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

List of Figures

List of Tables

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; with no stringent third parties involved in transactions, individuals have greater power in sourcing credit. This presents new realities for individuals disenfranchised by the traditional financial system (Alistair Milne & Paul Parboteeah, 2017). Traditional financial literature shows that individually, investors exhibit behavioural biases (Agrawal, 2012; Ayal, Hochman, & Zakay, 2011) in how they choose their investments, however, numerous studies demonstrate that these behavioural biases are present among non-professional P2P borrowers. For example, Herzenstein, Dholakia, & Andrews (2011) and Lee & Lee (2012) suggest that users cluster around popular loans, exhibiting 'herding' behaviour. Literature focuses predominantly on funding success (Yao, Chen, Wei, Chen, & Yang, 2019), however, Ayal, Bar-Haim, & Ofir (2018) suggests that there is insufficient attention on how borrowers form their loan portfolios. As the authors indicate, this area of research is important in understanding why less professional investors deviate from known models of rational investor behaviour. Our study aims to model and understand how P2P borrowers' behaviours influence what kinds of loans they acquire. In subsequent discussions, we consider investors as rational participants in credit markets, as described by theory, whereas borrowers refer to the users of P2P platforms.

Investigating behavioural biases, Ayal et al. (2018) find that investors' familiarity with specific assets can make them seem less risky, making recognisable assets more attractive. However, such assets lead to undiversified portfolios. Evidence with P2P platforms supports the familiarity bias in lenders, as Galak, Small, & Stephen (2010) demonstrate that lenders choose borrowers who are similar across demographic attributes such as gender, occupation, and ethnicity. Moreover, Amar, Ariely, Ayal, Cryder, & Rick (2011) explains that borrowers show an aversion to holding a *large number of debts,* preferring to first reduce debt count, even if they had other debts with higher interest rates. This irrational behaviour demonstrates that individual borrowers face behavioural biases on the *type* and *number* of debts they select. Based on these notions, we formulate our first study, asking:

Research Question 1: *Do borrowers vary in their loan use type and frequency based on their individual and loan characteristics?*

This question is answered using linear Quadratic Assignment Procedure (QAP) regressions, as we can represent similarities across loan uses and attributes as matrices, weighted by loan use frequency, as inputs in the model.

Using data from LendingClub, Serrano-Cinca, Gutiérrez-Nieto, & López-Palacios (2015) study funding success, finding that the reported loan use is a significant factor explaining differences in default rates. Specifically, the authors argue that there is heterogeneity in the credit riskiness of individuals seeking specific loan purposes, indicating that credit rating can explain some variation in borrowers' choice of loan use. Therefore, we formulate the following hypothesis:

Hypothesis 1: Borrowers with the same credit ratings choose similar loan uses

Supporting the familiarity bias, Ravina (2019) studies personal characteristics in P2P markets, finding that loans tend to be awarded to those sharing personal characteristics such as occupation. As Ayal et al. (2018) implies, this can be generalised to suggest that lenders' loan grants reflect a wider network of similar borrower preferences, which can reflect the homogeneity in loan use observed by Serrano-Cinca et al. (2015). Thus,

Hypothesis 2: Borrowers sharing the same occupation area tend to share loan uses

Further, behavioural finance suggests that individuals' financial decisions are not independent, but rather, shaped by bounded rationality and psychological preferences (Kahneman & Tversky, 1979). In the P2P context, Ayal et al. (2018) shows that these preferences can influence how borrowers choose loan types and how this can overlap with other borrowers' decisions. Using the Exponential Random Graph Model (ERGM) framework, we can identify how these behaviours manifest as structural regularities in a borrower-to-loan-use network. Therefore, we ask:

Research Question 2: *What behavioural mechanisms affect the structure of borrower-to-loan-use connections?*

The aforedescribed section outlined that the familiarity bias can lead to similarities across borrower portfolios in terms of their characteristics. However, in P2P settings, this can manifest itself as familiarity *clustering*, wherein borrowers who have previously been granted certain loan types are more likely to repeat their choices, or select other categories that appear familiar to different borrowers. Herzenstein et al. (2011), studying borrower decisions on Prosper.com, identifies a 'strategic herding' behaviour, where individuals gravitate towards popular or socially-validated loans. Considering that loan popularity is directly observable through its number of bids, we assert that borrowers linked to one loan purpose are likely to connect with others sharing the same loan use. In network terms, this behaviour results in cyclical substructures where there are overlapping nodes of each of the bipartite partitions. We capture a conservative form of this cycle through the ERGM term `cycle(4)`, formulating:

Hypothesis 3: Borrowers who share one loan purpose are likely to share another

Further, Ayal & Zakay (2009), expands on the theory of Debt Account Aversion with the Theory of Perceived Diversification, explaining that borrowers' distress increases with not just by the number of debts, but also the perceived *distinctiveness* of each debt type, suggesting that in the P2P context, debt mentally differentiated by purpose would also be seen as distressing. Therefore, we assert that borrowers try to minimise loan distinctiveness by requesting loans of a single type. This can be captured by the `b1degree(1)` term, which models whether borrowers are only linked to one loan purpose. Therefore, we formulate:

Hypothesis 4: Borrowers exhibit preferences for minimal debt distinctiveness, selecting one loan use

Notwithstanding, credit choices relate to non-structural factors. Cooper, Gorbachev, & Luengo-Prado (2023) show that younger borrowers face constraints in the types of loans they can access in typical credit markets. The authors argue that this constraint shapes how these borrowers form loan portfolios, often concentrating around different use cases such as education, personal consumption, or small business loans. Considering the systemic differences in how age affects credit consumption, we assert that younger borrowers have different loan use portfolios. The ERGM term `b1cov("age")` can capture this by assessing how changes in age affect how borrowers form connections with different loan types. Therefore:

Hypothesis 5: Younger borrowers have different loan use mixes than older borrowers

Alongside age, the borrower's gender is known to be a significant factor in influencing credit use decisions. Aliano, Alnabulsi, Cestari, & Ragni (2023) study the role of gender and other factors on borrowers' probability of default on the Bondora.com, finding a persistent gender effect on different loan uses; women tend to default less on health, home, and business loans. Croson & Gneezy (2009) shows that these differences can be due to risk perception and self-selection effects, where women exhibit greater risk aversion, leading to different borrowing patterns. We expect that these differences in risk tolerance and perceived creditworthiness by lenders affect loan use composition in terms of gender. The term `b1nodematch("gender")` captures this effect by assessing homophily in loan use selection. Our premise is supported by a negative estimate for the term, indicating heterophily. Therefore:

Hypothesis 6: Gender differences lead to different loan use mixes

Table 1: Hypotheses related to Study 2

Hypothesis	Term	Explanation
H_3^a : Borrowers who share one loan purpose are likely to share another	*cycle(4)*	This captures the tendency of borrowers and loan uses to be cyclical, where people cluster around shared uses of loans.
H_4^a : Borrowers exhibit preferences for minimal debt distinctiveness, selecting one loan use	*b1degree(1)*	This captures the tendency of borrowers to prefer minimal distinctiveness in their loan uses, preferring to hold only one distinct type to minimise their perceived risk.
H_5^a : Younger borrowers have different loan use mixes than older borrowers	*b1cov("age")*	This captures the tendency for younger borrowers to choose different kinds of loans based on how differently they consume credit.
H_6^a : Gender differences lead to different loan use mixes	*b1nodematch("gender")*	This captures the idea that there are systemic differences in what kinds of loan uses are preferred the genders due to their perceived riskiness and creditworthiness.

Following this section, we outline the methodology, explaining our dataset and network construction. To facilitate this, descriptive statistics are analysed for both the dataset and resulting network. Subsequently, we elaborate on our choice of models, discussing how network models can be used to conduct the study. Then, modelling results are shown and explained to evaluate our hypotheses. Simultaneously, the robustness of the study is evaluated through the models' diagnostics. Finally, we arrive at our conclusions, summarising the study, its results, and implications. Supporting items such as the source code and specific data processing steps are shown in section A.

2 Dataset

The study utilises publicly-available data from Bondora, a European P2P lending platform primarily operating in the Baltics and Spain. The dataset contains detailed information on defaulted and non-defaulted loans granted to users between February 2009 and July 2021. Specifically, contained is a range of numeric, binary, categorical, and time-series attributes across 85,087 unique users and 179,235 individual loans. Since the company's API is no longer accessible, we utilised a publicly available repository compiled by Manu Siddhartha (2021) on kaggle.com. The user made a series of API calls to collect the data. Following this, the dataset was processed according to Appendix A.1.

2.1 Pre-Processing & Network Formation

Initially, the dataset was filtered loans between 2014 and 2016 since the data is incomplete during other periods. Since we are studying interactions across two distinct set of nodes, a bipartite network was formed with unique users as the first partition and the purpose of loans as the second. This allows us to evaluate how borrowers interact with loan purposes; a one-mode projection onto users is not meaningful for P2P networks as it produces no discernible structure. For model efficiency and performance, we randomly sample 500 individuals from the larger sample. Ultimately, we have 500 and 9 nodes in the first and second partitions, respectively.

2.2 Descriptive Statistics & Preliminary Analysis

Table 2 summarises the numeric variables between the reduced and larger samples. First, borrowers take out relatively small loans with fairly high interest rates, possibly reflecting that the average borrower is deemed fairly risky. The average loan is quite long, at approximately 46 months out of a maximum 60. Figure 9 shows that the largest credit category is HR[1], which is the lowest possible rating. So, the typical borrower is relatively uncreditworthy. Moreover, the largest categories of loan use are "Other" and "Home Improvement", suggesting these are largely personal. We observe minimal differences between the random sampling and the larger sample of individuals, per Table 2 and Figure 9, however, the study may be biased by the fact that the dataset contains only granted loans, rather than all bids for loans. This means our inferences about how borrowers behave is limited to successful bids. Further, the time period is relatively distant, meaning that recent structural changes to the P2P sector cannot be accounted for.

Table 2: Descriptive Statistics of Numeric Variables

DataFrame	Variable	Mean	Median	Std. Dev.	Min	Max
Initial Sample	Amount	2652.01	2125	2151.28	115	10630
Initial Sample	Interest	35.94	31	24.77	7.62	263.63
Initial Sample	Age	38.53	37	11.4	19	70
Initial Sample	LoanDuration	45.5	60	17.53	3	60
Reduced Sample	Amount	2637.64	2112.5	2112.2	170	10630
Reduced Sample	Interest	35.85	30	27.24	10.17	253.08
Reduced Sample	Age	38.73	37	11.34	21	69
Reduced Sample	LoanDuration	46.23	60	17.35	3	60

Table 3 shows a network density of 0.12, indicating that only 12% of all possible borrower–loan-type connections exist. This density suggests that borrowers typically engage with few loan types,

[1] Bondora uses their own proprietary credit rating system where the best rating is AA and worst is HR (Bondora, 2024). The company bases these ratings on their calculated expected probability of loss on the loan. For example, someone rated AA ranges from a expected loss of 0% to 2%, whereas for HR this is 25% - >25%.

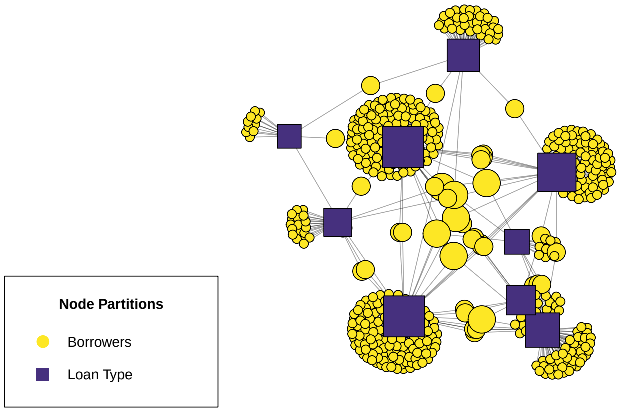
reflecting specialization in borrowing behaviour. The centralization value of 0.47 further indicates an uneven distribution across loan types, where a few popular categories attract many borrowers while most remain peripheral. The mean distance of approximately three implies that borrowers are closely connected, typically separated by only three-to-four loan-use steps.

Centrality distributions reinforce this structure Figure 2. Betweenness is highest among loan-type nodes, though several borrowers also bridge distinct loan clusters. Closeness centrality remains low but scattered, suggesting localized clustering constrained by the bipartite structure. Degree distributions show wide variation among loan types, consistent with a hub-like structure dominated by a few popular uses. However, the second partition has relatively small degree dispersion. Finally, the homogeneous eccentricity distribution indicates that no borrowers are highly isolated, reflecting a compact and moderately cohesive network overall.

Table 3: Basic Network Descriptive Statistics

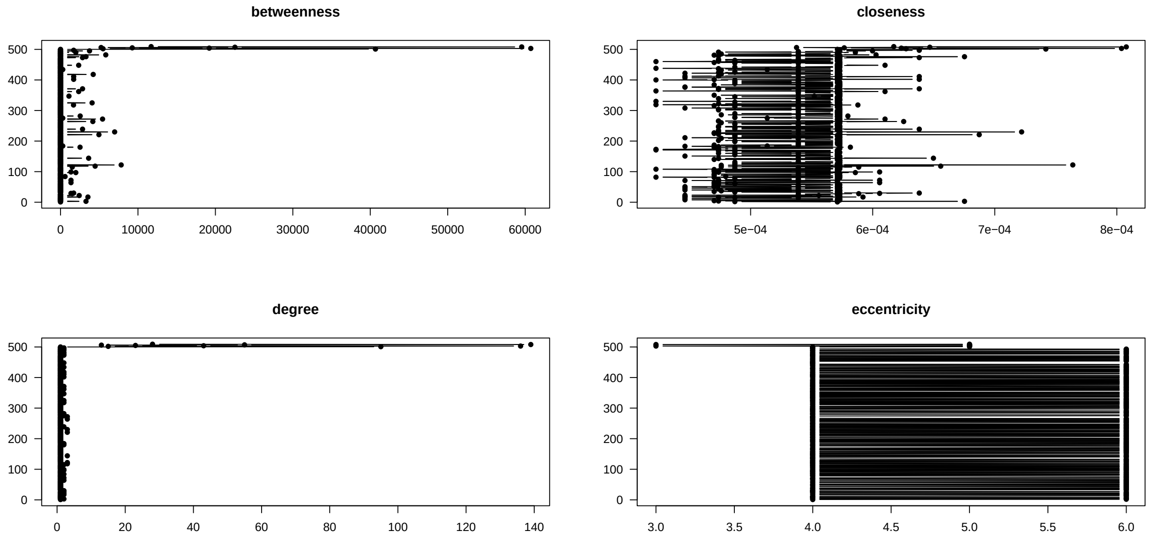
Statistic	Measure
Vertex Count	509.00
Edge Count	547.00
Density	0.12
Centralization	0.47
Mean Distance	3.66

Figure 1: Network Visualisation



Note: Each colour represents a different partition of the network. The partitions' nodes are scaled to their relative degrees. Users with many different loan types are larger than those with single loan uses. Similarly, the size of second partition nodes represents how many users belong to each.

Figure 2: Network Centrality Distribution Plots



3 Research Rationale

Our overall study analyses borrowers' behaviours, however, the studies are differentiated in their scope; the first study observes dyad-level relations among borrowers to understand how shared attributes affect borrowers' behaviours. This is inherently useful because behavioural finance aims to understand how individuals behave in *aggregate* (Ayal et al., 2018). Typically, the Quadratic Assignment Procedure is selected for dyad-level observations because basic linear regressions such as Ordinary Least Squares are fundamentally flawed and biased when using non-independent observations (Simpson, 2001).

Recognising this, studies such as Zuo (n.d.) assume that groups of borrowers have dependencies, and accordingly, use clustering methods such as K-Means or Fuzzy Clustering to understand behaviour. However, these methods are unsuitable when we require a distribution of estimates to make inferences from our results. Moreover, studying individuals as dyads rather than large clusters provides finer granularity in results. Therefore, QAP regression is the most appropriate choice for this study: it allows us to control for the non-independence among dyads, derive a statistical distribution of estimates through matrix permutation, and draw valid inferences about the relationship between borrower similarities and loan-use behaviour. This makes it both theoretically and methodologically well-suited to our research objectives and dataset.

The second study extends the analysis from a dyadic similarities to the structural formation of borrower-to-loan-use connections, which can identify the behavioural mechanisms that influence borrowers' specific loan use mixes. While QAP can account for exogenous influences, it cannot analyse higher-order network patterns such as clustering while accounting for simultaneous interactions between borrowers and loan uses. Using the Exponential Random Graph Model allows us to analyse how local behavioural tendencies aggregates to a global network structure.

Recent evidence highlights the need to focus on structural approaches in P2P lending. Liu, Baals, Osterrieder, & Hadji-Misheva (2024) studies borrowers' centrality with the Bondora P2P dataset, modelling edges through borrower-to-borrower similarity metrics. Ultimately, they find that a borrower's position is highly significant for their probability of default, using centrality as an exogenous variable in a logistic regression. In contrast, ERGM allows us to move away from predictive association and instead account for endogenous factors, allowing us to understand the likelihood that borrowers' selection of loan uses are random. Consequently, we can better understand how behavioural phenomena such as familiarity and debt account aversion can translate into *systemic* patterns of credit use.

As we aim to understand how two distinct groups of nodes interact, a bipartite approach is highly suitable. Specifically, Stivala, Wang, & Lomi (2025) shows that in bipartite networks, terms such as `cycle(4)` can be effective in capturing clustering as in hypothesis 3, whereas it would fail in one-mode projected networks. Furthermore, the flexibility of ERGM also allows for modelling exogenous impacts such as heterophily in gender-based loan use mixes, as in hypothesis 6, and covariate age effects as in hypothesis 5.

A Supplements

A.1 Data Preprocessing Steps

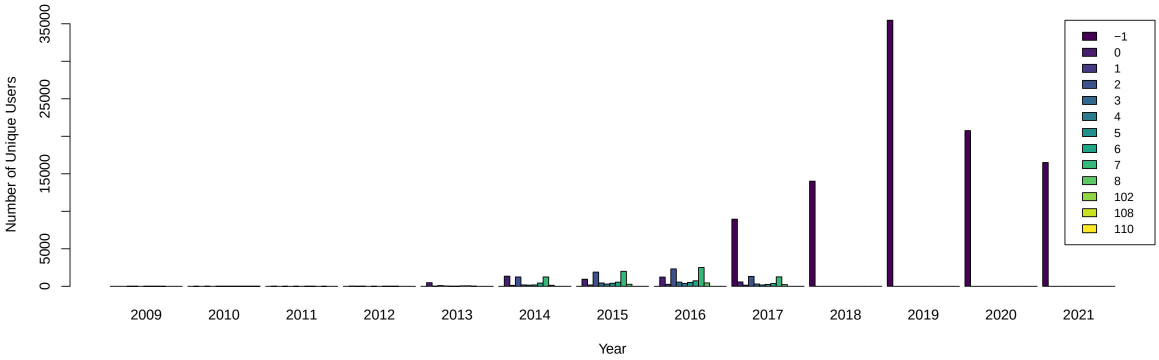
This section outlines the specific steps taken to process the data into the network object that was used in later analysis, outlining the rationale, where needed.

- Following the data import, only the following attributes were kept to make our processing steps more focused

```
keep_cols <- c("LoanId", "UserName","Age", "Gender",
               "Country", "Amount", "Interest","LoanDuration",
              "UseOfLoan", "Rating", "Restructured", "MonthlyPayment",
              "OccupationArea", "BiddingStartedOn")
```

- We removed rows with any `NA` values for the selected attributes to ensure that we have a complete dataset
- Observing Figure 3, we see that between 2017-2021, there exists many -1 values for `UseofLoan`. This is because Bondora no longer provided this detail in later datasets. However, between 2014 and 2016 this data is available. Because of this, we restrict the time period to have the least noisy data. `BiddingStartedOn` was chosen to record the time, because the auction start date can be a good indication of activity around the loan.

Figure 3: Number of Unique Users per Loan Type per Year



- Then, we sampled 500 individuals out of the filtered dataset to ensure that our models run sufficiently well and converge. By choosing 500 individuals, we preserve importance structures within the network while adding sufficient statistical power.
- When adding labels to each loan use and occupation area, we keep any possible missing values because these do not indicate lower quality data. Instead, they can carry information about lenders distribute loans according to the quality of information presented by borrowers (Yao et al., 2019). Additionally, keeping missing labels is important to ensure that the edges are not superficially limited and that nodes can be isolates if they are indeed that way in reality.
- To create the network object for ERGM models:

  - We first create an incidence matrix of size $n \times m$, borrower usernames by the loan use, where the values are the number of loans the user obtained for each loan use category. These values are then turned into binary because our modelling does not support the use of weighted edges.
  - A bipartite network is created from this incidence matrix, where the first partition are unique users, and the second partition is the loan use associated among all loans of the user. Users can share more than more loan use type.
  - We add age and gender attributes to each node of the first partition
- To create the relevant networks/matrices for QAP Regressions:
  - We first re-create the loan use incidence matrix, however, we keep the counts as the linear regression supports this.
    - We then convert the loan use incidence matrix into an adjacency matrix of shape $n \times n$ through the transformation $\mathbf{X} \cdot \mathbf{X}^{\mathbf{T}}$, which results in a matrix of shared loan uses weighted by the frequency of each user's loan count within each loan use category.
    - We then set the diagonal values of $\mathbf{X}$ to be zero to ensure that there are no loops within the network.
  - In accordance with this procedure, we create adjacency matrices for Credit Rating and Occupation Area which will be used as the main predictors in the QAP models.
    - Credit Rating is a weighted adjacency matrix based on the loan count belonging to each user for each credit rating category
    - Occupation Area is a binary adjacency matrix based on which occupation each user has reported to have held in the past when applying for each of their loans.
  - Additionally, we create a set of control variables to improve the validity and performance of the linear regression using Loan Amounts, Age, Gender, Loan Duration, and whether the loans have been restructured.
    - Loan Amounts is a weighted adjacency matrix based on five bins across the range of possible loan amounts
    - Age is an adjacency matrix based on differences in users' ages
    - Gender is a binary adjacency matrix based on users' shared gender.
    - Loan Duration is an adjacency matrix based on the differences in users' loans average durations
    - Restructured is a binary adjacency matrix based on whether the users share the fact that they have defaulted on any loans in the past. The users can either have shared both defaults and non-defaults, or both.

A.2 Distributions of Variables across Samples

Figure 4: Distribution of Loan Amount across Samples

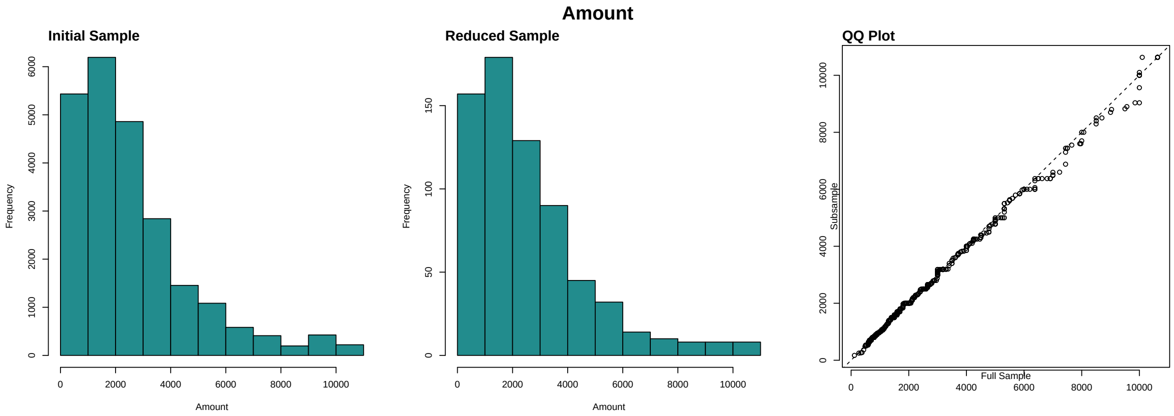


Figure 5: Distribution of Interest Rates across Samples

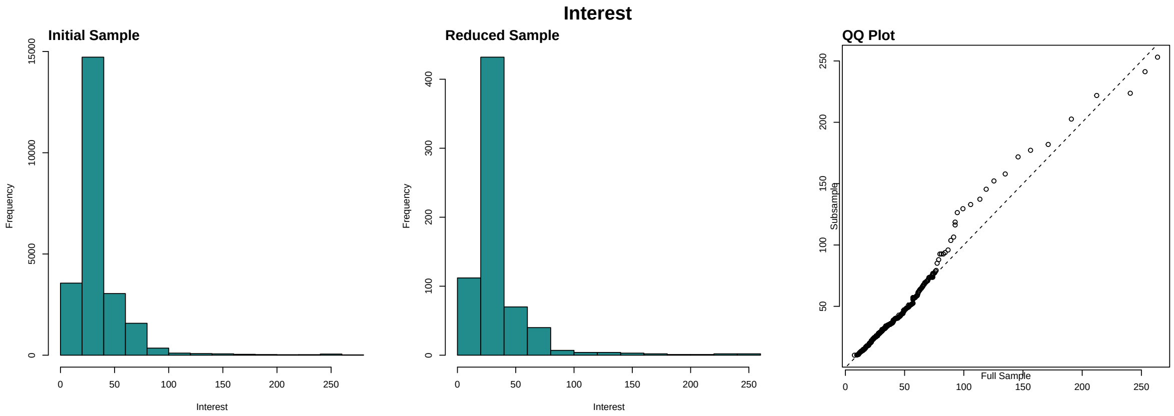


Figure 6: Distribution of Age across Samples

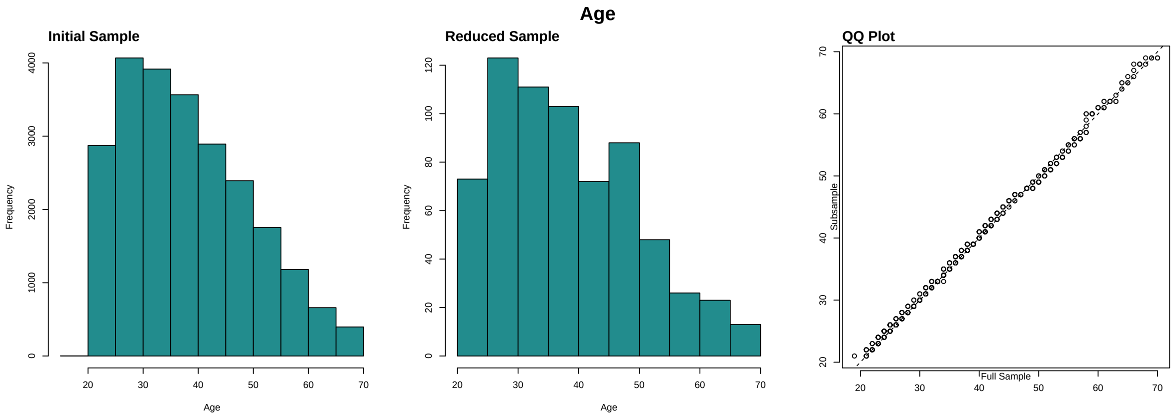


Figure 7: Distribution of Loan Duration across Samples

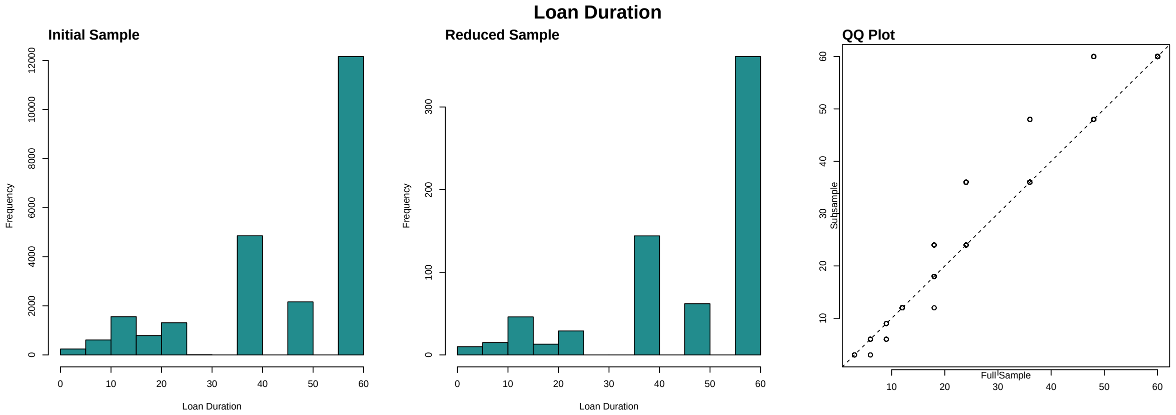


Figure 8: Distribution of Loan Purpose across Samples

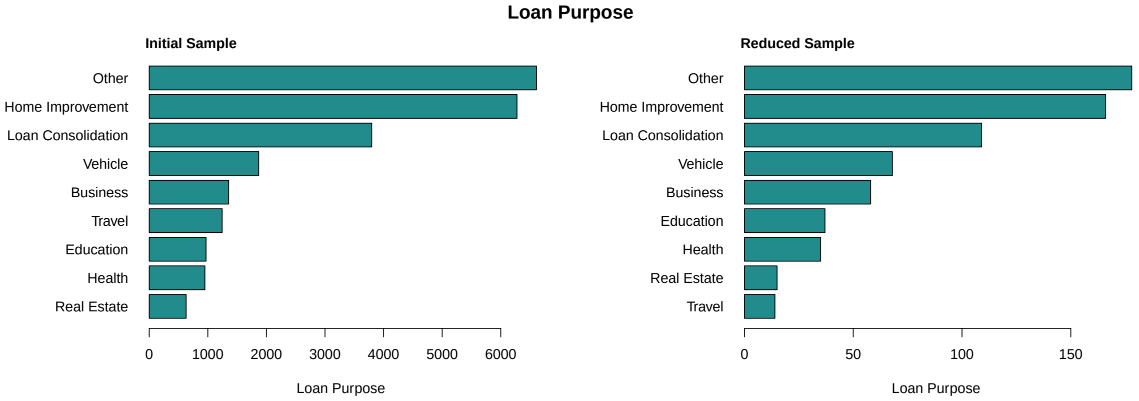


Figure 9: Distribution of Credit Ratings across Samples

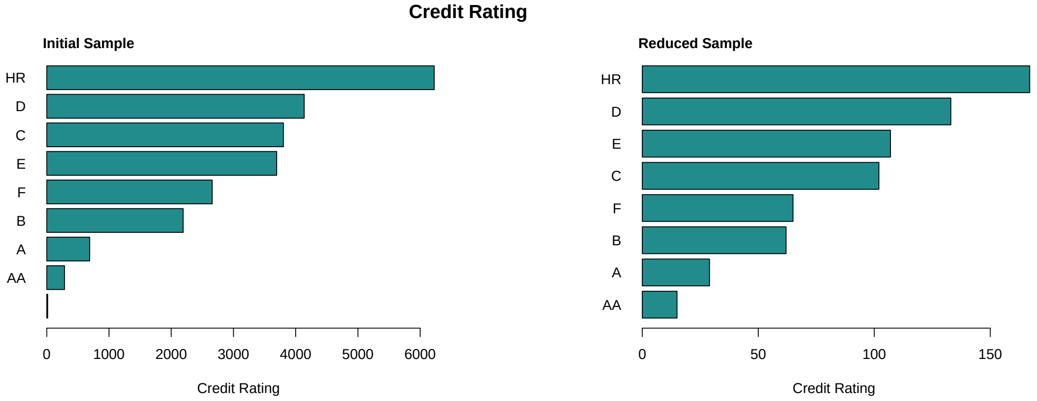
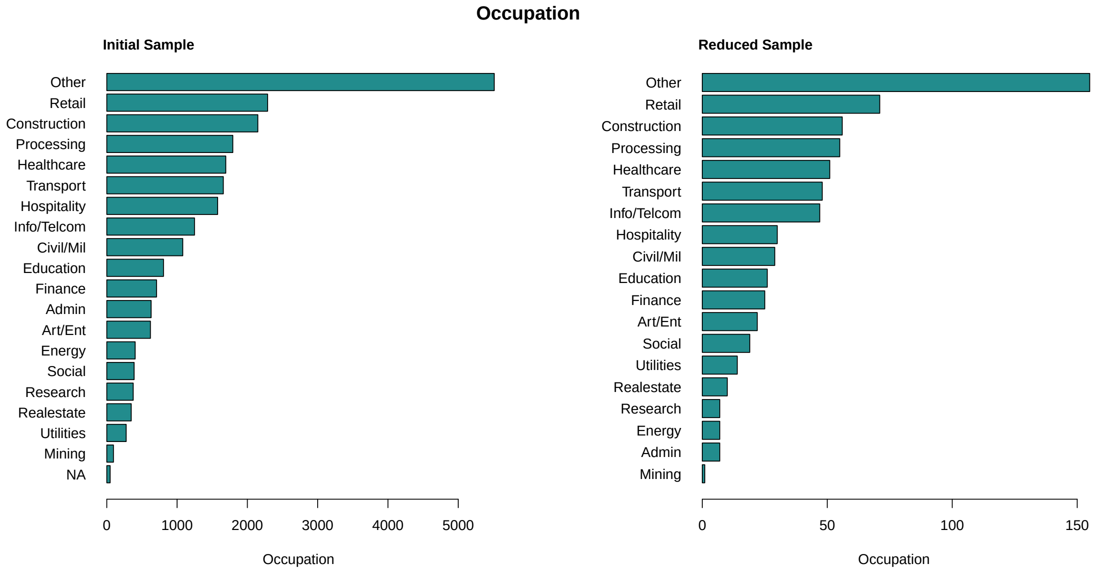


Figure 10: Distribution of Occupation Types across Samples



A.3 Source Code - Data Preprocessing

```r
# ---------------------------------------------------------------------------- #
install.packages("viridis") # For Colours
install.packages("here") # To locate files from RProj
set.seed(42)
# ---------------------------------------------------------------------------- #
# 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", "Rating", "Restructured", "MonthlyPayment",
              "OccupationArea", "BiddingStartedOn")

bondora <- bondora_raw[keep_cols]

# Change date format to the correct one
bondora$BiddingStartedOn <- as.POSIXct(bondora$BiddingStartedOn,
                                       format = "%Y-%m-%d %H:%M:%S")
bondora$year <- as.numeric(format(bondora$BiddingStartedOn, "%Y"))

# 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"))

# Observe the distribution of Loan Use over Time
unique_counts <- tapply(bondora$UserName,
                        list(bondora$year, bondora$UseOfLoan),
                        function(x) length(unique(x)))
unique_counts[is.na(unique_counts)] <- 0

barplot(t(unique_counts),           # transpose so bars = loan types
        beside = TRUE,              # side-by-side bars
        col = viridis::viridis(ncol(unique_counts)),
        legend.text = colnames(unique_counts),
        args.legend = list(x = "topright", cex = 0.8),
        xlab = "Year",
        ylab = "Number of Unique Users")
user_counts_plt <- recordPlot()
saveRDS(user_counts_plt,
        here::here("resources","objects","preprocessing","user_cnt_plt.Rds"))

# Restrict to Recent Time Period
bondora_test <- bondora[bondora$year > 2013 & bondora$year < 2017, ]

# See the distribution of LoanUses
see_counts <- function(var1, var2) {
  counts <- table(var1, var2)
  total_counts <- colSums(counts)
  prop <- total_counts / sum(total_counts)

  return(list(total_counts, round(prop,2)))
}
before_counts <- see_counts(bondora_test$UserName, bondora_test$UseOfLoan)
barplot(before_counts[[1]])

# Number of unique users in the subsample
unique_users <- unique(bondora_test$UserName)
sample_users <- sample(unique_users, 500)
bondora_clean <- bondora_test[bondora_test$UserName %in% sample_users, ]

after_counts <- see_counts(bondora_clean$UserName, bondora_clean$UseOfLoan)
```

```r
barplot(after_counts[[1]])

# DEPRECATED METHOD
# Filter People with 2+ Loans
#user_counts <- table(bondora_test$UserName)
#multi_users <- names(user_counts[user_counts > 2])
#bondora_clean <- bondora_test[bondora_test$UserName %in% multi_users, ]

# DEPRECATED METHOD
# 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, ]
#bondora_test <- bondora_raw[bondora_raw$UserName %in% multi_users, ]

# See if Ratings are Properly Encoded
unique(bondora_clean$Rating)
bondora_clean <- bondora_clean[bondora_clean$Rating != "", ]

# Extract UseofLoan Types and Turn into Factor
bondora_clean$UseOfLoan_factor <- as.factor(bondora_clean$UseOfLoan)
unique(bondora_clean$UseOfLoan_factor)

loan_use_labels <- c(
  "-1" = "NA",
  "0" ="Loan Consolidation",
  "1" = "Real Estate",
  "2" = "Home Improvement",
  "3" = "Business",
  "4" = "Education",
  "5" = "Travel",
  "6" = "Vehicle",
  "7" = "Other",
  "8" = "Health",
  "110" = "Other Business",
  "102" = "Undefined Business",
  "108" = "Undefined Business"
)
# Change Labels for Cleaned Dataset
bondora_clean$UseOfLoan_factor <- loan_use_labels[
  as.character(bondora_clean$UseOfLoan)]

# Change Labels for Uncleaned Dataset
bondora_test$UseOfLoan_factor <- loan_use_labels[
  as.character(bondora_test$UseOfLoan)]

# Add labels to the OccupationArea Variable
levels(as.factor(bondora_clean$OccupationArea)) # view codes

occupation_labels <- c(
  "-1" = "NA",
  "1" = "Other",
  "2" = "Mining",
  "3" = "Processing",
  "4" = "Energy",
  "5" = "Utilities",
  "6" = "Construction",
  "7" = "Retail",
  "8" = "Transport",
  "9" = "Hospitality",
  "10" = "Info/Telcom",
  "11" = "Finance",
  "12" = "Realestate",
  "13" = "Research",
  "14" = "Admin",
  "15" = "Civil/Mil",
  "16" = "Education",
  "17" = "Healthcare",
  "18" = "Social",
  "19" = "Art/Ent",
  "20" = "Agriculture",
  "21" = "Forestry/Fish"
)
# store original
bondora_clean$occupation_code <- bondora_clean$OccupationArea
bondora_clean$occupation_label <- occupation_labels[
  as.character(bondora_clean$OccupationArea)]

bondora_test$occupation_code <- bondora_test$OccupationArea
bondora_test$occupation_label <- occupation_labels[
  as.character(bondora_test$OccupationArea)]
# ---------------------------------------------------------------------------- #
# 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) {

  par(mfrow=c(1,3))

  hist(df1[[col_name]], xlab=type, col=cols[15], main="", breaks=10)
  mtext("Initial Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  hist(df2[[col_name]], xlab=type, col=cols[15], main="", breaks=10)
  mtext("Reduced Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  qqplot(df1[[col_name]], df2[[col_name]], main="", cex=1,
         xlab="Full Sample", ylab="Subsample", line=0.25)
  abline(0, 1, lty=2)

  mtext("QQ Plot", side=3, adj=0, line=0.25, cex=1, font=2)

  mtext(type, outer = TRUE, line = -2, side=3, cex = 1.3, font = 2)

  # Reset plot window
  par(mfrow=c(1,1), mar=c(5,4,4,2)+0.1)
}

plot_desc_bar <- function(df1, df2, col_name, type) {

  par(mfrow=c(1,2), mar=c(5,10,4,2))

  barplot(sort(table(df1[[col_name]]), decreasing = F),
          xlab=type, col=cols[15], horiz=TRUE, las=1)
  mtext("Initial Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  barplot(sort(table(df2[[col_name]]), decreasing = F),
          xlab=type, col=cols[15], horiz=TRUE, las=1)
  mtext("Reduced Sample", side=3, adj=0, line=0.25, cex=1, font=2)

  mtext(type, outer = TRUE, line = -2, side=3, cex = 1.3, font = 2)

  # Reset plot window
  par(mfrow=c(1,1), mar=c(5,4,4,2)+0.1)
```

```r
}

plot_desc_hists(bondora_test, bondora_clean, "Amount","Amount")
save_plot("hist_amt")

plot_desc_hists(bondora_test, bondora_clean, "Interest","Interest")
save_plot("hist_int")

plot_desc_hists(bondora_test, bondora_clean, "LoanDuration","Loan Duration")
save_plot("hist_loandur")

plot_desc_hists(bondora_test, bondora_clean, "MonthlyPayment","Monthly Payment")
save_plot("hist_monpmt")

plot_desc_hists(bondora_test, bondora_clean, "Age","Age")
save_plot("hist_age")

plot_desc_bar(bondora_test, bondora_clean,"UseOfLoan_factor","Loan Purpose")
save_plot("bar_loanuse")

plot_desc_bar(bondora_test, bondora_clean,"Rating","Credit Rating")
save_plot("bar_rating")

plot_desc_bar(bondora_test, bondora_clean,"occupation_label","Occupation")
save_plot("bar_occupation")

# Get Tabular Summary Statistics
tab_comps <- function(df1, df2, cols) {
  stats <- c("Mean"=mean, "Median"=median, "Std. Dev."=sd, "Min"=min, "Max"=max)

  get_stats <- function(d) {
    t(sapply(d[cols], function(x)
      sapply(stats, function(f) round(f(x, na.rm=TRUE), 2))
    ))
  }

  df1_stats <- get_stats(df1)
  df2_stats <- get_stats(df2)

  out <- rbind(
    cbind(DataFrame = "Initial Sample",
          Variable = rownames(df1_stats), df1_stats),
    cbind(DataFrame = "Reduced Sample",
          Variable = rownames(df2_stats), df2_stats)
  )
  rownames(out) <- NULL
  as.data.frame(out)
}

tab_results <- tab_comps(bondora_test, bondora_clean,
                         c("Amount","Interest","Age","LoanDuration"))

knitr::kable(tab_results)

# Save table for use in the Report
saveRDS(tab_results,
        file=paste0(obj_paths,"preprocessing/","summary_table.Rds"))
# ---------------------------------------------------------------------------- #
# Convert Dataset into Incidence Matrix to form Network Object (for ERGM)
bondora_slim <- bondora_clean

# Create the Incidence Matrix for Use of Loan
bondora_matrix <- table(
  bondora_slim$UserName, bondora_slim$UseOfLoan)
bondora_matrix[bondora_matrix > 0] <- 1 # Given that ergm.counts fails with GOF

```

```
# Create network object with counts as Edge attribute
bondora_net <- network::network(
  bondora_matrix, directed=FALSE, bipartite=nrow(bondora_matrix),
  ignore.eval = FALSE, names.eval="frequency", loops=FALSE)

# Set the bipartite Attribute - UNNCESSARY GIVEN bipartite=length(n)
len <- dim(bondora_matrix)[1]
len_b2 <- dim(bondora_matrix)[2]
b_indicator <- c(rep(1,len),rep(2,len_b2))

# Extract Partition 2 Labels
loan_use <- levels(bondora_clean$UseOfLoan_factor)

# Create Loan Type Attribute for Partition 2
b2_loantype <- rep(NA, len)
b2_loantype <- c(b2_loantype, loan_use)

if (length(b2_loantype) == network::network.size(bondora_net)) {
  network::set.vertex.attribute(
    bondora_net, "b2_loantype", value = b2_loantype)
  }

# Add Age Vertex Attribute to B1
age <- bondora_clean$Age[match(
  rownames(bondora_matrix), bondora_clean$UserName)]
b1_age <- c(age, rep(NA, len_b2))
network::set.vertex.attribute(
  bondora_net, "b1_age", value=b1_age
)

# Add Gender Vertex Attribute to B1
gender <- bondora_clean$Gender[match(
  rownames(bondora_matrix), bondora_clean$UserName)]
unique(gender) # Check to see if encoded properly
gender <- ifelse(gender == 0, "male","female")
b1_gender <- c(gender, rep(NA, len_b2))
network::set.vertex.attribute(
  bondora_net, "b1_gender", value = b1_gender
)

# Save the network and data frame object
saveRDS(bondora_slim, file=paste0(obj_paths,"preprocessing/","bondora_df.Rds"))
saveRDS(bondora_net, file=paste0(obj_paths,"preprocessing/","bondora_net.Rds"))
# ---------------------------------------------------------------------------- #
# First, create Incidence Matrix again but with Frequency Counts
loan_use_matrix <- table(
  bondora_slim$UserName, bondora_slim$UseOfLoan)

# Get the Adjacency Matrix for Loan Use Similarity (Dependent QAP Variable)
adj_mat_loan_use <- loan_use_matrix %*% t(loan_use_matrix)
# Remove self weights to remove any loops
diag(adj_mat_loan_use) <- 0

# Get the Adjacency Matrix for Credit Rating Similarity (Predictor in QAP)
incidence_rating <- table(bondora_slim$UserName, bondora_slim$Rating)
adj_mat_rating <- incidence_rating %*% t(incidence_rating)
diag(adj_mat_rating) <- 0

# Get Adjacency Matrix for Occupation Similarity (Predictor in QAP, Binary)
incidence_occupation <- table(bondora_slim$UserName,
                              bondora_slim$occupation_label)
adj_mat_occupation <- incidence_occupation %*% t(incidence_occupation)
adj_mat_occupation[adj_mat_occupation > 0] <- 1
diag(adj_mat_occupation) <- 0

# Get the Adjacency Matrix for Loan Amount (Control in QAP) - BINARY
```

```r
# First, bin the Loan Amounts
bondora_slim$Amount_bins <- cut(
  bondora_slim$Amount, breaks=c(0,2000,4000,6000,8000,10000),
  labels = c(1:5)
)
incidence_amount_bins <- table(bondora_slim$UserName, bondora_slim$Amount_bins)
adj_mat_amount_bins <- incidence_amount_bins %*% t(incidence_amount_bins)
diag(adj_mat_amount_bins) <- 0

# Continous Absolute Difference Approach
avg_amount <- tapply(bondora_slim$Amount, bondora_slim$UserName, mean)
adj_mat_amount_diff <- outer(avg_amount, avg_amount,
                             FUN = function(x,y) abs(x - y))
diag(adj_mat_amount_diff) <- 0

# Get Matrix for Differences in Age
incidence_age <- table(bondora_slim$UserName, bondora_slim$Age)
borrower_ages <- as.numeric(colnames(incidence_age)[max.col(incidence_age)])
names(borrower_ages) <- rownames(incidence_age)
adj_mat_age <- outer(borrower_ages, borrower_ages,
                     FUN = function(x, y) abs(x - y))
rownames(adj_mat_age) <- colnames(adj_mat_age) <- names(borrower_ages)

# Get Adjacency Matrix for (same) Gender
incidence_gender <- table(bondora_slim$UserName, bondora_slim$Gender)
adj_mat_gender <- incidence_gender %*% t(incidence_gender)
# Make the matrix binary for homophily
adj_mat_gender <- ifelse(adj_mat_gender > 0, 1, 0)
diag(adj_mat_gender) <- 0

# Get Adjacency Matrix for Differences in Average Loan Duration
borrower_loandur <- sapply(tapply(bondora_slim$LoanDuration,
                                  bondora_slim$UserName, unique),
                           mean)
adj_mat_loandur_diff <- outer(borrower_loandur, borrower_loandur,
                              FUN=function(x,y) abs(x-y))
diag(adj_mat_loandur_diff) <- 0

# Get Adjacency Matrix for Homophily in Restructure of Loans
incidence_restructure <- table(bondora_slim$UserName, bondora_slim$Restructured)
adj_mat_rest <- incidence_restructure %*% t(incidence_restructure)
diag(adj_mat_rest) <- 0

# Save the objects for the QAP Regression in different Script
qap_paths = paste0(obj_paths,"/qap/")
saveRDS(b_indicator, file=paste0(
  "resources/objects/preprocessing/indicator.Rds"))

saveRDS(adj_mat_loan_use, file=paste0(qap_paths,"adj_mat_loanuse.Rds"))
saveRDS(adj_mat_occupation, file=paste0(qap_paths,"adj_mat_occup.Rds"))
saveRDS(adj_mat_rating, file=paste0(qap_paths,"adj_mat_rating.Rds"))
saveRDS(adj_mat_amount_diff, file=paste0(qap_paths,"adj_mat_amtdiffs.Rds"))
saveRDS(adj_mat_age, file=paste0(qap_paths,"adj_mat_agediffs.Rds"))
saveRDS(adj_mat_gender, file=paste0(qap_paths,"adj_mat_gender.Rds"))
saveRDS(adj_mat_loandur_diff, file=paste0(qap_paths,"adj_mat_loandurdiffs.Rds"))
saveRDS(adj_mat_rest, file=paste0(qap_paths,"adj_mat_rest.Rds"))
# ---------------------------------------------------------------------------- #
```

scripts/bondora_preprocessing.R

A.4 Source Code - QAP Linear Regression

```r
# ---------------------------------------------------------------------------- #
# NOTE: NOT THE LATEST QAP ANALYSIS!

# If not already Installed
install.packages("viridis") # For Colours
install.packages("here")    # To locate files from RProj

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

# Ensure repeatability
set.seed(42)
# ---------------------------------------------------------------------------- #
# Load Relevant Files
qap_path="resources/objects/qap/"

loan_use_mat <- readRDS(paste0(qap_path,"adj_mat_loanuse.RDS"))
rating_mat <- readRDS(paste0(qap_path,"adj_mat_rating.RDS"))
amt_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_amtdiffs.RDS"))
age_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_agediffs.RDS"))
gender_mat <- readRDS(paste0(qap_path,"adj_mat_gender.RDS"))
loandur_diffs_mat <- readRDS(paste0(qap_path,"adj_mat_loandurdiffs.RDS"))
rest_mat <- readRDS(paste0(qap_path,"adj_mat_rest.RDS"))
occup_mat <- readRDS(paste0(qap_path,"adj_mat_occup.Rds"))
# ---------------------------------------------------------------------------- #
# Create function to determine significance from t-value statistic
t_to_stars <- function(t) {
  stars <- rep("", length(t))
  stars[abs(t) >= 1.96]  <- "*"
  stars[abs(t) >= 2.576] <- "**"
  stars[abs(t) >= 3.291] <- "***"

  return(stars)
}

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

# Make function to run QAPs automatically
run_QAP <- function(y_mat, xlist, varnames, model_name) {

  # Run QAP Linear Regression
  model <- sna::netlm(y = y_mat, x = xlist, nullhyp = "qapspp", reps = 1000)
  model$names <- varnames
  summary(model)

  # Obtain results from Model and Name them
  results <- model$coefficients
  names(results) <- varnames

  # Add significance stars to model results
  results_sig <- paste(round(results,3), t_to_stars(model$tstat))

  # Save Model
  saveRDS(model, here::here("resources","objects","qap",
                            paste0(model_name,".Rds")))

  # Prepare function output
  list_names <- c("model","results","results_sig")
  items <- list(model, results, results_sig)
  names(items) <- list_names

  return(items)
}

var_names <- c("Intercept", "Rating", "Occupation","Loan Amount", "Age",
               "Gender", "Loan Duration", "Restructured")
main_pred_names <- c("Intercept", "Rating", "Occupation")
main_pred_vars <- list(rating_mat, occup_mat)
all_pred_vars <- list(rating_mat, occup_mat, amt_diffs_mat, age_diffs_mat,
                  gender_mat, loandur_diffs_mat, rest_mat)
# --------------------------------------------------------------------------- #
# Basic QAP Linear Regression 1 - Unstandardised + No Controls

qap_m1 <- run_QAP(loan_use_mat, main_pred_vars, main_pred_names, "qap_m1")
summary(qap_m1$model)

# --------------------------------------------------------------------------- #
# Basic QAP Linear Regression 1 - Standardised + No Controls
scaled_dep <- scale(loan_use_mat)
scaled_pred <- lapply(main_pred_vars, scale)

qap_m2 <- run_QAP(scaled_dep, scaled_pred, main_pred_names, "qap_m2")
summary(qap_m2$model)

# --------------------------------------------------------------------------- #
# Plot the result for Model 1 Unstandardised vs Standardised

par(mfrow=c(1,2))

qap_plot_m1 <- barplot(results_m1, col = cols[15],  border = cols[10],
                       ylim = c(min(results_m1) + min(results_m1)*0.15,
                                max(results_m1) + max(results_m1)*0.15),
                       main="QAP Model Results (Unstandardised)")
text(x = qap_plot_m1,
     y = results_m1 + sign(results_m1)*(0.075*diff(range(results_m1))),
     labels = results_sig_m1, font = 2)


qap_plot_m2 <- barplot(results_m2, col = cols[15],  border = cols[10],
                       ylim = c(min(results_m2) + min(results_m2)*0.15,
                                max(results_m2) + max(results_m2)*0.15),
                       main="QAP Model Results (Standardised)")
text(x = qap_plot_m2,
     y = results_m2 + sign(results_m2)*(0.075*diff(range(results_m2))),
     labels = results_sig_m2, font = 2)

save_qap_plot("unstd_std_plot")

par(mfrow=c(1,1))
# --------------------------------------------------------------------------- #
# Basic QAP Linear Regression 2 - Unstandardised + Controls

qap_m3 <- run_QAP(loan_use_mat, all_pred_vars, var_names, "qap_m3")
summary(qap_m3$model)

# --------------------------------------------------------------------------- #
# Basic QAP Linear Regression 2 - Standardised + Controls
scaled_pred_2 <- lapply(all_pred_vars, scale)

qap_m4 <- run_QAP(scaled_dep, scaled_pred_2, var_names, "qap_m4")
summary(qap_m4$model)

# --------------------------------------------------------------------------- #
```

scripts/qap_network_analysis.R

A.5 Source Code - ERGM Network Analysis

```r
# ---------------------------------------------------------------------------- #
# NOTE: NOT THE LATEST ERGM ANALYSIS! FOR REFERENCE ONLY

# If not already Installed
install.packages("viridis")     # For Colours
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"))

# Repeatability
set.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 <- sna::degree(bondora_plot)
counts_att <- ifelse(type_indicator, log(counts)*4, counts*2.5)
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(0.15, 0.01),
       col = c(cols[30], cols[5]),
       pch = c(16, 15),
       title = "Node Partitions",
       title.font = 2,
       cex = 1,              # Increase the text size
       pt.cex = 2,             # Increase the point symbol size
       box.lwd = 1,            # Thin box border
       box.col = "black",      # Box color
       bty = "o"               # Use a box around legend
)
save_ergm_plot("network_plot")

# Summary Statistics and Save Them
density <- snafun::g_density(bondora_net)[1]
centralization <- snafun::g_centralize(bondora_net)[1]
vertices <- snafun::count_vertices(bondora_net)[1]
edges <- snafun::count_edges(bondora_net)[1]
dist <- snafun::g_mean_distance(bondora_net)[1]

net_names <- c("Vertex Count","Edge Count","Density","Centralization",
               "Mean Distance")
net_stats <- c(vertices, edges, density, centralization, dist)
net_stats <- sapply(net_stats, function(x) round(as.numeric(x),2))

net_summary <- data.frame(Statistic = net_names,
                          "Measure" = net_stats)
knitr::kable(net_summary)
saveRDS(net_summary, here::here("resources","objects","ergm","net_summary.Rds"))

# Plot Network Summary Statistics
snafun::plot_centralities(bondora_net)
save_ergm_plot("network_plots")
# --------------------------------------------------------------------------- #
# Make function to calculate probabilities from log odds
lodds_to_prob <- function(l_odd) {
  return(exp(l_odd) / (1 + exp(l_odd)))
}
# Make function to save ERGM object
save_ergm <- function(object, id) {
  saveRDS(object, file=here::here(
    "resources","objects","ergm",id,".Rds"))
}
# Make function to conduct ERGMs automatically
auto_ergm <- function(model, mcmc, name) {

  # Conducts the GOF Diagnostics and then saves the model,
  # mcmc diagnostics and gof object in a list.
  # This list can be imported as an .RDS object into the R environment

  # Diagnostics
  if (mcmc) {
    ergm::mcmc.diagnostics(model)
  }

  # The GOF must be adjusted otherwise it takes too long
  # We do not limit the GOF by changing its range of parameters
  gof <- ergm::gof(model,
                   control = ergm::control.gof.ergm(
                     nsim = 200,
                     MCMC.burnin = 5000,
                     MCMC.interval = 1000,
                     parallel = n_cores,
                     parallel.type = "PSOCK"
                   ))

  # Return List to view each item separately
  result <- list(model, gof)
  names(result) <- c("model","gof")
  save_ergm(result, paste0(name,"_panel"))

  return(result)
}
# -------------------------------------------------------------------------- #
# Find max degree
(max_deg <- max(summary(bondora_net ~ b2factor("b2_loantype"))))
# -------------------------------------------------------------------------- #
# Base Model + GOF
formula_base_model <- bondora_net ~ edges
base_ergm <- ergm::ergm(formula_base_model)
base_ergm_panel <- auto_ergm(base_ergm, mcmc = FALSE, name = "ergm_base")
snafun::stat_plot_gof(base_ergm_panel$gof)
models = list(base_ergm)
texreg::screenreg(models)
# -------------------------------------------------------------------------- #
# Base Model + Edge Counts + GOF
#base_model_counts <- ergm::ergm(bondora_net ~ edges, response="frequency",
#                                reference = ~ Poisson)
#basemodel_counts_gof <- ergm::gof(base_model_counts)
#snafun::stat_plot_gof(basemodel_counts_gof)
#
#texreg::screenreg(list(base_model, base_model_counts))
# -------------------------------------------------------------------------- #
# Iteration 1 + MCMC Diagnostics + GOF
model_1_params <- bondora_net ~ edges +
  # b1 decay can be very low since 9 b2
  gwb1degree(decay=0.15, fixed=TRUE)

model_1 <- ergm::ergm(
  model_1_params,

  # Max b2 degree is 72, so this constraint is reasonable
  # and helps convergence significantly.
  # Technically in the Bondora population this can be
  # far higher but we are studying a subsample.
  #constraints = ~ bd(minout = 0, maxout = 80),

  control = ergm::control.ergm(
    # Greater burn-in for cleaner result
    MCMC.burnin = 20000,
    # Greater sample size for greater stability
    MCMC.samplesize = 100000,
    seed = 42,
    MCMC.interval = 1000,
    # Only needed for convergence pvals to improve
    MCMLE.maxit = 45,
    # Smaller steps for stability
    MCMLE.steplength = 0.25,
    parallel = n_cores,
    parallel.type = "PSOCK"
  )
)

model_1_panel <- auto_ergm(model=model_1, mcmc=TRUE, name="ergm_m1")
model_1_panel$gof
snafun::stat_plot_gof(model_1_panel$gof)
texreg::screenreg(list(base_ergm, model_1))
# -------------------------------------------------------------------------- #
# Iteration 2 + MCMC Diagnostics + GOF
model_2_params <- bondora_net ~ edges +
  # low decay important because there is high clustering around low degrees
  gwb1degree(decay=0.15, fixed=TRUE) +
  # decay should be higher due to wider variation in degree but
  # too high of degree makes the traces concentrated around the tails.
  gwb1dsp(decay=0.5, fixed=TRUE)

model_2 <- ergm::ergm(
  model_2_params,

  # Max b2 degree is 72, so this constraint is reasonable
  # and helps convergence significantly.
  # Technically in the Bondora population this can be
  # far higher but we are studying a subsample.
  constraints = ~ bd(minout = 0, maxout = 80),

  control = ergm::control.ergm(
    # Greater burn-in for cleaner result
    MCMC.burnin = 20000,
    # Greater sample size for greater stability
    MCMC.samplesize = 100000,
    seed = 42,
    MCMC.interval = 1000,
    # Only needed for convergence pvals to improve
    MCMLE.maxit = 45,
    # Smaller steps for stability
    MCMLE.steplength = 0.25,
    parallel = n_cores,
    parallel.type = "PSOCK"
    )
  )

model_2_panel <- auto_ergm(model=model_2, mcmc=TRUE, name="ergm_m2")
snafun::stat_plot_gof(model_2_panel$gof)
model_2_panel$gof
models <- list(base_ergm, model_1, model_2)
texreg::screenreg(models)
# --------------------------------------------------------------------------- #
# Iteration 3 + MCMC Diagnostics + GOF
model_3_params <- bondora_net ~ edges +
  # low decay important because there is high clustering around low degrees
  gwb1degree(decay=0.15, fixed=TRUE) +
  # decay should be higher due to wider variation in degree but
  # too high of degree makes the traces concentrated around the tails.
  gwb1dsp(decay=0.5, fixed=TRUE) +
  # See differences across genders (implicitly, since b1nodemix unavailable)
  b1nodematch("b1_gender", diff=FALSE)

model_3 <- ergm::ergm(
  model_3_params,

  # Max b2 degree is 72, so this constraint is reasonable
  # and helps convergence significantly.
  # Technically in the Bondora population this can be
  # far higher but we are studying a subsample.
  constraints = ~ bd(minout = 0, maxout = 80),

  control = ergm::control.ergm(
    # Greater burn-in for cleaner result
    MCMC.burnin = 20000,
    # Greater sample size for greater stability
    MCMC.samplesize = 100000,
    seed = 42,
    MCMC.interval = 1000,
    # Only needed for convergence pvals to improve
    MCMLE.maxit = 45,
    # Smaller steps for stability
    MCMLE.steplength = 0.25,
    parallel = n_cores,
```

```r
    parallel.type = "PSOCK"
  )
)

model_3_panel <- auto_ergm(model=model_3, mcmc=TRUE, name="ergm_m3")
snafun::stat_plot_gof(model_3_panel$gof)
model_3_panel$gof
models <- list(base_ergm, model_1, model_2, model_3)
texreg::screenreg(models)
# ---------------------------------------------------------------------------- #
# Iteration 4 + MCMC Diagnostics + GOF
model_4_params <- bondora_net ~ edges +
  # low decay important because there is high clustering around low degrees
  gwb1degree(decay=0.15, fixed=TRUE) +
  # decay should be higher due to wider variation in degree but
  # too high of degree makes the traces concentrated around the tails.
  gwb1dsp(decay=0.5, fixed=TRUE) +
  # See differences across genders (implicitly, since b1nodemix unavailable)
  b1nodematch("b1_gender", diff=TRUE) +
  # See if higher ages make a difference
  b1cov("b1_age")

model_4 <- ergm::ergm(
  model_4_params,

  # Max b2 degree is 72, so this constraint is reasonable
  # and helps convergence significantly.
  # Technically in the Bondora population this can be
  # far higher but we are studying a subsample.
  constraints = ~ bd(minout = 0, maxout = 80),

  control = ergm::control.ergm(
    # Greater burn-in for cleaner result
    MCMC.burnin = 20000,
    # Greater sample size for greater stability
    MCMC.samplesize = 100000,
    seed = 42,
    MCMC.interval = 1000,
    # Only needed for convergence pvals to improve
    MCMLE.maxit = 45,
    # Smaller steps for stability
    MCMLE.steplength = 0.25,
    parallel = n_cores,
    parallel.type = "PSOCK"
  )
)

model_4_panel <- auto_ergm(model=model_4, mcmc=TRUE, name="ergm_m4")
model_4_panel$gof
snafun::stat_plot_gof(model_4_panel$gof)
models <- list(base_ergm, model_1, model_2, model_3, model_4)
texreg::screenreg(models)
# ---------------------------------------------------------------------------- #
```

scripts/ergm_network_analysis.R

References

Agrawal, V. (2012). Managing the diversified team: Challenges and strategies for improving performance. *Team Performance Management: An International Journal*, *18*(7), 384–400. https://doi.org/10.1108/13527591211281129

Aliano, M., Alnabulsi, K., Cestari, G., & Ragni, S. (2023). The role of gender and education in peer-to-peer lending activities: Evidence from a european cross-country study. *European Scientific Journal ESJ*, *2*. https://doi.org/10.19044/esipreprint.2.2023.p95

Alistair Milne, & Paul Parboteeah. (2017). *The business models and economics of peer-to-peer lending*. Retrieved from https://repository.lboro.ac.uk/articles/online_resource/The_business_models_and_economics_of_peer-to-peer_lending/9494891

Amar, M., Ariely, D., Ayal, S., Cryder, C. E., & Rick, S. I. (2011). Winning the battle but losing the war: The psychology of debt management. *Journal of Marketing Research*, *48*, S38–S50.

Ayal, S., Bar-Haim, D., & Ofir, M. (2018). Behavioral biases in peer-to-peer (P2P) lending. *Forthcoming in Behavioral Finance: The Coming of Age (Venezia I. Ed., World Scientific Publishers)*.

Ayal, S., Hochman, G., & Zakay, D. (2011). Two sides of the same coin: Information processing style and reverse biases. *Judgment and Decision Making*, *6*(4), 295–305. https://doi.org/10.1017/S193029750000190X

Ayal, S., & Zakay, D. (2009). The perceived diversity heuristic: The case of pseudodiversity. *Journal of Personality and Social Psychology*, *96*(3), 559–573. https://doi.org/10.1037/a0013906

Bondora. (2024, August 28). How are bondora risk ratings calculated? Bondora help center. Retrieved November 13, 2025, from https://help.bondora.com/hc/en-us/articles/14814705732881-How-are-Bondora-risk-ratings-calculated

Cooper, D., Gorbachev, O., & LuengoPrado, M. J. (2023). Consumption, credit, and the missing young. *Journal of Money, Credit and Banking*, *55*(2), 379–405.

Croson, R., & Gneezy, U. (2009). Gender differences in preferences. *Journal of Economic Literature*, *47*(2), 448–474.

Galak, J., Small, D. A., & Stephen, A. T. (2010). Micro-finance decision making: A field study of prosocial lending. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1634949

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

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 263–291. https://doi.org/10.2307/1914185

Lee, E., & Lee, B. (2012). Herding behavior in online P2P lending: An empirical investigation. *Electronic Commerce Research and Applications*, *11*(5), 495–503. https://doi.org/10.1016/j.elerap.2012.02.001

Liu, Y., Baals, L. J., Osterrieder, J., & Hadji-Misheva, B. (2024). Network centrality and credit risk: A comprehensive analysis of peer-to-peer lending dynamics. *Finance Research Letters*, *63*, 105308. https://doi.org/10.1016/j.frl.2024.105308

Manu Siddhartha. (2021). Bondora peer to peer lending loan data. Kaggle. Retrieved November 12, 2025, from https://www.kaggle.com/datasets/sid321axn/bondora-peer-to-peer-lending-loan-data

Ravina, E. (2019). Love & loans: The effect of beauty and personal characteristics in credit markets. *Available at SSRN 1107307*.

Serrano-Cinca, C., Gutiérrez-Nieto, B., & López-Palacios, L. (2015). Determinants of default in

P2P lending. *PLOS ONE*, *10*(10), e0139427. https://doi.org/10.1371/journal.pone.0139427

Simpson, W. (2001). QAP: The quadratic assignment procedure. *North American STATA Users' Group Meeting*, *1*, 1–17.

Stivala, A., Wang, P., & Lomi, A. (2025). *Improving exponential-family random graph models for bipartite networks*. arXiv. https://doi.org/10.48550/ARXIV.2502.01892

Yao, J., Chen, J., Wei, J., Chen, Y., & Yang, S. (2019). The relationship between soft information in loan titles and online peer-to-peer lending: Evidence from RenRenDai platform. *Electronic Commerce Research*, *19*(1), 111–129. https://doi.org/10.1007/s10660-018-9293-z

Zuo, Y. (n.d.). *CLUSTERING ANALYSIS TO SUPPORT LENDER's DECISION-MAKING IN P2P LENDING*.

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. Additionally, some R functions written by the group were improved with ChatGPT's suggestions. No AI tool was utilised to independently write parts of the report. The following parts of the assignment were affected/generated by AI tool usage: **INTRODUCTION;** The tool was utilised to evaluate the validity of certain theoretical concepts and refine them. **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, **Group 7** 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, **Group 7** takes full responsibility for the content of their work.