wrap(~panel, scales = "free_x") + scale_y_continuous(labels = percent) +
  theme(text = element_text(family = "Palatino"))
```

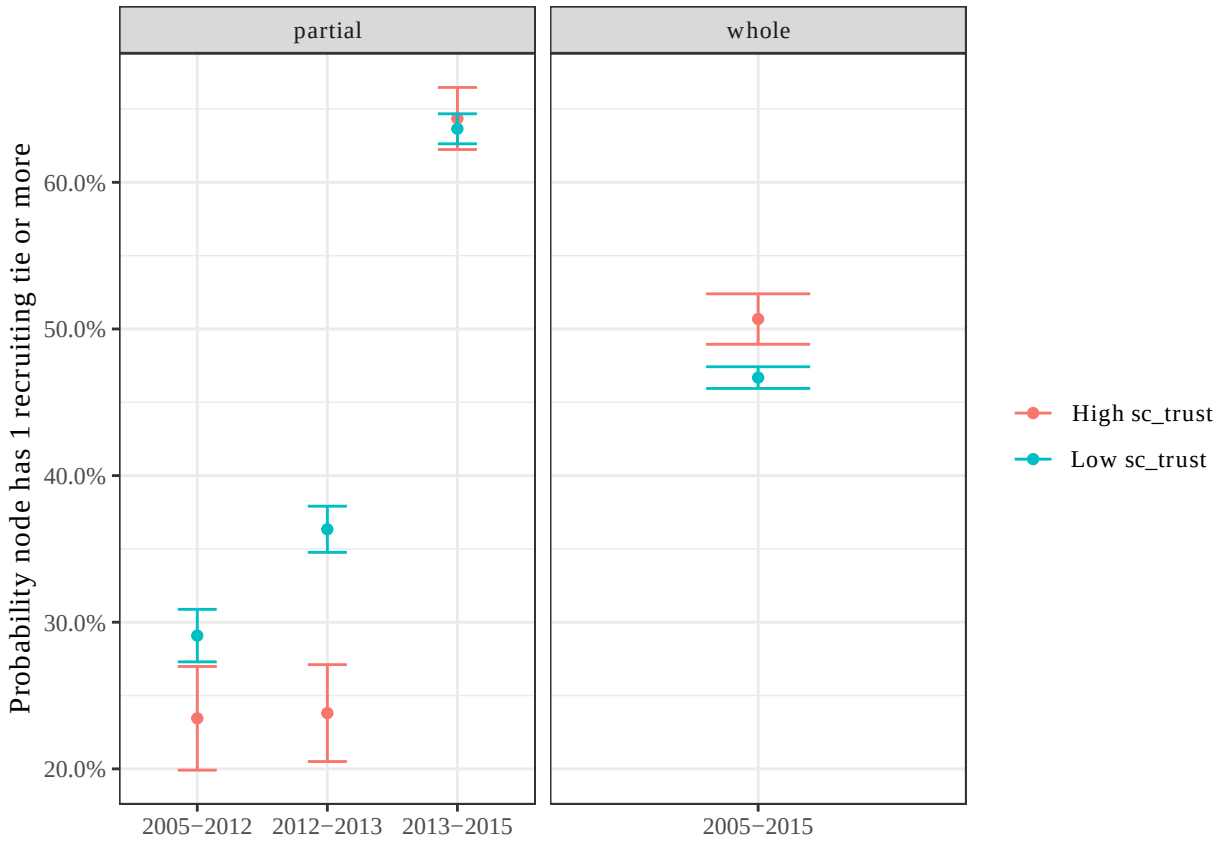



```
pr_df_net <- pr_df
pr_df_net$what <- 'sc_net'
```

Only sc_trust varies

```
newdata <-
  data.frame(gender =
               'male',
             pre_active_buff_20km = mean(g_recruitment_df$pre_active_buff_20km, na.rm = T),
             sc_trust =
               c(mean(g_recruitment_df$sc_trust[
                 g_recruitment_df$DEN_REG %in% 'Veneto']),
                 mean(g_recruitment_df$sc_trust[
                   g_recruitment_df$DEN_REG %in% 'Calabria'])),
             sc_net =
               median(g_recruitment_df$sc_net, na.rm = T),
             north_south_num = 1.5)

pr1 <-
  predict(log_mod1,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr2 <-
  predict(log_mod2,
          newdata = newdata,
          type="response", se.fit = TRUE)
```

```r
pr3 <-
  predict(log_mod3,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr0 <-
  predict(log_mod0,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr_df <-
  data.frame(where = rep(c('High sc_trust','Low sc_trust'), 4),
             time = c(rep("2005-2012", 2),
                      rep("2012-2013", 2),
                      rep("2013-2015", 2),
                      rep("2005-2015", 2)),
             fit = c(pr1$fit, pr2$fit, pr3$fit, pr0$fit),
             se.fit = c(pr1$se.fit, pr2$se.fit, pr3$se.fit, pr0$se.fit),
             stringsAsFactors = F)

pr_df$time <-
  factor(pr_df$time,
         levels = c("2005-2012", "2012-2013", "2013-2015", "2005-2015"))

pr_df$panel <-
  factor(ifelse(pr_df$time == "2005-2015", "whole", "partial"),
         levels = c("partial","whole"))

ggplot(pr_df, aes(colour = where, y = fit, x = time)) +
  geom_point() +
  geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit), width = .3) +
  theme_bw() +
  labs(x = NULL, y = "Probability node has 1 recruiting tie or more", colour = NULL) +
  facet_wrap(~panel, scales = "free_x") + scale_y_continuous(labels = percent) +
  theme(text = element_text(family = "Palatino"))
```



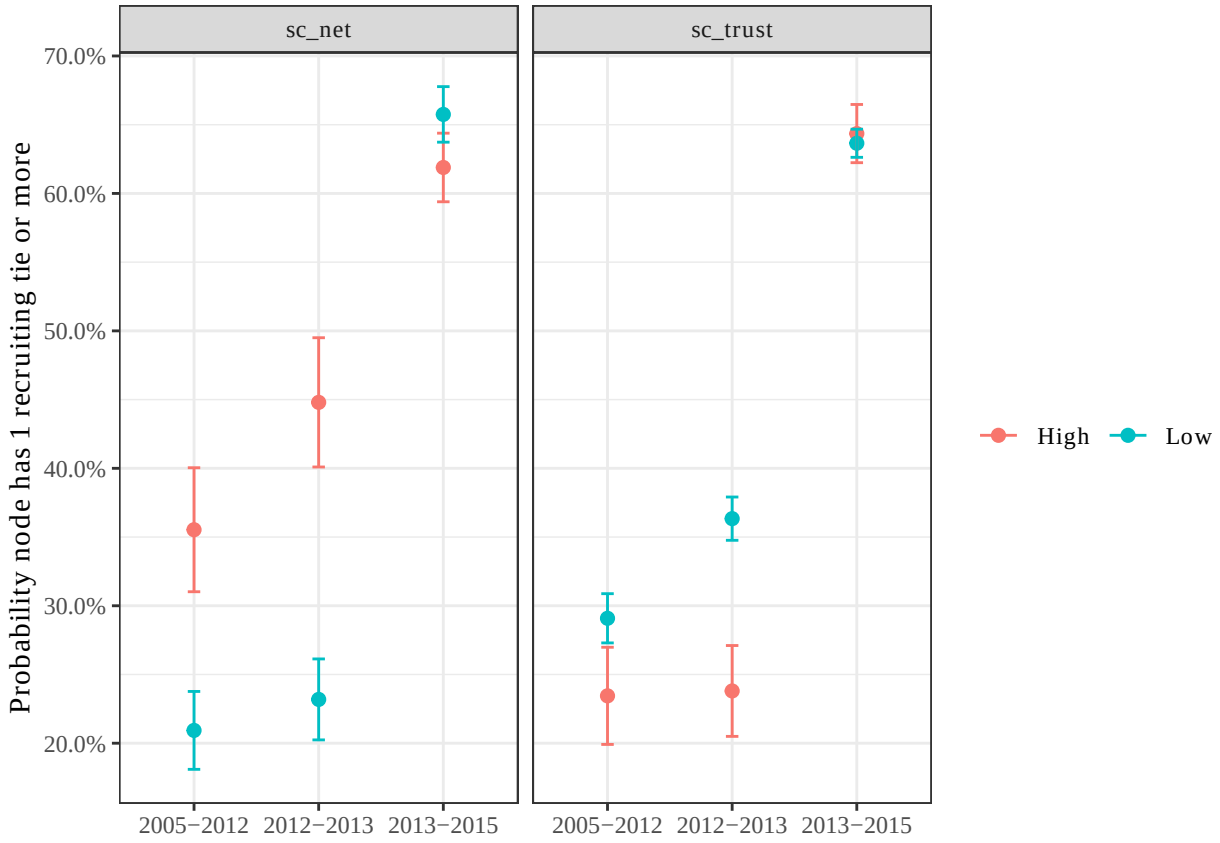

```
pr_df_trust <- pr_df
pr_df_trust$what <- 'sc_trust'
```

Figure 9

```
pr_df <- rbind(pr_df_net, pr_df_trust)

kable(pr_df, digits = 3)
```

where	time	fit	se.fit	panel	what
High sc_net	2005-2012	0.355	0.045	partial	sc_net
Low sc_net	2005-2012	0.209	0.028	partial	sc_net
High sc_net	2012-2013	0.448	0.047	partial	sc_net
Low sc_net	2012-2013	0.232	0.029	partial	sc_net
High sc_net	2013-2015	0.619	0.025	partial	sc_net
Low sc_net	2013-2015	0.657	0.020	partial	sc_net
High sc_net	2005-2015	0.504	0.019	whole	sc_net
Low sc_net	2005-2015	0.446	0.016	whole	sc_net
High sc_trust	2005-2012	0.234	0.035	partial	sc_trust
Low sc_trust	2005-2012	0.291	0.018	partial	sc_trust
High sc_trust	2012-2013	0.238	0.033	partial	sc_trust
Low sc_trust	2012-2013	0.363	0.016	partial	sc_trust
High sc_trust	2013-2015	0.644	0.021	partial	sc_trust
Low sc_trust	2013-2015	0.636	0.010	partial	sc_trust
High sc_trust	2005-2015	0.507	0.017	whole	sc_trust
Low sc_trust	2005-2015	0.467	0.007	whole	sc_trust

```
pr_df$colour <- factor(ifelse(grepl("High", pr_df$where), "High", "Low"), levels = c("High", "Low"))

ggplot(pr_df[pr_df$panel=='partial',], aes(colour = colour, y = fit, x = time)) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit), width = .1) +
  theme_bw() +
  facet_wrap(~what) +
  scale_shape_manual(values = c(2,6)) +
  scale_y_continuous(labels = percent) +
  guides(colour=guide_legend(ncol=2, title=NULL)) +
  labs(x = NULL, y = "Probability node has 1 recruiting tie or more") +
  theme(text = element_text(family = "Palatino"))
```




```
ggsave("figure/logit-predict-from-sc-net-trust.pdf", width = 8, height = 4,
       ggplot(pr_df[pr_df$panel=='partial',], aes(colour = colour, y = fit, x = time)) +
         geom_point(size = 2) +
         geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit), width = .1) +
         theme_bw() +
         facet_wrap(~what) +
         scale_shape_manual(values = c(2,6)) +
         scale_y_continuous(labels = percent) +
         guides(colour=guide_legend(ncol=2, title=NULL)) +
         labs(x = NULL, y = "Probability node has 1 recruiting tie or more") +
         theme(text = element_text(family = "Palatino")))
```

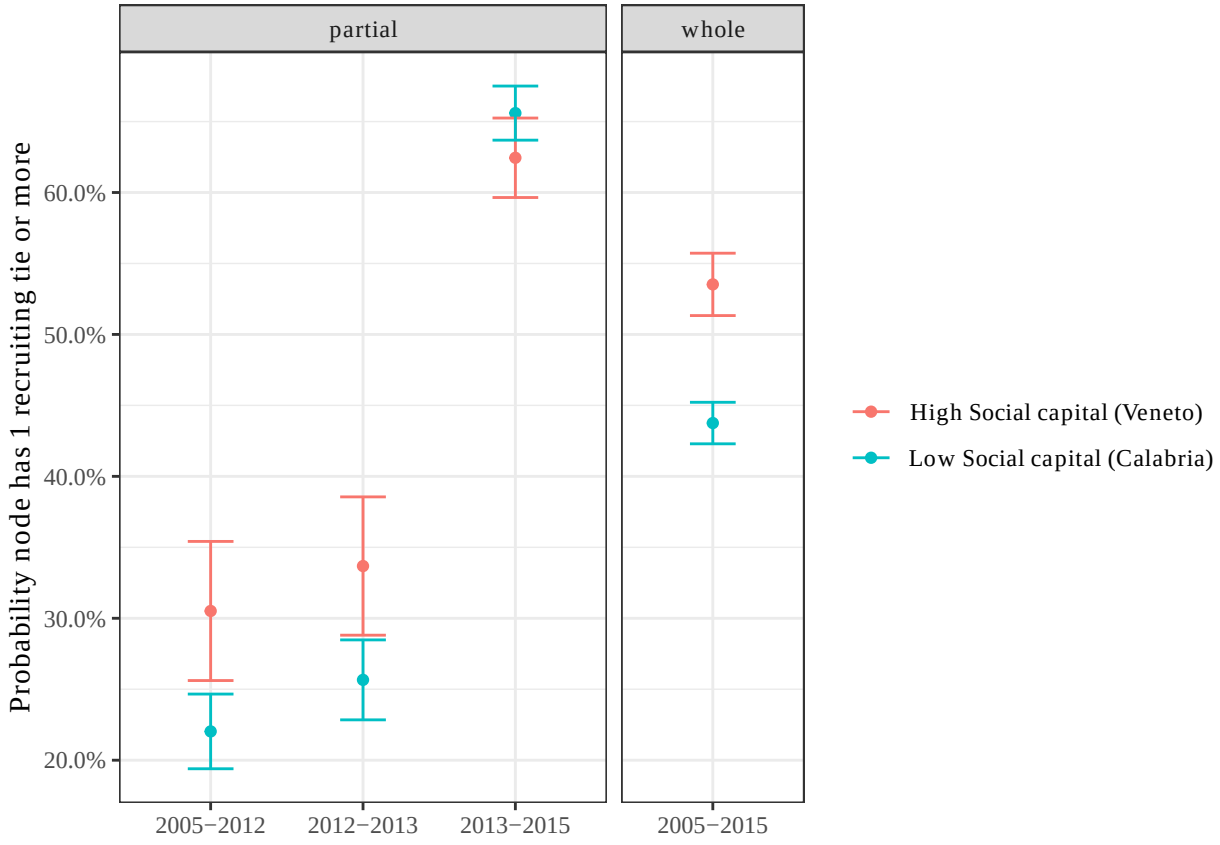
Figure 10

```
newdata <-
  data.frame(gender =
               'male',
             pre_active_buff_20km = mean(g_recruitment_df$pre_active_buff_20km, na.rm = T),
             sc_net =
               c(mean(g_recruitment_df$sc_net[
                 g_recruitment_df$DEN_REG %in% 'Veneto']),
                 mean(g_recruitment_df$sc_net[
                   g_recruitment_df$DEN_REG %in% 'Calabria'])),
             sc_trust =
               c(mean(g_recruitment_df$sc_trust[
                 g_recruitment_df$DEN_REG %in% 'Veneto']),
                 mean(g_recruitment_df$sc_trust[
                   g_recruitment_df$DEN_REG %in% 'Calabria'])),
             north_south_num = 1.5)

pr1 <-
  predict(log_mod1,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr2 <-
  predict(log_mod2,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr3 <-
  predict(log_mod3,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr0 <-
  predict(log_mod0,
          newdata = newdata,
          type="response", se.fit = TRUE)

pr_df <-
  data.frame(where = rep(c('High Social capital (Veneto)','Low Social capital (Calabria)'), 4),
             time = c(rep("2005-2012", 2),
                      rep("2012-2013", 2),
                      rep("2013-2015", 2),
                      rep("2005-2015", 2)),
             fit = c(pr1$fit, pr2$fit, pr3$fit, pr0$fit),
             se.fit = c(pr1$se.fit, pr2$se.fit, pr3$se.fit, pr0$se.fit),
             stringsAsFactors = F)

pr_df$time <-
  factor(pr_df$time,
         levels = c("2005-2012", "2012-2013", "2013-2015", "2005-2015"))

pr_df$panel <-
  factor(ifelse(pr_df$time == "2005-2015", "whole", "partial"),
         levels = c("partial","whole"))
```

```
kable(pr_df, digits = 2)
```

where	time	fit	se.fit	panel
High Social capital (Veneto)	2005-2012	0.31	0.05	partial
Low Social capital (Calabria)	2005-2012	0.22	0.03	partial
High Social capital (Veneto)	2012-2013	0.34	0.05	partial
Low Social capital (Calabria)	2012-2013	0.26	0.03	partial
High Social capital (Veneto)	2013-2015	0.62	0.03	partial
Low Social capital (Calabria)	2013-2015	0.66	0.02	partial
High Social capital (Veneto)	2005-2015	0.54	0.02	whole
Low Social capital (Calabria)	2005-2015	0.44	0.01	whole

```
ggplot(pr_df, aes(colour = where, y = fit, x = time)) +
  geom_point() +
  geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit), width = .3) +
  theme_bw() +
  labs(x = NULL, y = "Probability node has 1 recruiting tie or more", colour = NULL) +
  facet_grid(.~panel, scales = "free_x", space = "free_x") +
  scale_y_continuous(labels = percent) +
  theme(text = element_text(family = "Palatino"))
```




```
ggsave("figure/logit-predict-from-social-capital.pdf", width = 8, height = 4,
       ggplot(pr_df, aes(colour = where, y = fit, x = time)) +
         geom_point() +
         geom_errorbar(aes(ymin = fit - se.fit, ymax = fit + se.fit), width = .3) +
```

```
        theme_bw() +
        labs(x = NULL, y = "Probability node has 1 recruiting tie or more", colour = NULL) +
        facet_grid(.~panel, scales = "free_x", space = "free_x") +
        scale_y_continuous(labels = percent) +
        theme(text = element_text(family = "Palatino")))
```