Credit Allocation Mix in Peer-to-Peer Lending: A Network Study on Bondora

Social Network Analysis - Group 7

Floris Vermeulen   Martijn van Iterson   Niek Fleerakkers

Patryk Grodek   Samir Sabitli

2025-11-16

Table of Contents

1   Introduction

With the advent of online peer-to-peer (P2P) lending platforms, the traditional methods of financial intermediation have been usurped by individual choice. The surging popularity and accessibility of services such as Prosper.com has grown to become one of the most important avenues through which individuals can secure micro-loans (Cai, Lin, Xu, & Fu, 2016). Arguably, while the democratisation of credit alleviates numerous pre-existing issues, it serves to exaggerate well-known informational asymmetries associated with determining creditworthiness (Mingfeng Lin, N.R. Prabhala, & Siva Viswanathan, 2009). Per, Chen, Zhou, & Wan (2016), from a traditional financial lens, these asymmetries can contribute to moral hazard and adverse selection, ultimately causing systemic financial losses. This arises from the fact the users are anonymous, challenging the veracity of their information. Further, the unsecured nature of credit makes its collection relatively ineffective. Jointly, these factors characterise the lemons market theory by Akerlof (1978).

However, social networks can act a crucial avenue through which information transfer could be facilitated, alleviate some of the existing asymmetries (Mingfeng Lin et al., 2009). There are several prominent theories and research that attempt to bridge this gap. First, the theory of social capital is particularly salient because it relates directly to the reputation and trustworthiness of the users (Adler & Kwon, 2002). While its definition is not strictly defined, according to Putnam (2015), it can be seen as a means through which social organisation facilitate coordination and trust. Nahapiet & Ghoshal (1998) define three dimensions, including structural, relational, and cognitive. We primarily study the structural component, whereby the network properties relating to tie formation between borrowers is considered. Following this perspective, Chen et al. (2016) argue that social capital is earned through connections to others. However, per the authors, the internet facilitates weak forms of these ties, diminishing the benefits of social capital on P2P platforms, especially for non-friendship networks. Notably, they cite a low interdependence and closed structures to this effect.

With this, we arrive at our first research problem, aiming to understand **to what extent are borrower networks weak in P2P networks?** More specifically, we formulate the hypothesis

H_1^a Borrower networks based on similarity have a low degree of centralisation

To understand social capital, we employ techniques related to the occurance of dyadic connections rather than studying their structural components. As such, this problem can be studied with the use of the Quadratic Assignment Procedure (QAP) and Conditional Uniform Graph (CUG) tests, whereby the theoretical distributions of network characteristics can be tested.

Our second study attempts to model social capital as a statistical model, specifically considering the structural dimension to social capital. According to the comprehensive review of Bachmann et al. (2011), past studies have considered financial, demographic, and soft informational factors in the financial outcomes of P2P lending, however, these perspectives have not been considered with relation to each other. With respect to this premise, we posit **what is the role of structural and non-structural factors in the formation of social capital?** This problem is arguably best studied with the use of Exponential Random Graph Models (ERGM) as they are considered to work well in identifying a combination of structural and exogenous patterns in the context of a variety of network specifications (Jackson, 2011).

In studying demographic factors, Pope & Sydnor (2008) suggest that age plays a significant role

in funding success. In particular, compared to individuals aged 35-60, those aged 35 or younger have a 40-90 basis point higher success rate. Additionally, those aged 60 and above are 1.1-2.3 percentage points less successful. While this is not a direct relationship with social capital, it may suggest that these individuals may be placed lower in the P2P market, per Putnam (2015).

In studying gender, Barasinska & Schaefer (2010) find that female lenders are less risk averse than male lenders, funding credit with lower interest rates at a higher frequency. In conjunction, Pope & Sydnor (2008) find that single women pay 0.4% less interest than men. These factors may indicate that edges are more likely to form between borrowers if they are women.

The structural dynamics of the network could be explained by the theory of Relational Herding, as formulated by De Liu, Brass, Lu, & Chen (2015). The authors explain that in the face of uncertainty, actors tend to exhibit a clustering behaviour, even putting aside their own private information. Though, other social dynamics may also explain the phenomenon. Devenow & Welch (1996) posit that herding may also occur if individuals blindly follow others without rational analysis. In the P2P context, Herzenstein, Dholakia, & Andrews (2011) show evidence of strategic herding when borrowers observe that a loan has gotten funding in the past.

Further, Podolny (1993) explains that the ways in which information flows in social networks can explain its outcomes. In particular, if the reputation or status of an individual can be observed, they are said to be "pipes" for other actors to transact with them. However, if this reputation is merely *perceived*, then it said to be a "prism". In our context, prism seems more appropriate as users usually cannot directly observe prestige-related characteristics, however, this can be tested by constructing a variable related to latent status.

Hypotheses	Type	Dyad	ERGM Term	Motivation
H_2^a Younger borrowers tend to form more relationships with each other	Exogenous	Independent	nodecov()	The term captures how the age covariate affects the likelihood of edge formation. Since we want a numeric range, this term is most appropriate.
H_3^a Women tend to form more relationships with other borrowers, regardless of gender	Exogenous	Independent	nodefactor()	The term captures the tendency of women to form more edges, regardless of the other gender.
H_4^a Borrowers tend to exhibit herding around each other	Endogenous	Interdependent	gwesp()	The term captures the tendency for triadic relationships to form. In other words, for ties to form due to the presence of other ties in the dyad's neighbourhood.
H_5^a Borrowers tend to form ties around other high-status borrowers	Endogenous	Interdependent	gwd() & nodecov()	The term describes degree distribution, indicating whether a borrower is popular to the degree specified. It does not necessarily relate to community. It is necessary to also use an indicator of prestige to understand whether pipe or prism dynamics may be more prevalent.

Our research makes notable contributions to existing research on social network analyses on P2P networks. The study primarily expands upon the existing ideas of social capital by analysing it with respect to European P2P lending market and evaluating the interactions between structural and non-structural factors such as demographics. Typically, research tends to consider them independently and utilises predictive, rather than purely statistical models.

Following this, the report will outline the methodology employed. In this section, we will discuss the dataset utilised, specific data processing steps and other considerations relating to the empirical or network environment. Further, we justify the empirical structure of the analysis, evaluating the specific methodologies utilised with respect to the wider quantitative landscape. Subsequently, the results are outlined for each model and hypothesis, alongside their interpretations for our research problem. Finally, we conclude the report by summarising the core research problem, our

empirical set-up alongside considerations for future research.

2 Methodology

Within this section, we construct the empirical layout of our research problem, including the dataset and data processing steps.

2.1 Dataset

The study utilises a publicly-retrieved dataset from a leading European P2P platform called Bondora. The dataset contains detailed information on both defaulted and non-defaulted loans given to users between February 2009 and July 2021. Prior to any manipulation, the dataset contains a range of numeric, binary, categorical, and time-series attributes across 85,087 unique users and 179,235 individual loans. The data was originally collected by Siddhartha (n.d.) in 2021 and published on kaggle.com. While the user published the data, they simply downloaded it from a now-defunct page on Bondora's website. Public data is no longer offered by Bondora, access to newer data is not possible.

2.2 Data Processing and Network Construction

To make the data usable for our research, it was processed. The full steps can be seen in the Appendix. First, only attributes relevant to the research were kept for resource efficiency and ease of use. Following this, any rows with missing values were entirely removed to preserve a complete dataset. This step did not remove a significant number of data points. Following this, it is mandatory to reduce the size of the overall dataset to ensure that any analyses conducted can converge and do so in a timely manner. We did this by randomly removing data points until the dataset had 500 remaining observations. While a sampling bias is technically possible as a result, the pseudo-random data reduction should minimize this effect.

To construct the network, we need to specify edges between users. While there are many ways to do this, we prefer an approach that does not introduce unnecessary complexity to models' interpretations and maintains a meaningful semantic relationship between borrowers. Ultimately, we create edges based on how similar borrowers are across several dimensions, computed by cosine similarity. The intuition here arises from the fact that certain behaviours or social processes *groups* borrowers together. This technique has the advantage of discounting vector size for its angle, which reduces the influence of outliers in defining how similar individuals are. However, to avoid a fully connected network that would make ERGM completely redundant, we establish a threshold of `cosine_sim=0.5` to determine whether an edge can exist. This enables us to control how similar individuals must be for our analysis while maintaining a reasonable density.

Defining the dimensions that make an individual similar to another is a difficult task, however, we opted for attributes describing users' loans rather than their demographic or financial characteristics. This is *crucial* because we must isolate these variables from any outcome being studied. Otherwise, our analysis will be severely biased by data leakage and possibly even simultaneous equation bias. Further, we chose to standardise the attributes chosen for cosine similarity to avoid bias introduced by differing variable scales. Ultimately, the edges formed are weighted [1] by how

---

[1]We understand that weighted edges make ERGMs significantly more difficult. We can definitely make them binary for

similar individuals are to each other. The attributes excluded from this similarity measure were later added to the `igraph` network object as vertex attributes.

2.3 Descriptive Statistics and Network Overview

Having processed the dataset and created a weighted network of similar borrowers, we provide an overview of the descriptive statistics relating to the network and wider dataset. The figure below shows an overview of the network, however, since we rely on sampling for the nodes, the network is not equivalent to the underlying population. Observing the figure, we see that there are indeed prevalent communities, however, there is also significant overlap between them.

Figure 1: Plot of the Bondera Network with and without Community Detection

No Community Detection   **Walktrap Communities**
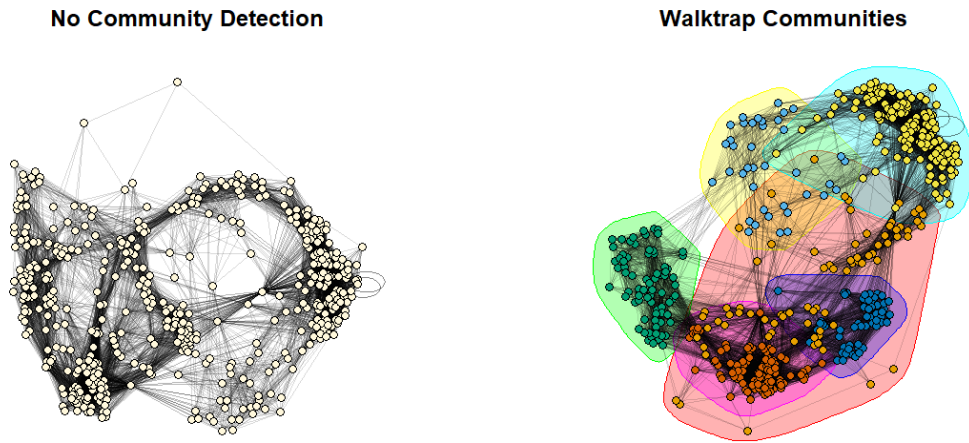
2

Observing the figure below, we see low betweenness centrality and degree distribution without any logarithmic adjustments. The latter indicates that most nodes have very low connectivity. This may have been a result of the threshold set for the similarity. The former suggests that few nodes act as the bridge between different communities of borrowers and that the few highly connected nodes are crucial for the connection of the wider P2P network.

---

further analysis. We believe that making them weighted can make the analysis more meaningful because it would allow us to determine the degree to which a predictor makes someone similar which can be important in determining something like the relational herding or prism/pipe effects. To be discussed.

[2]We recognise that this reciprocity may not beideal and we are looking for ways to deal with the 100% reciprocity. Potential point for discussion.

Table 1: Network Descriptive Statistics

Network Measure	Value
Number of Vertices	495
Number of Edges	14,017
Density	0.109
Reciprocity	1.000
Transitivity	0.692
Mean Distance	2.112
Dyad Census: Mutual	13,269
Dyad Census: Null	108,996

Figure 2: Degree and Betweeness Distribution of the P2P Network
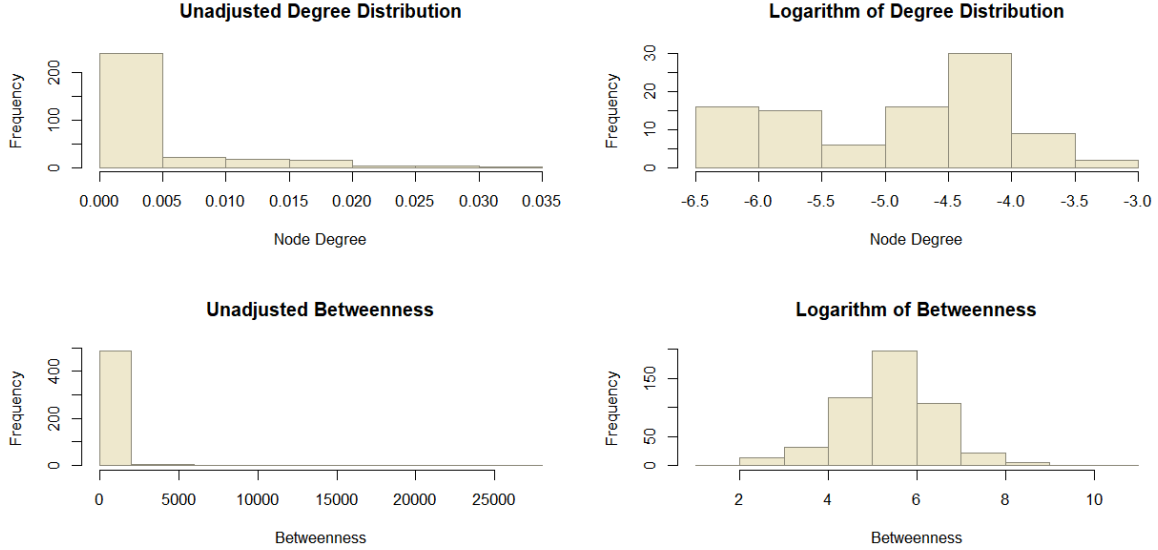
The figure below shows various aspects of the wider dataset. Firstly, we observe that there are significantly more males within our subsample borrowing than females. Notably, some users did note "Other" for their gender. It is unclear whether this relates to the user being non-binary or whether it was a data error. Generally, borrowers tend to be young, with the median hovering around 22. The interquartile range is approximately 10 years, suggesting that most users lean younger. The duration of users' loans tends to be extremely left skewed, indicating the the majority of users request very long loans. There is quite a large range in terms of the loan amounts and their interest rates. The median loan amount hovers around 2000 euros, with most of the users not borrowing more than 4000. This distribution reaffirms that users prefer loans on the lower end. For most users, the interest rate is tightly dispersed and below 40%. This could suggest that loan providers are quite selective in their clientèle. Notably, some users do exceed 100% interest rates, suggesting that the data has users with extraordinary circumstances.

Figure 3: Descriptive Statistics of the wider Dataset
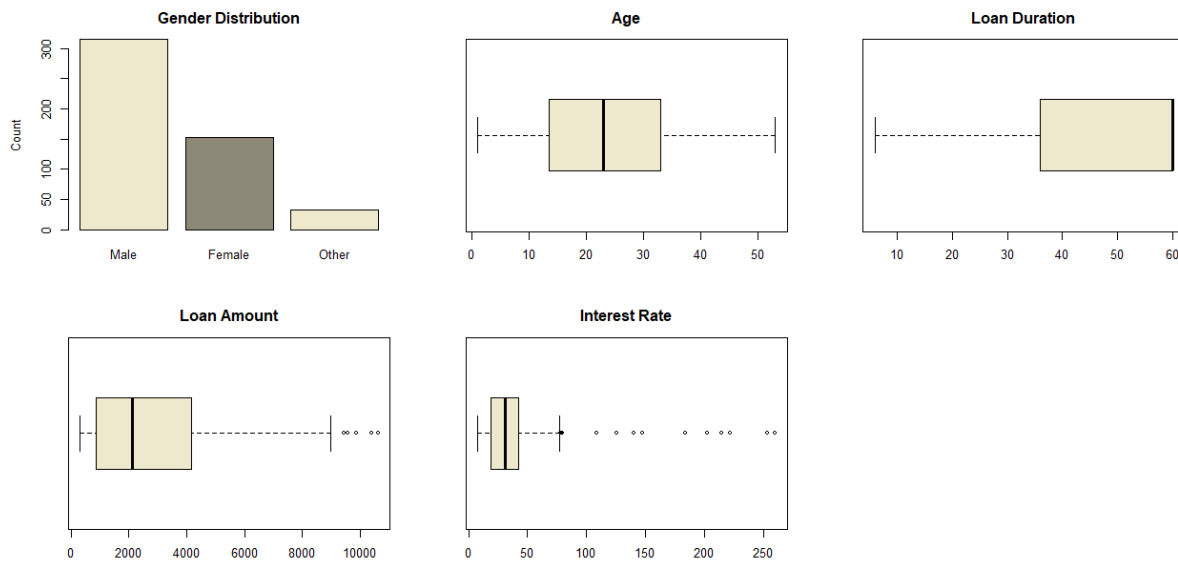
A Supplements

A.1 Data Preprocessing

A.2 Source Code - Data Preprocessing

```r
# ------------------------------------------------------------------------------------- #
#install.packages("viridis") # For Colours
#install.packages("here") # To locate files from RProj
# ------------------------------------------------------------------------------------- #
# Import the Bondora P2P Dataset
obj_paths = "resources/objects/"
bondora_raw <- read.csv("dataset/LoanData_Bondora.csv")
raw_cols <- colnames(bondora_raw)
# ------------------------------------------------------------------------------------- #
# Select Columns to Keep
keep_cols <- c("LoanId", "UserName","Age", "Gender",
               "Country", "Amount", "Interest","LoanDuration",
               "UseOfLoan", "Education", "MaritalStatus",
               "Rating", "Restructured", "MonthlyPayment")

bondora <- bondora_raw[keep_cols]

# Remove Rows with any NAs -> Complete Dataset Preferred
print(paste("NA Count |",sum(is.na(bondora)),"rows"))
bondora <- na.omit(bondora)
print(paste("NA Count |",sum(is.na(bondora)),"rows"))

# Remove users with only -1 Stated UseofLoan
bondora_clean <- bondora[bondora$UseOfLoan != -1, ]

# Remove Users with only One Loan
user_counts <- table(bondora_clean$UserName)
multi_users <- names(user_counts[user_counts > 5])
bondora_clean <- bondora_clean[bondora_clean$UserName %in% multi_users, ]

# See if Ratings are Properly Encoded
unique(bondora_clean$Rating)

# See distribution of UserName Counts
hist(table(bondora_clean$UserName))
barplot(table(bondora_clean$UseOfLoan))

# Extract UseofLoan Types and Turn into Factor
bondora_clean$UseOfLoan_factor <- as.factor(bondora_clean$UseOfLoan)
unique(bondora_clean$UseOfLoan_factor)
levels(bondora_clean$UseOfLoan_factor) <- c(
  "Loan Consolidation", "Real Estate","Home Improvement",
  "Business","Education","Travel","Vehicle","Other","Health")

bondora$UseOfLoan_factor <- as.factor(bondora$UseOfLoan)
unique(bondora$UseOfLoan_factor)
levels(bondora$UseOfLoan_factor) <- c(
  "Unknown", "Loan Consolidation", "Real Estate","Home Improvement",
  "Business","Education","Travel","Vehicle","Other","Health",
  "Machinery Purchase", "Acquisition Real Estate","Other Business"
)
# ------------------------------------------------------------------------------------- #
# Observe Descriptive Statistics

cols <- viridis::viridis(30)

# Make function to save plots
save_plot <- function(plt_nam) {
  plt <- recordPlot()
  saveRDS(plt, here::here("resources","objects","preprocessing",
                          paste0(plt_nam,".Rds")))
}

# Make function to consistently plot comparisons
plot_desc_hists <- function(df1, df2, col_name, type) {
```

```r
66
67     par(mfrow=c(1,3))
68
69     hist(df1[[col_name]], xlab=type, col=cols[15], main="", breaks=10)
70     mtext("Full Sample", side=3, adj=0, line=0.25, cex=1, font=2)
71
72     hist(df2[[col_name]], xlab=type, col=cols[15], main="", breaks=10)
73     mtext("Processed Sample", side=3, adj=0, line=0.25, cex=1, font=2)
74
75     qqplot(df1[[col_name]], df2[[col_name]], main="", cex=1,
76            xlab="Full Sample", ylab="Subsample", line=0.25)
77     abline(0, 1, lty=2)
78
79     mtext("QQ Plot", side=3, adj=0, line=0.25, cex=1, font=2)
80
81     mtext(type, outer = TRUE, line = -2, side=3, cex = 1.3, font = 2)
82
83     # Reset plot window
84     par(mfrow=c(1,1), mar=c(5,4,4,2)+0.1)
85   }
86
87   plot_desc_bar <- function(df1, df2, col_name, type) {
88
89     par(mfrow=c(1,2), mar=c(5,10,4,2))
90
91     barplot(sort(table(df1[[col_name]]), decreasing = F),
92             xlab=type, col=cols[15], horiz=TRUE, las=1)
93     mtext("Full Sample", side=3, adj=0, line=0.25, cex=1, font=2)
94
95     barplot(sort(table(df2[[col_name]]), decreasing = F),
96             xlab=type, col=cols[15], horiz=TRUE, las=1)
97     mtext("Processed Sample", side=3, adj=0, line=0.25, cex=1, font=2)
98
99     mtext(type, outer = TRUE, line = -2, side=3, cex = 1.3, font = 2)
100
101    # Reset plot window
102    par(mfrow=c(1,1), mar=c(5,4,4,2)+0.1)
103  }
104
105  plot_desc_hists(bondora, bondora_clean, "Amount", "Amount")
106  save_plot("hist_amt")
107
108  plot_desc_hists(bondora, bondora_clean, "Interest", "Interest")
109  save_plot("hist_int")
110
111  plot_desc_hists(bondora, bondora_clean, "LoanDuration", "Loan Duration")
112  save_plot("hist_loandur")
113
114  plot_desc_hists(bondora, bondora_clean, "MonthlyPayment", "Monthly Payment")
115  save_plot("hist_monpmt")
116
117  plot_desc_hists(bondora, bondora_clean, "Age", "Age")
118  save_plot("hist_age")
119
120  plot_desc_bar(bondora, bondora_clean,"UseOfLoan_factor","Loan Purpose")
121  save_plot("bar_loanuse")
122
123  plot_desc_bar(bondora, bondora_clean,"Rating","Credit Rating")
124  save_plot("bar_rating")
125  # ---------------------------------------------------------------------------- #
126  # Convert Dataset into Incidence Matrix to form Network Object (for ERGM)
127  bondora_slim <- bondora_clean
128
129  # Create the Incidence Matrix for Use of Loan
130  bondora_matrix <- table(
131    bondora_slim$UserName, bondora_slim$UseOfLoan)
```

```
132  bondora_matrix[bondora_matrix > 0] <- 1 # Given that ergm.counts fails with GOF
133
134  # Create network object with counts as Edge attribute
135  bondora_net <- network::network(
136    bondora_matrix, directed=FALSE, bipartite=nrow(bondora_matrix),
137    ignore.eval = FALSE, names.eval="frequency", loops=FALSE)
138
139  # Set the bipartite Attribute - UNNCESSARY GIVEN bipartite=length(n)
140  len <- dim(bondora_matrix)[1]
141  len_b2 <- dim(bondora_matrix)[2]
142  b_indicator <- c(rep(1,len),rep(2,len_b2))
143  #network::set.vertex.attribute(
144  #  bondora_net, "bipartite", value = rep(len,len), v=1:len
145  #)
146
147  # Extract Partition 2 Labels
148  loan_use <- levels(bondora_clean$UseOfLoan_factor)
149
150  # Create Loan Type Attribute for Partition 2
151  b2_loantype <- rep(NA, len)
152  b2_loantype <- c(b2_loantype, loan_use)
153
154  if (length(b2_loantype) == network::network.size(bondora_net)) {
155    network::set.vertex.attribute(
156      bondora_net, "b2_loantype", value = b2_loantype)
157    }
158
159  # Add Age Vertex Attribute to B1
160  age <- bondora_clean$Age[match(
161    rownames(bondora_matrix), bondora_clean$UserName)]
162  b1_age <- c(age, rep(NA, len_b2))
163  network::set.vertex.attribute(
164    bondora_net, "b1_age", value=b1_age
165  )
166
167  # Add Gender Vertex Attribute to B1
168  gender <- bondora_clean$Gender[match(
169    rownames(bondora_matrix), bondora_clean$UserName)]
170  unique(gender) # Check to see if encoded properly
171  gender <- ifelse(gender == 0, "male","female")
172  b1_gender <- c(gender, rep(NA, len_b2))
173  network::set.vertex.attribute(
174    bondora_net, "b1_gender", value = b1_gender
175  )
176
177  # Save the network object
178  saveRDS(bondora_net, file=paste0(obj_paths,"preprocessing/","bondora_net.Rds"))
179  # ------------------------------------------------------------------------------ #
180  # Get the Adjacency Matrix for Loan Use Similarity (Dependent QAP Variable)
181  adj_mat_loan_use <- bondora_matrix %*% t(bondora_matrix)
182  # Remove self weights
183  diag(adj_mat_loan_use) <- 0
184
185  # Get the Adjacency Matrix for Credit Rating Similarity (Predictor in QAP)
186  incidence_rating <- table(bondora_slim$UserName, bondora_slim$Rating)
187  adj_mat_rating <- incidence_rating %*% t(incidence_rating)
188  diag(adj_mat_rating) <- 0
189
190  # Get the Adjacency Matrix for Loan Amount (Control in QAP) - BINARY
191  # First, bin the Loan Amounts
192  bondora_slim$Amount_bins <- cut(
193    bondora_slim$Amount, breaks=c(0,2000,4000,6000,8000,10000),
194    labels = c(1:5)
195  )
196  incidence_amount_bins <- table(bondora_slim$UserName, bondora_slim$Amount_bins)
197  adj_mat_amount_bins <- incidence_amount_bins %*% t(incidence_amount_bins)
```

```
198  diag(adj_mat_amount_bins) <- 0
199
200  # Continous Absolute Difference Approach
201  avg_amount <- tapply(bondora_slim$Amount, bondora_slim$UserName, mean)
202  adj_mat_amount_diff <- outer(avg_amount, avg_amount,
203                               FUN = function(x,y) abs(x - y))
204  diag(adj_mat_amount_diff) <- 0
205
206  # Get Matrix for Differences in Age
207  incidence_age <- table(bondora_slim$UserName, bondora_slim$Age)
208  borrower_ages <- as.numeric(colnames(incidence_age)[max.col(incidence_age)])
209  names(borrower_ages) <- rownames(incidence_age)
210  adj_mat_age <- outer(borrower_ages, borrower_ages,
211                       FUN = function(x, y) abs(x - y))
212  rownames(adj_mat_age) <- colnames(adj_mat_age) <- names(borrower_ages)
213
214  # Get Adjacency Matrix for (same) Gender
215  incidence_gender <- table(bondora_slim$UserName, bondora_slim$Gender)
216  adj_mat_gender <- incidence_gender %*% t(incidence_gender)
217  # Make the matrix binary for homophily
218  adj_mat_gender <- ifelse(adj_mat_gender > 0, 1, 0)
219  diag(adj_mat_gender) <- 0
220
221  # Get Adjacency Matrix for Differences in Average Loan Duration
222  borrower_loandur <- sapply(tapply(bondora_slim$LoanDuration,
223                                    bondora_slim$UserName, unique),
224                             mean)
225  adj_mat_loandur_diff <- outer(borrower_loandur, borrower_loandur,
226                                FUN=function(x,y) abs(x-y))
227  diag(adj_mat_loandur_diff) <- 0
228
229  # Get Adjacency Matrix for Homophily in Restructure of Loans
230  incidence_restructure <- table(bondora_slim$UserName, bondora_slim$Restructured)
231  adj_mat_rest <- incidence_restructure %*% t(incidence_restructure)
232  diag(adj_mat_rest) <- 0
233
234  # Save the objects for the QAP Regression in different Script
235  qap_paths = paste0(obj_paths,"/qap/")
236  saveRDS(b_indicator, file=paste0(
237    "resources/objects/preprocessing/indicator.Rds"))
238
239  saveRDS(adj_mat_loan_use, file=paste0(qap_paths,"adj_mat_loanuse.Rds"))
240  saveRDS(adj_mat_rating, file=paste0(qap_paths,"adj_mat_rating.Rds"))
241  saveRDS(adj_mat_amount_diff, file=paste0(qap_paths,"adj_mat_amtdiffs.Rds"))
242  saveRDS(adj_mat_age, file=paste0(qap_paths,"adj_mat_agediffs.Rds"))
243  saveRDS(adj_mat_gender, file=paste0(qap_paths,"adj_mat_gender.Rds"))
244  saveRDS(adj_mat_loandur_diff, file=paste0(qap_paths,"adj_mat_loandurdiffs.Rds"))
245  saveRDS(adj_mat_rest, file=paste0(qap_paths,"adj_mat_rest.Rds"))
246  # ----------------------------------------------------------------------- #
```

scripts/bondora_preprocessing.R

A.3 Source Code - QAP Linear Regression

```r
1  # ---------------------------------------------------------------------- #
2  # If not already Installed
3  install.packages("viridis") # For Colours
4  #install.packages("here")    # To locate files from RProj
5
6  # Set colour palette
7  cols <- viridis::viridis(30)
8  # ---------------------------------------------------------------------- #
9  # Load Relevant Files
10 qap_path="resources/objects/qap/"
11
12 loan_use_mat <- readRDS(paste0(qap_path,"adj_mat_loanuse.RDS"))
13 rating_mat <- readRDS(paste0(qap_path,"adj_mat_rating.RDS"))
14 amt_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_amtdiffs.RDS"))
15 age_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_agediffs.RDS"))
16 gender_mat <- readRDS(paste0(qap_path,"adj_mat_gender.RDS"))
17 loandur_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_loandurdiffs.RDS"))
18 rest_mat <- readRDS(paste0(qap_path,"adj_mat_rest.RDS"))
19 # ---------------------------------------------------------------------- #
20 # Create function to determine significance from t-value statistic
21 t_to_stars <- function(t) {
22   stars <- rep("", length(t))
23   stars[abs(t) >= 1.96]  <- "*"
24   stars[abs(t) >= 2.576] <- "**"
25   stars[abs(t) >= 3.291] <- "***"
26
27   return(stars)
28 }
29
30 # Make function to save plots
31 save_qap_plot <- function(plt_nam) {
32   plt <- recordPlot()
33   saveRDS(plt, here::here("resources","objects","qap",
34                           paste0(plt_nam,".Rds")))
35 }
36
37 var_names <- c("Intercept", "Rating","Loan Amount","Age","Gender",
38                "Loan Duration","Restructured")
39 pred_vars <- list(rating_mat, amt_diffs_mat, age_diffs_mat,
40                   gender_mat, loandur_diffs_mat, rest_mat)
41 # ---------------------------------------------------------------------- #
42 # Basic QAP Linear Regression
43 qap_m1 <- sna::netlm(y = loan_use_mat,
44                      x = list(rating_mat, amt_diffs_mat, age_diffs_mat,
45                               gender_mat, loandur_diffs_mat, rest_mat),
46                      nullhyp = "qapspp", reps = 2500)
47 qap_m1$names <- var_names
48 summary(qap_m1)
49
50 results_m1 <- qap_m1$coefficients
51 names(results_m1) <- var_names
52 results_sig_m1 <- paste(round(results_m1,3), t_to_stars(qap_m1$tstat))
53
54 # Plot Residuals
55 hist(qap_m1$residuals, main="QAP Residuals", col = cols[15])
56
57 # Save Model
58 saveRDS(qap_m1, file = paste0(qap_path,"qap_m1.RDS"))
59 # ---------------------------------------------------------------------- #
60 # Standardised QAP Linear Regression
61 scaled_dep <- scale(loan_use_mat)
62 scaled_pred <- lapply(pred_vars, scale)
63
64 qap_m2 <- sna::netlm(y = scaled_dep,
65                      x = scaled_pred,
```

14

```r
66                        nullhyp = "qapspp", reps = 2500)
67 qap_m2$names <- var_names
68 summary(qap_m2)
69
70 # Plot the result
71 results_m2 <- qap_m2$coefficients
72 names(results_m2) <- var_names
73 results_sig_m2 <- paste(round(results_m2,3), t_to_stars(qap_m2$tstat))
74
75 # Plot Residuals
76 hist(qap_m2$residuals, main="QAP Residuals", col = cols[15])
77
78 # Save Model
79 saveRDS(qap_m2, file = paste0(qap_path,"qap_m2.RDS"))
80 # ------------------------------------------------------------------------ #
81 # Plot the result
82 par(mfrow=c(1,2))
83
84 qap_plot_m1 <- barplot(results_m1, col = cols[15],  border = cols[10],
85                        ylim = c(min(results_m1) + min(results_m1)*0.15,
86                                 max(results_m1) + max(results_m1)*0.15),
87                        main="QAP Model Results (Unstandardised)")
88 text(x = qap_plot_m1,
89      y = results_m1 + sign(results_m1)*(0.075*diff(range(results_m1))),
90      labels = results_sig_m1, font = 2)
91
92
93 qap_plot_m2 <- barplot(results_m2, col = cols[15],  border = cols[10],
94                        ylim = c(min(results_m2) + min(results_m2)*0.15,
95                                 max(results_m2) + max(results_m2)*0.15),
96                        main="QAP Model Results (Standardised)")
97 text(x = qap_plot_m2,
98      y = results_m2 + sign(results_m2)*(0.075*diff(range(results_m2))),
99      labels = results_sig_m2, font = 2)
100
101 save_qap_plot("unstd_std_plot")
102
103 par(mfrow=c(1,1))
```

scripts/qap_network_analysis.R

A.4 Source Code - ERGM Network Analysis

```r
# ------------------------------------------------------------------------------------ #
# If not already Installed
install.packages("viridis")      # For Colours
#install.packages("ergm.count") # No longer needed
install.packages("Rglpk")        # Additional solver for ERGMs
#install.packages("here")        # To locate files from RProj

# Import the Network and Other Object
ergm_path <- "resources/objects/ergm/"
bondora_net <- readRDS(here::here(
  "resources","objects","preprocessing","bondora_net.Rds"))
b_indicator <- readRDS(here::here(
  "resources","objects","preprocessing","indicator.Rds"))

# Set colour palette
cols <- viridis::viridis(30)

# Determine acceptable core count
n_cores <- parallel::detectCores() - 3 # Leave some out for other processes
print(paste("You have",n_cores,"usable cores"))

# Replicability
seed(42)

# Save plots
save_ergm_plot <- function(plt_nam) {
  plt <- recordPlot()
  saveRDS(plt, here::here("resources","objects","ergm",
                          paste0(plt_nam,".Rds")))
}
# ------------------------------------------------------------------------------------ #
# Copy network for plotting
bondora_plot <- bondora_net

# Get node type for plotting
type_indicator <- ifelse(b_indicator == 2, TRUE,FALSE)
shape <- ifelse(type_indicator,"square","circle")
network::set.vertex.attribute(bondora_plot, "shape", shape)

# Get Category Count for Vertex Size
counts <- summary(bondora_plot ~ b2sociality)
counts_att <- ifelse(type_indicator, counts^0.75, counts^0.6)
network::set.vertex.attribute(bondora_plot, "size", counts_att)

# Colours for the Node Types
plot_cols <- ifelse(type_indicator, cols[5], cols[30])
network::set.vertex.attribute(bondora_plot, "color", plot_cols)

# Legend Plotting
type_legend <- ifelse(type_indicator, "Borrowers", "Loan Type")
type_legend <- as.factor(type_legend)

# Plot the Network
plot(snafun::to_igraph(bondora_plot),
     main = "Bipartite User-LoanUse",
     edge.arrow.size = 0.3,
     edge.color = rgb(0,0,0, alpha = 0.35),
     vertex.frame.color = "black",
     vertex.label = NA,
     vertex.frame.size = 3,
     edge.curved = FALSE,
     layout=igraph::layout.fruchterman.reingold)
legend("bottomleft", legend = levels(type_legend),
       inset = c(0.1, 0.02),
       col = c(cols[30], cols[5]),
```

```r
66            pch = c(16,15),
67            title = "Node Partitions", title.font = 2,
68            cex = 0.8,
69            lwd = 1,
70            bg = rgb(0,0,0, alpha=0.025))
71  save_ergm_plot("network_plot")
72
73  # Summary Statistics
74  snafun::g_density(bondora_net)
75  snafun::g_centralize(bondora_net)
76
77  # Degree and Betweenness Distribution
78  deg_dist <- snafun::g_degree_distribution(bondora_net)
79  bet_dist <- snafun::v_betweenness(bondora_net)
80
81  # Other Descriptives
82  summary(bondora_net ~ b1degree(1:9))
83  summary(bondora_net ~ b2degree(1:10))
84
85  par(mfrow = c(1,2))
86
87  hist(deg_dist,
88       main = "Unadjusted Degree Distribution",
89       xlab = "Node Degree",
90       col = cols[15],
91       border = cols[10])
92
93  hist(bet_dist,
94       main = "Unadjusted Betweenness",
95       xlab = "Betweenness",
96       col = cols[15],
97       border = cols[10])
98
99  par(mfrow = c(1,1))
100 # ------------------------------------------------------------------------ #
101 # Make function to calculate probabilities from log odds
102 lodds_to_prob <- function(l_odd) {
103   return(exp(l_odd) / (1 + exp(l_odd)))
104 }
105 # Make function to save ERGM object
106 save_ergm <- function(object, id) {
107   saveRDS(object, file=here::here(
108     "resources","objects","ergm",id,".Rds"))
109 }
110 # Make function to conduct ERGMs automatically
111 auto_ergm <- function(model, mcmc, name) {
112
113   # Conducts the GOF Diagnostics and then saves the model,
114   # mcmc diagnostics and gof object in a list.
115   # This list can be imported as an .RDS object into the R environment
116
117   # Diagnostics
118   if (mcmc) {
119     ergm::mcmc.diagnostics(model)
120   }
121
122   # The GOF must be adjusted otherwise it takes too long
123   # We do not limit the GOF by changing its range of parameters
124   gof <- ergm::gof(model,
125                    control = ergm::control.gof.ergm(
126                      nsim = 200,
127                      MCMC.burnin = 5000,
128                      MCMC.interval = 1000,
129                      parallel = n_cores,
130                      parallel.type = "PSOCK"
131                    ))
```

```
132
133    # Return List to view each item separately
134    result <- list(model, gof)
135    names(result) <- c("model","gof")
136    save_ergm(result, paste0(name,"_panel"))
137
138    return(result)
139  }
140  # ---------------------------------------------------------------------------- #
141  # Find max degree
142  (max_deg <- max(summary(bondora_net ~ b2factor("b2_loantype"))))
143  # ---------------------------------------------------------------------------- #
144  # Base Model + GOF
145  formula_base_model <- bondora_net ~ edges
146  base_ergm <- ergm::ergm(formula_base_model)
147  base_ergm_panel <- auto_ergm(base_ergm, mcmc = FALSE, name = "ergm_base")
148  snafun::stat_plot_gof(base_ergm_panel$gof)
149  models = list(base_ergm)
150  texreg::screenreg(models)
151  # ---------------------------------------------------------------------------- #
152  # Base Model + Edge Counts + GOF
153  #base_model_counts <- ergm::ergm(bondora_net ~ edges, response="frequency",
154  #                                reference = ~ Poisson)
155  #basemodel_counts_gof <- ergm::gof(base_model_counts)
156  #snafun::stat_plot_gof(basemodel_counts_gof)
157  #
158  #texreg::screenreg(list(base_model, base_model_counts))
159  # ---------------------------------------------------------------------------- #
160  # Iteration 1 + MCMC Diagnostics + GOF
161  model_1_params <- bondora_net ~ edges +
162    # b1 decay can be very low since 9 b2
163    gwb1degree(decay=0.15, fixed=TRUE)
164
165  model_1 <- ergm::ergm(
166    model_1_params,
167
168    # Max b2 degree is 72, so this constraint is reasonable
169    # and helps convergence significantly.
170    # Technically in the Bondora population this can be
171    # far higher but we are studying a subsample.
172    #constraints = ~ bd(minout = 0, maxout = 80),
173
174    control = ergm::control.ergm(
175      # Greater burn-in for cleaner result
176      MCMC.burnin = 20000,
177      # Greater sample size for greater stability
178      MCMC.samplesize = 100000,
179      seed = 42,
180      MCMC.interval = 1000,
181      # Only needed for convergence pvals to improve
182      MCMLE.maxit = 45,
183      # Smaller steps for stability
184      MCMLE.steplength = 0.25,
185      parallel = n_cores,
186      parallel.type = "PSOCK"
187    )
188  )
189
190  model_1_panel <- auto_ergm(model=model_1, mcmc=TRUE, name="ergm_m1")
191  model_1_panel$gof
192  snafun::stat_plot_gof(model_1_panel$gof)
193  texreg::screenreg(list(base_ergm, model_1))
194  # ---------------------------------------------------------------------------- #
195  # Iteration 2 + MCMC Diagnostics + GOF
196  model_2_params <- bondora_net ~ edges +
197    # low decay important because there is high clustering around low degrees
```

```r
198    gwb1degree(decay=0.15, fixed=TRUE) +
199    # decay should be higher due to wider variation in degree but
200    # too high of degree makes the traces concentrated around the tails.
201    gwb1dsp(decay=0.5, fixed=TRUE)
202
203  model_2 <- ergm::ergm(
204    model_2_params,
205
206    # Max b2 degree is 72, so this constraint is reasonable
207    # and helps convergence significantly.
208    # Technically in the Bondora population this can be
209    # far higher but we are studying a subsample.
210    constraints = ~ bd(minout = 0, maxout = 80),
211
212    control = ergm::control.ergm(
213      # Greater burn-in for cleaner result
214      MCMC.burnin = 20000,
215      # Greater sample size for greater stability
216      MCMC.samplesize = 100000,
217      seed = 42,
218      MCMC.interval = 1000,
219      # Only needed for convergence pvals to improve
220      MCMLE.maxit = 45,
221      # Smaller steps for stability
222      MCMLE.steplength = 0.25,
223      parallel = n_cores,
224      parallel.type = "PSOCK"
225      )
226    )
227
228  model_2_panel <- auto_ergm(model=model_2, mcmc=TRUE, name="ergm_m2")
229  snafun::stat_plot_gof(model_2_panel$gof)
230  model_2_panel$gof
231  models <- list(base_ergm, model_1, model_2)
232  texreg::screenreg(models)
233  # ----------------------------------------------------------------------------- #
234  # Iteration 3 + MCMC Diagnostics + GOF
235  model_3_params <- bondora_net ~ edges +
236    # low decay important because there is high clustering around low degrees
237    gwb1degree(decay=0.15, fixed=TRUE) +
238    # decay should be higher due to wider variation in degree but
239    # too high of degree makes the traces concentrated around the tails.
240    gwb1dsp(decay=0.5, fixed=TRUE) +
241    # See differences across genders (implicitly, since b1nodemix unavailable)
242    b1nodematch("b1_gender", diff=FALSE)
243
244  model_3 <- ergm::ergm(
245    model_3_params,
246
247    # Max b2 degree is 72, so this constraint is reasonable
248    # and helps convergence significantly.
249    # Technically in the Bondora population this can be
250    # far higher but we are studying a subsample.
251    constraints = ~ bd(minout = 0, maxout = 80),
252
253    control = ergm::control.ergm(
254      # Greater burn-in for cleaner result
255      MCMC.burnin = 20000,
256      # Greater sample size for greater stability
257      MCMC.samplesize = 100000,
258      seed = 42,
259      MCMC.interval = 1000,
260      # Only needed for convergence pvals to improve
261      MCMLE.maxit = 45,
262      # Smaller steps for stability
263      MCMLE.steplength = 0.25,
```

```
264        parallel = n_cores,
265        parallel.type = "PSOCK"
266      )
267  )
268
269  model_3_panel <- auto_ergm(model=model_3, mcmc=TRUE, name="ergm_m3")
270  snafun::stat_plot_gof(model_3_panel$gof)
271  model_3_panel$gof
272  models <- list(base_ergm, model_1, model_2, model_3)
273  texreg::screenreg(models)
274  # ------------------------------------------------------------------------------ #
275  # Iteration 4 + MCMC Diagnostics + GOF
276  model_4_params <- bondora_net ~ edges +
277      # low decay important because there is high clustering around low degrees
278      gwb1degree(decay=0.15, fixed=TRUE) +
279      # decay should be higher due to wider variation in degree but
280      # too high of degree makes the traces concentrated around the tails.
281      gwb1dsp(decay=0.5, fixed=TRUE) +
282      # See differences across genders (implicitly, since b1nodemix unavailable)
283      b1nodematch("b1_gender", diff=TRUE) +
284      # See if higher ages make a difference
285      b1cov("b1_age")
286
287  model_4 <- ergm::ergm(
288      model_4_params,
289
290      # Max b2 degree is 72, so this constraint is reasonable
291      # and helps convergence significantly.
292      # Technically in the Bondora population this can be
293      # far higher but we are studying a subsample.
294      constraints = ~ bd(minout = 0, maxout = 80),
295
296      control = ergm::control.ergm(
297        # Greater burn-in for cleaner result
298        MCMC.burnin = 20000,
299        # Greater sample size for greater stability
300        MCMC.samplesize = 100000,
301        seed = 42,
302        MCMC.interval = 1000,
303        # Only needed for convergence pvals to improve
304        MCMLE.maxit = 45,
305        # Smaller steps for stability
306        MCMLE.steplength = 0.25,
307        parallel = n_cores,
308        parallel.type = "PSOCK"
309      )
310  )
311
312  model_4_panel <- auto_ergm(model=model_4, mcmc=TRUE, name="ergm_m4")
313  model_4_panel$gof
314  snafun::stat_plot_gof(model_4_panel$gof)
315  models <- list(base_ergm, model_1, model_2, model_3, model_4)
316  texreg::screenreg(models)
317  # ------------------------------------------------------------------------------ #
```

scripts/ergm_network_analysis.R

References

Adler, P. S., & Kwon, S.-W. (2002). Social capital: Prospects for a new concept. *Academy of Management Review*, *27*(1), 17–40.

Akerlof, G. A. (1978). The market for "lemons": Quality uncertainty and the market mechanism. In *Uncertainty in economics* (pp. 235–251). Elsevier.

Bachmann, A., Becker, A., Buerckner, D., Hilker, M., Kock, F., Lehmann, M., . . . Funk, B. (2011). Online peer-to-peer lending – a literature review. *Journal of Internet Banking and Commerce*, *16*.

Barasinska, N., & Schaefer, D. (2010). Does gender affect funding success at the peer-to-peer credit markets? Evidence from the largest german lending platform. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1738837

Cai, S., Lin, X., Xu, D., & Fu, X. (2016). Judging online peer-to-peer lending behavior: A comparison of first-time and repeated borrowing requests. *Information & Management*, *53*(7), 857–867. https://doi.org/10.1016/j.im.2016.07.006

Chen, X., Zhou, L., & Wan, D. (2016). Group social capital and lending outcomes in the financial credit market: An empirical study of online peer-to-peer lending. *Electronic Commerce Research and Applications*, *15*, 1–13. https://doi.org/10.1016/j.elerap.2015.11.003

De Liu, Brass, D. J., Lu, Y., & Chen, D. (2015). Friendships in online peer-to-peer lending. *MIS Quarterly*, *39*(3), 729–742. Retrieved from https://www.jstor.org/stable/26629629

Devenow, A., & Welch, I. (1996). Rational herding in financial economics. *Papers and Proceedings of the Tenth Annual Congress of the European Economic Association*, *40*(3), 603–615. https://doi.org/10.1016/0014-2921(95)00073-9

Herzenstein, M., Dholakia, U. M., & Andrews, R. L. (2011). Strategic herding behavior in peer-to-peer loan auctions. *Journal of Interactive Marketing*, *25*(1), 27–36. https://doi.org/10.1016/j.intmar.2010.07.001

Jackson, M. O. (2011). An overview of social networks and economic applications. In *Handbook of social economics* (Vol. 1, pp. 511–585). Elsevier. https://doi.org/10.1016/B978-0-444-53187-2.00012-7

Mingfeng Lin, N.R. Prabhala, & Siva Viswanathan. (2009). *Social networks as signaling mechanisms: Evidence from online peer-to-peer lending*. Retrieved from https://pages.stern.nyu.edu/~bakos/wise/papers/wise2009-p09_paper.pdf

Nahapiet, J., & Ghoshal, S. (1998). Social capital, intellectual capital, and the organizational advantage. *Academy of Management Review*, *23*(2), 242–266.

Podolny, J. M. (1993). A status-based model of market competition. *American Journal of Sociology*, *98*(4), 829–872. https://doi.org/10.1086/230091

Pope, D., & Sydnor, J. (2008). What's in a picture? Evidence of discrimination from prosper.com. *Monetary Economics eJournal*, *46*. https://doi.org/10.1353/jhr.2011.0025

Putnam, R. D. (2015). Bowling alone: America's declining social capital. In *The city reader* (pp. 188–196). Routledge.

Siddhartha, M. (n.d.). Bondora peer to peer lending loan data. Retrieved October 9, 2025, from https://www.kaggle.com/datasets/sid321axn/bondora-peer-to-peer-lending-loan-data

B Technology Statement

During the preparation of this work, we used ChatGPT in order to generate select parts of the R script utilised to process the dataset. Specifically, the tool was used to transform the processed dataset into a format that `igraph` would accept as a network object. No AI tool was utilised to write parts of the report. The following parts of the assignment were affected/generated by AI tool usage: **DATASET;** the data described within this section was processed partly by some code drafted by ChatGPT and edited by the group. After using this tool/service, **Samir Sabitli** evaluated the validity of the tool's outputs, including the sources that generative AI tools have used, and edited the content as needed. As a consequence, **Samir Sabitli** takes full responsibility for the content of their work.