# The One with the Social Network Analysis: the extraction, analysis and modelling of temporal social networks from narratives

Michelle Edwards

July 16, 2019

*Thesis submitted for the degree of*
*Master of Philosophy*
*in*
*Applied Mathematics*
*at The University of Adelaide*
*Faculty of Engineering, Computer and Mathematical Sciences*
*School of Mathematical Sciences*


THE UNIVERSITY
*of* ADELAIDE

# Contents

# List of Tables

# List of Figures

# Abstract

Narratives tell us about the people, cultures, and time periods in and about which they were written. Therefore, narrative analysis is a powerful tool for understanding culture. One way to analyse narratives is through their social networks, however extracting the network is a complex task. Manually recording characters and their interactions is an accurate, but time consuming method for narrative social network extraction, however efficient automatic extraction methods may introduce errors.

In this thesis, we perform a detailed comparative study of narrative social network extraction techniques, and investigate the effect the techniques have on the analysis of the narrative. We use the 1994–2004 television series *Friends* as a case study to model and compare extraction techniques. By designing a simulated social network and observation processes resembling different network extraction techniques, we find that automated network extraction methods are reliable for computing many network metrics, but can distort the clustering coefficient. Our comparison of extraction techniques allows for many more narratives to be extracted and analysed efficiently.

We also analyse and model the social networks of *Friends*, to gain new insights into the the series, and what made it successful. We show which are the most important characters and relationships, and through modelling social network features we find the most informative features to predict success. Our analysis of *Friends* provides and example and a building block for deeper understanding about particular narratives and narratives in general.

# Signed Statement

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signed: ..... ..... Date: 15/6/2019

# Acknowledgements

First and foremost, I wish to thank my supervisors – Prof. Matthew Roughan, Dr. Lewis Mitchell and Dr. Jono Tuke for their endless ideas, advice and editing, as well as the coffees, chats and banter. I hope this thesis will make you proud, and that you are now inspired to watch *Friends*.

Many aspects of my thesis would not have been possible without Prof. Ana Bazzan's dataset of manually recorded interactions for *Friends*, so I thank her for her time spent collecting the data and allowing me to use it for my research.

To the other post-graduate students in the Adelaide University School of Mathematical Sciences – thank you for the Cake Mondays, Burrito Fridays, games afternoons, company over lunchtime, and everything in between. In particular, I would like to thank Matt, who was always up for going for a walk and listening to me complain about problems with my code.

Thank you to my friends outside of the School of Mathematical Sciences for being interested in my research, even if you didn't understand all of it, and for reminding me to take breaks by inviting me to brunches, board games nights and catch-ups.

I wish to thank my family for supporting and encouraging me for my whole life, and especially for feeding me and being understanding when I came home late and was too busy to help out. The family dinners and Potato Wednesdays have played an important part in getting me through my Masters degree.

Finally, I want to thank Jonno for celebrating with me on my good days, putting up with me on the bad days and always being interested in my work. You have been incredibly supportive and I don't know how I would have done it without you.

# Dedication

This thesis is dedicated to my parents, who have raised me and supported me in whatever I do for 23 years, and to my fiancé who has been there for me through the best and worst times in my studies.

# Chapter 1

# Introduction

"How you doin'?"

*Matt LeBlanc* as *Joey Tribbiani*
*Season 8, Episode 19*

## 1.1 Overview

Narratives, or stories, are reports of real or fictional events. They can be presented with words, images, or both, and are an important aspect of human behaviour. Yuval Noah Harari, in his book *Sapiens* [61], claims that humans were able to co-operate in large numbers due to their ability to tell stories and believe in fiction. Today, narratives are just as crucial, and come in many different forms. In Western society it is rare to find someone without a favourite movie, novel or television series.

We watch and read narratives for many different reasons. Non-fiction narratives are informative about historical events, scientific topics and recent news. Fictional narratives can also tell us about the world around us. For example, myths and legends from different cultures inform us about those cultures, and fictional narratives from different time periods inform us about those time periods. Narratives are a powerful way to learn about cultures because they are enjoyable and memorable.

It follows that being able to understand narratives at a deep level is advantageous. However, there are more reasons to analyse narratives. While narrative analysis can help us to understand particular narratives in great detail, and details of different cultures and time periods, it can also help create new narratives, and of better quality.

Most narratives are primarily about their characters and the relationships between them. The narrative introduces the characters and we learn about their interactions. In many cases these characters and interactions define the narrative. In this way, the social structure contains substantial information of the narrative, and so it is reasonable to analyse the narrative by analysing the social structure. An important feature of narratives is their evolution over time, so we analyse the social structure temporally.

By analysing the social structure of a narrative, we may find information that is not obvious in its original form. We can also use mathematical and statistical social network techniques to quantify attributes of the social structure and to identify trends. We use time windows, such as episodes or seasons (for a television series), or paragraphs or chapters (for a novel) to capture the social structure at particular times, and we analyse how relationships and characters change over the course of the narrative.

From here, we can discover elements of the social structure of a narrative that makes it particularly successful. We do this by modelling quantitive features of the relationships and characters in a narrative with some measure of success, such as the rating of a television series.

Before we can model the ratings and other aspects of a narrative's social structure, we must extract and record the social structure from the narrative. A variety of techniques have been developed, but the effect of the different techniques on the analysis of the narrative needs to be considered. In this thesis we discuss and compare common extraction techniques. In particular, extracting a social network manually is accurate, but very time consuming. In contrast, automated network extraction is generally quick, but can introduce errors. We model and compare these errors and find that automated network extraction is reliable for many narrative analyses.

We use the television situation comedy *Friends* as a case study for narrative social network analysis. We find new insights into the series, as well as possible useful insights into narratives in general. For example, we use features of social networks of episodes in the series to predict success, and find the features which increase success. With further analysis of other narrative social networks, generalisation of our results could lead to improvement in new and developing narratives. Improving narratives makes them more enjoyable and memorable, which means they are more likely to stand the test of time.

## 1.2 Preliminary background

In this thesis, we analyse narrative social networks and related extraction techniques from text. Before outlining the thesis, we present important preliminary background information.

Social networks describe the characters and relationships between characters in narratives. A narrative social network is made up of nodes, which represent characters, and edges, which represent interactions between characters. In some social networks, the edges are weighted so that they represent the *number* of interactions between characters.

An important feature of narratives is their evolution over time. Therefore, a social networks of a narrative should have a temporal component. We define social networks temporally by splitting the narrative into time windows. For example, in a television series, the time windows could be episodes, seasons or the entire series. For each time window, we have a single set of characters and interactions, and the time window networks together form the temporal network.

To extract a social network from a narrative, several methods have been developed. An accurate, but time-consuming method is to manually watch or read the narrative and record interactions. We call networks extracted like this **manual** networks. Automatic methods are less time consuming, but can introduce errors. One automatic method is to extract a **co-occurrence** network, where an "interaction" is assumed to occur if characters appear together. For example, in a television show, an interaction could be defined as characters appearing in the same scene.

Once the network has been extracted, we perform analysis using network metrics. Network metrics quantitatively describe the network and its characters. We calculate three types of metrics; *global*, *character* and *relationship* metrics. Global metrics describe the overall structure of the network, which can be used to compare different social networks or changes to temporal networks over time. Character metrics measure attributes such as centrality for each character, which aids analysis of character roles and importance. Relationship metrics measure the importance of the different relationships in the narrative.

As a case study for narrative social network analysis, we use the television series *Friends*, which was aired from 1994 to 2004. *Friends* is a situation comedy about six "friends" living in Manhattan – Chandler, Joey, Monica, Phoebe, Rachel and Ross – and their lives, relationships and careers.

## 1.3 Summary of thesis

In Chapter 2 we discuss previous works in narrative analysis, particularly using social networks. We also define social networks and associated metrics and terms.

In Chapter 3 we describe the process of collecting two datasets of social networks for the television series *Friends*; **manual** networks and **co-occurrence** networks. We use these datasets throughout the thesis.

In Chapter 4 we compare three different narrative social network extraction techniques. We simulate social networks for a narrative and extract three networks from each simulation, based on extraction techniques seen in the literature. We then compare the extracted networks using network metrics, and discuss the effect the extraction technique has on the analysis of the narrative. The content of Chapter 4 has been submitted for a journal.

In Chapter 5 we exhaustively analyse the social networks of *Friends*. We examine the television show using the two social network datasets by calculating global, character and edge metrics. We perform univariate analysis on the network metrics of the series networks, season networks and episode networks. We also perform bivariate analysis on the network metrics of the season and episode networks over time. Our analysis leads to interesting insights into the series about the most important characters and relationships.

In Chapter 6 we model the social networks from *Friends* and their metrics. First, we attempt to fit simple models to the network datasets, which make use of the difference in screen-time between the core characters and the other characters, but find that a suitable model needs to be more complex. We then fit a stochastic block model, which automatically infers the class structure on the network.

We also fit linear models to network metrics over time to find patterns in the data. The patterns we find provide evidence that "the *Friends* get less friendly". Similarly we model the success of the series by fitting a multivariate model to the Internet Movie Database rating using episode network metrics. Our final model shows the relationship between network structure and narrative success.

# Chapter 2

# The One With the Background Information

"They don't know that we know they know we know."

*Lisa Kudrow* as *Phoebe Buffay*
*Season 5, Episode 14*

## 2.1 Narrative analysis

Quantitative narrative analysis has become increasingly popular in recent years with the increasing availability of literary works and film and television scripts online. Narratives can be of the form of films [35, 88, 97, 108, 126], television shows [21, 30, 84, 89] or novels [24, 43, 65, 76, 83, 94, 125]. Reasons to analyse these include:

- to gain a deeper understanding of a particular narrative, narratives of a certain type, or narratives in general;
- to make predictions about what will happen next in a narrative that has not yet been written or released; or
- to help determine what would improve narratives in the future.

### 2.1.1 Understanding the narrative

There are several examples of narrative analyses that shed light on particular narratives in the literature. For example, Min and Park [83] analysed Victor

Hugo's novel *Les Misérables* and found how the presence of different characters and communities affected how happy other characters or communities were.

Prado *et al.* [94] analysed Lewis Carroll's *Alice in Wonderland*, and the anonymous *La Chanson de Roland*. They looked at influential characters at different points and discussed the best ways to conduct such analyses. They find that social networks with a temporal component are more appropriate for narrative analyses than time-independent networks because they capture more features.

While these examples focussed on novels, Tan *et al.* [120] analysed and compared two scientific fiction television series: *Star Trek: The Next Generation* and *Stargate SG-1*. These series seem similar, however *Star Trek* ran for 48 years, compared to *Stargate*'s 18 years. Tan *et al.* found that they are surprisingly similar in the way characters interact with each other, however new characters in *Star Trek* were introduced more than they were in *Stargate*.

Gaining a deeper understanding of particular narratives is also popular within an online blog context where fans of television shows such as *Game of Thrones* [55], *Seinfeld* [118], *The Simpsons* [104], *Grey's Anatomy* [73] and *Friends* [18, 105, 112] are able to visualise data without having detailed mathematical knowledge. Similarly, films [19, 50, 53] and even plays [59, 129] have been analysed quantitatively through this media.

Fans of narratives are interested in which characters have the most dialogue, and how that changes over time [19, 104, 112, 118]. They want to know about the most prominent relationships [18, 83], and the sentiment of the relationships [44, 57, 83, 111] to distinguish between friends and enemies.

Sentiment analysis of narratives has also been used to classify the narratives based on their emotional arcs [97]. Waumans *et al.* [125] also use narrative analysis to classify narratives. They look at how characters interact with each other in novels and attempt to classify the novels into authors, with reasonably successful results. Alternatively, Luczak-Roesch *et al.* [74] created a tool for interactive visualisation of character occurrences in Victorian novels.

Learning more about narratives and mythological characters in general was the purpose of other narrative analyses in the literature. Mac Carron and Kenna [75] analysed three fictional narratives based on varying degrees of facts: *Beowulf*, *Iliad* and *Táin Bó Cualinge* to find the difference between historical narratives and completely fictional narratives. Similarly in a seperate article, they analysed viking sagas and compared the characters and their relationships to real-life relationships [76]. Ricardo *et al.* [17] also compared mythological social structures to real-life social structures, using the characters from the Marvel comic books.

### 2.1.2 Making predictions

Another reason to analyse narratives is to make predictions about what will happen next. This is useful to:

- automatically create or complete narratives to inspire writers [26, 49, 106, 108, 130],
- to allow fans to guess the events of the next season of a television show [66], or
- to summarise documents, such as film scripts, automatically [30, 35, 47, 58, 113].

In particular, Fortuin *et al.* [49] used narrative analysis to inspire script-writers suffering from writer's block. They modelled the sequence of stories and predicted what sequences might come next. These sequences can act as suggestions for writers who are struggling to write some parts of their narrative. Their model also generates visualisations, which help provide deeper understanding to the narrative.

Janosov [66] quantitatively analysed the characters in the television series *Game of Thrones* to predict which character would die next, and probabilities that each character would die, based on the first six seasons. This is interesting for fans of the show waiting for the next season to be released.

Summarising documents automatically is useful for anyone who has to understand large documents within a time limit. For narratives, automatic summarisation can help with creating previews and/or synopses of films, books and television shows. In particular, Gorinski *et al.* [58] analysed film scripts to find a logical chain of important events, with the purpose of generating a shorter version of the script that includes all the important parts. This allows them to summarise film scripts automatically.

Event prediction in narratives also suggests potential methods for predicting real-world events from news texts [60, 72].

### 2.1.3 Improving narratives

Making predictions in narratives means that writers have more ideas at their fingertips, so they can choose the best direction for their narrative. They can also focus on perfecting the finer details instead of struggling for ideas, so in general, narratives may be improved.

Analysing narratives can also indicate what the audience likes in a narrative, and what is missing and needs to be included. For example, researchers [19, 79, 104] have noticed a difference in the appearance of males and females

in narratives. For example, Anderson and Daniels [19]. found that in a vast majority of films, men have at least 60% of the dialogue.

Narrative analysis gives us awareness of issues such as gender imbalance, which is the first step to addressing the issue. However, in this thesis, we are interested in the specific field of social network analysis. Many of the narrative studies in the literature use social networks of characters to analyse narratives. Before we explain how to analyse narratives using social networks, we must define a social network and associated concepts.

## 2.2 Social networks

### 2.2.1 Social network definitions

A network is defined as a set of nodes (or vertices) $V = \{1, \ldots, n\}$, some of which are pairwise connected by edges $E = \{w_{ij} | i, j \in V\}$, where $n$ is the number of nodes in the network [91]. In the field of social network analysis, nodes will usually represent people or characters, and are often called "actors". In narrative social networks, nodes represent characters in the narrative.

An edge in a social network connects two actors if they share some form of interaction, such as a friendship. The edges of a social network are generally called "links". In narrative social networks, there are many possibilities for the types of interactions used to define edges.

Here, we focus only on *simple*, *undirected*, but *weighted* networks. A simple network is one in which a node cannot have an edge to itself, (*i.e.*, no "self-loops"), and there is at most one edge between each pair of nodes.

Undirected networks have undirected edges which represent interactions that are mutual for the characters. For example, "character A and character B are friends" describes a mutual interaction, whereas "character A likes character B" describes a directed interaction, where the edge would go from character A to character B.

In a weighted network, each edge has a number associated with it, called a "weight". The number could indicate a distance or capacity for the edge. In many narrative social networks, the weight is a non-negative integer that represents the number of times characters interact, *e.g.* the number of times characters speak to each other.

### 2.2.2 Metric definitions

Let $G = (V, E)$ be a network with nodes

$$V = \{1, \ldots, n\},$$

and edge weights

$$E = \{w_{ij} \in \mathbb{N} | i, j \in V, i < j\}.$$

Our networks are undirected, so $w_{ij} = w_{ji}\ \forall i, j \in V$. From this definition, we define a non-negative edge weight between every pair of nodes. We take an edge weight of 0 to mean there is no edge. Our networks are loop-free, so $w_{ii} = 0\ \forall i \in V$.

A *path* from node $i$ to node $j$ is a sequence of edges connecting the nodes. We define a *geodesic path* between node $i$ and node $j$ by a path that contains the least number of edges, $d_{ij}$. Note that this definition ignores the edge weights. In our networks, nodes connected with higher edge weights are "closer" to each other, so for a weighted shortest path we would use the reciprocal of the edge weights, however we only use unweighted geodesic paths in this thesis.

A *subgraph* of $G$ is a network $G_{\text{sub}} = (V_{\text{sub}}, E_{\text{sub}})$ such that

$$V_{\text{sub}} \subseteq V, \text{ and } E_{\text{sub}} \subseteq \{w_{ij} \in E | i, j \in V_{\text{sub}}\}.$$

Finally, we define an *adjacency matrix* for $G$ as

$$A = [A_{ij}],$$

where $A_{ij} = w_{ij} \in E$, for $i, j \in V$.

Here we define the commonly used metrics for narrative social network analysis. For the following definitions, define the indicator function

$$I[\text{statement}] = \begin{cases} 1 & \text{if statement is true, and} \\ 0 & \text{if statement is false.} \end{cases}$$

#### Global metrics

- The *size* of $G$ is the number of nodes/characters:

$$|G| = |V| = n.$$

- The *total edge weight* of $G$ is the sum of all edge weights:

$$\sum_{ij} w_{ij}.$$

- The *total number of edges* of $G$ is the number of edges with positive edge weight:
$$\sum_{i<j} I[w_{ij} > 0].$$

- The *density* of $G$ is the proportion of observed edges:
$$\text{dens}(G) = \frac{2}{n(n-1)} \sum_{i<j} I[w_{ij} > 0].$$

- The *average path length* of $G$ is the average of the geodesic path lengths between each pair of nodes:
$$l_G = \frac{2}{n(n-1)} \sum_{i<j} d_{ij}.$$
For an unconnected network (*i.e.*, a network in which there are no paths between some pairs of nodes), we only calculate and average geodesic path lengths of nodes that have a path between them.

- The *diameter* of $G$ is the longest geodesic path:
$$d = \max_{i,j \in V, d_{ij} < \infty} d_{ij}.$$

- The *clustering coefficient* of $G$ is
$$C(G) = \frac{(\text{number of triangles}) \times 3}{(\text{number of connected triples})},$$
where a triangles is three mutually adjacent vertices and a connected triple is three vertices with at least two edges connecting the three.

- The *number of connected components* of $G$ is the minimum number of subgraphs of $G$ where every node in a subgraph is connected (*i.e.*, there is at least one path between the nodes) to every other node in the subgraph by some sequence of edges.

**Character metrics**

- The *degree* of node $i$ is the number of adjacent edges:
$$k_i = \sum_j I[w_{ij} > 0].$$

- The *normalised degree* of node $i$ is the number of adjacent edges, divided by the maximum possible degree:

$$\text{normdeg}(i) = \frac{1}{n-1} \sum_j I[w_{ij} > 0].$$

- The *weighted degree* of node $i$ is the sum of the weights of the adjacent edges:

$$\text{weighteddeg}(i) = \sum_j w_{ij}.$$

- The *betweenness centrality* of node $i$ is

$$\text{between}(i) = \sum_{s,t} \frac{n_{st}^i}{g_{st}},$$

  where $n_{st}^i$ is the number of geodesic paths from node $s$ to $t$ that go through node $i$ and $g_{st}$ is the total number of geodesic paths from node $s$ to $t$.

- The *closeness centrality* of node $i$ is the inverse of the average length of geodesic paths from node $i$:

$$\text{close}(i) = \frac{1}{\sum_j d_{ij}/n}.$$

- The *eigenvector centrality* of node $i$ is the $i$th element of the eigenvector corresponding to the largest eigenvalue of the adjacency matrix $A$ of $G$.

- The *local clustering coefficient* of node $i$ is the proportion of triangles centred on node $i$ that are closed:

$$C(i) = \frac{2|\{w_{jk} \in E | j,k \in N_i, w_{jk} > 0\}|}{k_i(k_i-1)},$$

  where $N_i$ is the set of nodes connected to node $i$ by an edge (*i.e.*, the neighbourhood of $i$) and $k_i$ is the number of neighbours of node $i$ (or unweighted degree). This is sometimes referred to as the *transitivity* of node $i$.

**Relationship metrics**

- The *edge weight* of the edge from node $i$ to node $j$ is $w_{ij}$. In our narrative social network context $w_{ij}$ represents the number of interactions that occur between node $i$ and node $j$, so it must be a non-negative integer.

## 2.3 Social networks in narratives

Narratives are stories about characters and their interactions [58, 83], so it makes sense to analyse narratives using social networks. In fact, many of the examples in literature of narrative analysis use social networks.

Some researchers look at global metrics to obtain general results about the narrative [36, 46, 83, 88]. For example, Chen *et al.* [36] found that the social network of Cao Xueqin's *Dream of the Red Chamber* exhibits properties of a scale-free network (the frequency of character degrees follows a power law) and a small-world network (short average geodesic path length and high clustering).

Others calculate network metrics and model the networks to compare the social networks from narratives to other narrative social networks [35, 120, 125], or to real-life social networks [17, 75, 76].

Researchers also investigate the characters in narrative social networks [11, 21, 24, 36, 94, 120, 126], and underlying class structures of the characters [30, 56, 74, 127]. In particular, Bost *et al.* [30] demonstrate methods to extract the multiple complex storylines from modern television series using social networks, and show that characters within storylines cluster for the duration of the storyline.

Other narrative social network literature involves analysis of the relationships between characters [21, 44, 49, 83, 89]. Some of these analyses involve measures of sentiment, such as Deleris *et al.*'s analysis of *Friends* [44], which focussed on visualising the characters' relationships. In their social networks of the *Friends* characters, edges were labelled with the average sentiment of the words characters spoke to one-another, *i.e.*, whether they spoke in anger, disgust, joy, fear or sadness.

The online blog analyses of narratives mirror the formal literature. Many blogs involving narrative analysis focus on visualising the narrative, and so social networks are a convenient method [27, 31, 53, 55, 59, 66, 73, 81, 84, 105, 118, 128].

### 2.3.1 Temporally-evolving networks

Several studies also discuss the necessity of viewing a social network of a narrative temporally [11, 21, 30, 94, 127]. Temporal social networks are more informative than static networks as they incorporate a time component. In narrative analysis (and many other applications), a static network is not appropriate because much of the interest is in how the story unfolds and transforms over time. Most attempts at analysing temporal social networks involve time windows in which we view the static network over each window, and observe the changes in these static networks as the time window moves [13, 30, 69, 83].

When the time window is short, there may not be enough structure for a decent analysis. However, if the time window is too large, we lose some finer details about the timing of interactions. Some attempts have been made to determine the optimal length of time windows [116], but for narrative analysis, there are some obvious choices. For example, in novels, the time windows could be paragraphs, pages or chapters. In television series, the time windows could be scenes, episodes or seasons, which is what we consider here.

### 2.3.2 Social network extraction

A vital part of narrative social network analysis is extracting the social network from the narrative. A variety of methods have been used to do this in the literature.

A *co-occurrence network* is a popular way of automating the extraction process, where characters "interact" if they appear together in some time window, *e.g.*, in the same scene [47, 89, 127], book [17, 56], chapter [36, 83, 94] or within a number of words [24] as each other.

Alternatively, we could define an interaction as a character mentioning or conversing with another character. Several methods for automating this process are also under development [12, 34, 38, 44, 46, 62, 65, 101, 125].

Other ways to extract the social network are to do so manually [21, 75, 76], or to use alternative types of "interactions". For example, Gorinski and Lapata [58] created a bipartite network for their film analysis, where nodes are split into two groups; characters and scenes, and edges only go between groups. Note that this similar to our approach, however we retain the bipartite information.

We describe extraction methods in more detail in Chapters 3 and 4. In this study, we extract a co-occurrence social network, and use a previously manually extracted network, for *Friends* and use it as a case study for narrative social network analysis.

## 2.4 *Friends*

*Friends* is an American situation comedy (sitcom) created by David Crane and Marta Kauffman, with ten seasons aired from 1994 to 2004. We choose *Friends* as a case study for our model because the series is well-known, long-running and a popular subject and narrative case study amongst researchers [21, 38, 44, 79, 89].

*Friends* is known for having six core characters:

- Chandler Bing, played by Matthew Perry,
- Joey Tribbiani, played by Matt LeBlanc,
- Monica Geller (later Bing), played by Courteney Cox,
- Phoebe Buffay, played by Lisa Kudrow,
- Rachel Green, played by Jennifer Aniston, and
- Ross Geller, played by David Schwimmer.

Figure 2.1 shows these six characters underneath the *Friends* logo. Here, Rachel appears to be the central character, however our analysis in Chapter 5 shows that this is not the case.

The series takes place in Manhattan, New York City, where the eponymous "friends" live. The friends are commonly seen in Monica and Rachel's apartment, across the hall in Joey and Chandler's apartment, or in their favourite coffee shop, "Central Perk". Throughout the ten seasons, we watch the friends as they enter new relationships, break-up, start new jobs, quit old jobs, and many other events along the way.

Whilst *Friends* has no dominant storylines, there are several recurrent stories that are built on over the series. One of the most notable storylines is the intermittent relationship between Ross and Rachel. The producers intended for this to be the central romance [41], but several other significant relationships form over the ten seasons. Using social network analysis, we can verify whether Rachel and Ross remained the central romance. This will be one of our findings in Chapter 5.

Between 1994 and 2004, when it was being aired, *Friends* was very popular. Every season ranked within the top ten of Nielsen ratings for American primetime television series [2], and ranked first in its eighth season. The series finale in 2004 was particularly popular, with approximately 52.2 million American viewers [67]. In fact, it had the 5th highest viewer rating for any series finale as of 2015.

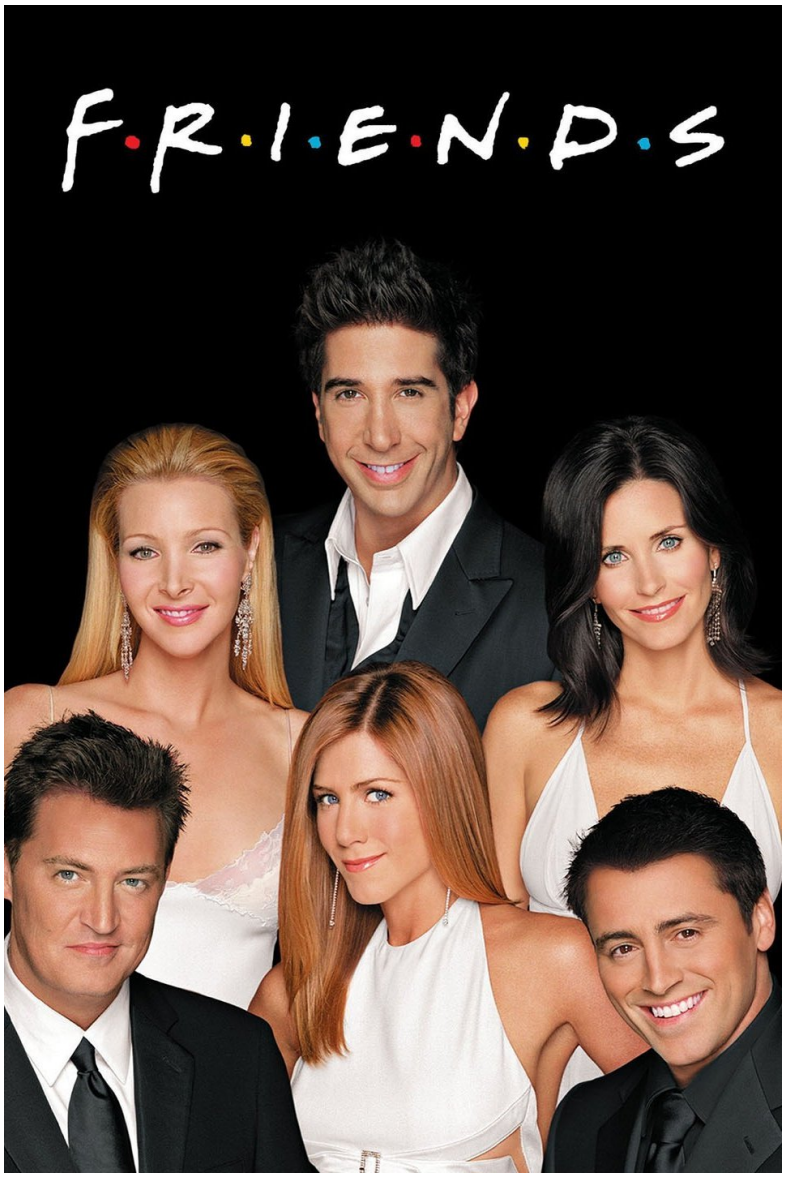



Figure 2.1: Image of the *Friends* logo, and the six core characters: Phoebe (top left), Ross (top middle), Monica (top right), Chandler (bottom left), Rachel (bottom middle) and Joey (bottom right) [5].

More recently, *Friends* has arrived on television streaming services such as Netflix [42] and Stan [4] (in Australia). Through these services, *Friends* is reaching a whole new generation, and hence it remains very popular [117].

*Friends* has been the subject of several narrative network case studies [38, 44, 89], and some in-depth analysis [21, 79]. Marshall [79] analysed

representations of friendship, gender, race and class in the series in his thesis, whereas Quaglio [95] analysed the language used in the series and compared it to natural conversation language.

Bazzan [21] manually created a temporal social network for the entire series of *Friends*, and analysed it using global, character and relationship metrics, and looked at how different clustering methods perform on the networks. We perform similar metric analyses with a different dataset representing the *Friends* characters and their interactions in Chapter 5, and extend the analysis to modelling the network and network features in Chapter 6.

In the next chapter, we describe how we extracted the *Friends* network from available resources, and then we will perform analysis and modelling of the networks and the network's metrics.

# Chapter 3

# The One With the Data Collection

"Now, I need you to be careful and efficient. And remember: if I'm harsh with you, it's only because you're doing it wrong."

*Courtney Cox* as *Monica Bing*
*Season 10, Episode 16*

## 3.1 Introduction

Our first step towards analysing a social network is data collection. The data collection task involves converting a narrative into a social network. We want the social network to represent the characters of the narrative and their relationships.

Mathematically, we define a social network $G = (V, E)$ by its nodes $V$ and edges $E$. The nodes represent *characters* in the narrative, and the edges represent *interactions* between characters. Here, we extract weighted social networks, where edges have non-negative integers associated with them, representing the *number of interactions* between characters within the timeframe of the network.

We have some options as to how we define *characters* and *interactions*. While many characters are obviously characters, some "characters" may not be considered characters by everyone. For example, pets may interact with obvious characters, but do they count as characters themselves? Or if obvious characters talk to their answering machine, does that mean the answering

machine should be considered a character? Therefore we need to carefully define the characters in the social network.

Similarly, we could define an interaction as a pair of characters having a conversation, or one character speaking to another character. In the former case, the interactions are mutual, so the social network is undirected, whereas in the latter case, the edge would go from the speaking character to the character being spoken to. Other possibilities for interaction definitions include; characters are friends, characters touch each other, characters are in a romantic relationship, or characters appear in the same place as each other. Hence we must clearly define interactions.

As a case study for narrative social network analysis, we extract social networks from *Friends*. The whole series consists of 236 episodes, making up 10 seasons.

Our goal is to analyse the social networks, characters and relationships in *Friends*, and how these change throughout the series. Therefore our social network needs to represent all characters of interest, and the strength of their relationships as they change over time. For the temporal component of the analysis, we partition the series into individual episodes and create social networks for each episode. For season and series time windows, we merge the relevant episode networks by adding the weights of the edges.

We discuss the elements of different techniques for social network extraction, and the effect they have on narrative social network analysis, in Chapter 4, but here we describe how we used two different methods, manual extraction and co-occurrence extraction, to obtain two datasets of social networks for *Friends*.

## 3.2 Manual network

### 3.2.1 Motivation

Manual extraction involves the data collector familiarising themselves with the narrative and recording each character and interaction, however characters and interactions are defined. Additionally, the temporal component involves recording times of interactions, which could be grouped into scenes or episodes if the narrative is a movie or television show, or in order of occurrence. Manually extracting the social network produces high-quality data, as it is not necessary to make coarse assumptions that may approximate and introduce errors.

### 3.2.2 Definitions

In the **manual** dataset of *Friends*, we define an interaction as two characters talking (even if one talks and the other listens), touching or making eye contact as done by Bazzan [21]. Bazzan defines a character as any human or animal that talks to, touches or looks at any other human or animal. The characters are checked against the credits and character list on the Internet Movie Database (IMDb) [1]. Bazzan defines the edge weight as the number of interactions between two characters in a given time frame. The smallest time frame is an episode, and the largest is the whole series.

### 3.2.3 Method

Prof. Ana Bazzan [21] manually collected the data by watching each of the 236 episodes, and it is available at `github.com/anabazzan/friends`. The networks are stored as edge lists in a text file, with comments signalling the season and episode number, as well as any extra information, such as if the episode is a Thanksgiving or flashback episode. Table 3.1 shows some features of the **manual** dataset.

| | Manual | Co-occurrence |
|---|---|---|
| Episodes | 236 | 227 |
| Seasons | 10 | 10 |
| Interactions | 16569 | 18574 |
| Interacting pairs | 1609 | 2695 |
| Characters | 746 | 669 |

Table 3.1: Table of features of **manual** dataset and **co-occurrence** dataset for the social network of *Friends*.

### 3.2.4 Issues

Inspection shows that the **manual** data represents the characters of interest and their relationships well, and we can be confident that the data is of high quality. However, there are some disadvantages of obtaining the social network in this way. The first issue is that the process is time consuming. Each episode of *Friends* is approximately 22–23 minutes long, so it takes approximately 88.5 hours to watch the entire series, let alone the time it takes to note the interactions, check characters against the credits and correct any mistakes. This makes manual extraction impractical for analysing a large corpus of narrative social networks. Other possible issues with this method

of network extraction are human errors, such as typographical errors, and the decisions made around "grey areas". Grey areas could occur in the data collection when characters almost interact, but not quite, or if it is unclear who a character is talking to. In these cases, a decision is made as to whether to include the interaction or not. Although a single person collecting all the data is more likely to be consistent with such decisions, it is difficult to be perfectly consistent, especially over the 88.5 hours of the entire series. Also, these cases mean that if someone else extracted a network in the same way, using the same definitions, the resulting network could be different.

Additionally, the length and type of interactions were not recorded. Although some characters interact hold long conversations and some only make eye contact for a fraction of a second, each interaction is weighted equally. We could also have friendly interactions, where characters get along, or hostile interactions where characters are fighting. Manual extraction methods could include length and type of interaction, however this would increase the complexity of the task, which would take even longer and could introduce more errors.

## 3.3 Co-occurrence network

### 3.3.1 Motivation

An alternative method to manual extraction is to automate the process. Different automated extraction processes make use of different features of the characters or network. Some such processes are discussed in Chapter 4. One way to automate the extraction process is to extract a **co-occurrence** network, where an interaction is assumed to occur if characters both appear within a scene, number of words or other constraint. Beveridge and Shan [24] extracted a **co-occurrence** social network from the third book in the *Game of Thrones* series: *A Storm of Swords* by incrementing the edge weight between two characters whenever their names appeared within 15 words of each other. Weng *et al.* [127] used character co-occurrences in scenes to create social networks of various movies. Min and Park [83] extracted a temporal social network of Victor Hugo's *Les Misérables* based on whether characters are mentioned in a chapter together. **Co-occurrence** networks are simple to extract as scenes and chapters (and character names within them) are easy for a machine to identify from a television or movie script or the text from a novel.

### 3.3.2 Definitions

For our **co-occurrence** network of *Friends*, we define an interaction as two characters speaking in the same scene. Notice we only count characters that speak in the scene, as these are the characters that are easily identified from a script. For this reason, we define characters as anyone or anything with a speaking role as expressed in the script. In contrast to the **manual** networks, here we include characters that talk only to themselves, and also include objects such as "answering machine" and "oven" as characters if they make noises written in the script, such as beeping. As in the **manual** dataset, the smallest timeframe we have in the **co-occurrence** dataset is episodes. Episode networks are combined to form seasons and the overall series networks.

### 3.3.3 Method

To extract the **co-occurrence** network of *Friends*, we clean the scripts to identify the scenes and characters speaking in the scenes. We use the cleaned scripts to create co-occurrence edge lists for each episode, and iteratively clean any remaining errors in the data. Python [123] code for parsing and cleaning the scripts and networks is in the Github repository for this thesis: `https://github.com/AdelaideUniversityMathSciences/MediaStudies`.

The scripts for each episode are on the fan webpage `www.livesinabox.com/friends/scripts.shtml` [6]. We parse the episode script Hypertext Markup Language (HTML) file to classify each line into;

- Scene (signalling the start of a new scene),
- Title (of the episode),
- Action (of a character),
- Dialogue (beginning with the speaking character),
- Information (signalling the end or the start of the end credits), or
- Not important (information about HTML link, copyright, *etc.*).

Lines that do not fit into any of these classes are part of the previous line.

Using the classified lines, for each episode we separate the episode into scenes and each character speaking in the scenes. In the **co-occurrence** dataset, the set of characters speaking in a scene creates a fully connected network (or a clique), so for each pair of characters in a scene, we add one to their edge weight to signify an interaction. We end up with a weighted

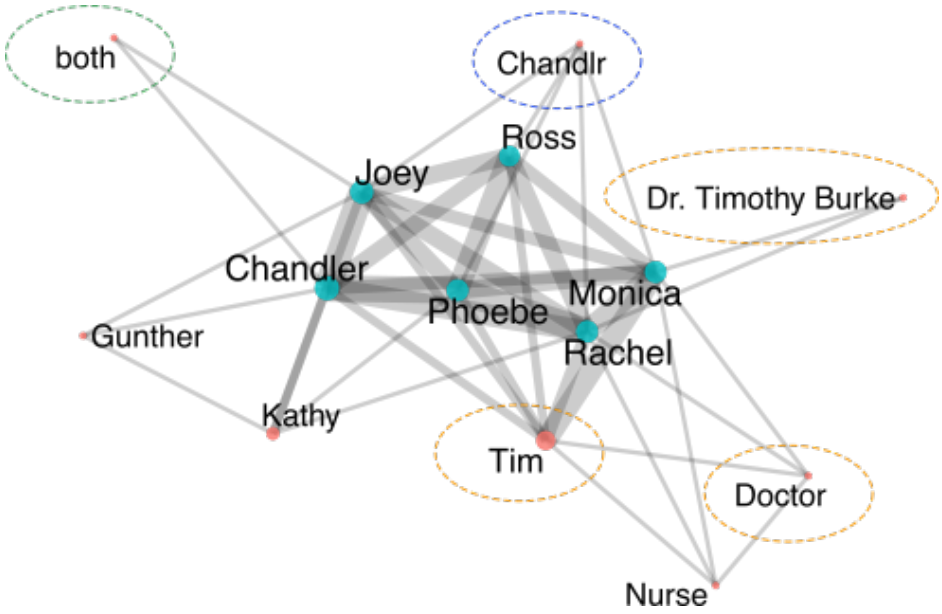



Figure 3.1: Example of uncleaned **co-occurrence** network for Season 4, Episode 8: *The One with Chandler in a Box*. The node representing more than one character is circled in green, the typographical error is circled in blue and the aliases are circled in orange.

edge list for each episode, where the weights of edges between characters correspond to how many scenes they co-occurred in.

Once the edge lists have been extracted, we analyse each network for errors. Figure 3.1 highlights three common errors we encounter;

1. Nodes representing more than one character (circled in green), *e.g.* "both",

2. Typographical errors and misspellings (circled in blue), *e.g.* "Chandlr", and

3. Aliases (circled in orange), *e.g.* "Tim" = "Doctor".

Figure 3.1 shows a node labelled "both". This is not the name of a character, but represents two characters who speak at once. In this context, we could take a guess that the two characters represented by "both" are Chandler and Joey, as the apparent character interacts with these two, however it may not be as obvious simply from the network in every case. We choose to ignore any "characters" that represent more than one character on the basis that if a character speaks at the same time as other characters, it is likely that they have already spoken in the scene, and hence will already be

counted. If the character does not speak alone in the scene, we assume their presence in the scene is not important enough to warrant interactions with every other character.

The node labelled "Chandlr" in Figure 3.1 is a misspelling of the character Chandler. The scripts we use were manually transcribed by fans of *Friends*, so human typographical errors arise. While we can tell that the spelling of "Chandlr" is very close to Chandler, and "Chandlr" interacts with similar characters to whom we would expect Chandler to interact with, we cannot consistently identify typographical errors like this. For example, in Season 2, Episode 10: *The One with Russ*, Rachel dates a character named Russ. The producers intentionally named this character similarly to Ross: Ross and Russ are comically similar in appearance and persona. Therefore "Russ" is not a misspelling of Ross, even though it may appear that way without outside knowledge. Hence, we manually identify these errors by plotting and looking through each episode network.

Finally, three characters are circled in orange in Figure 3.1, highlighting the issue of aliases. Aliases occur when one character is referred to by more than one name in the script, for example Jack could be referred to as "Jack", "Jack Geller", "Mr. Geller" or "Ross's dad". Once identified, we can change all of these instances to simply "Jack". Here, Dr. Timothy Burke is referred to as "Dr. Timothy Burke", "Tim" and "Doctor". We can change all instances of "Dr. Timothy Burke" to "Tim", but if we also change all instances "Doctor" to "Tim", we could be changing the label of unnamed doctors in other episodes, even if they aren't actually Dr. Timothy Burke. We can either change these instances manually, but here we leave the errors in the data, assuming they won't significantly affect the analysis of the social networks as these cases are rare. Hence for this dataset, we only change non-generic aliases.

We use regular expressions to deal with the three types of errors. For the nodes representing more than one characters, we create a list of all instances of these nodes. After identifying the set of "characters" in a scene, we remove any names that appear in this list. Similarly for the typographical and alias errors, we create a regular expression dictionary. The dictionary includes the name to change, and the preferred name. We refer to this list when we find the set of characters in the scene and change the names. Although manual work is sometimes involved in identifying these errors, the process of fixing the errors (using regular expressions) is automated. The regular expression dictionary code is in Appendix B.1.

Table 3.1 shows some features of the cleaned **co-occurrence** dataset.

### 3.3.4 Issues

Even after removing nodes that refer to more than one character and changing the names where there have been typographical or inconsistency errors, there are some issues with the **co-occurrence** dataset. Firstly, we are bound to have missed some of the errors that need changing. Over a total of 227 episodes, checking each one closely for these errors is time consuming, which defeats the purpose of the automated network. Also, as mentioned, some aliases are impossible to change automatically, so these errors could still affect the quality of the dataset.

Secondly, the labelling of scenes is not consistent throughout the scripts. Many scene changes are indicated by a line beginning with

"`[Scene: `",

but some scene changes start with "`[Cut `", "`[Time lapse`", "`[Flashback`", *etc.*, and others simply say the new location and the characters that start in the scene so our scene identification must account for these. With all these inconsistencies, the task of automatically identifying scene breaks becomes complex, which creates more room for error. We also include commercial breaks written into the script as scene changes, even though after the break the location and time setting may not have changed.

Another issue with the **co-occurrence** dataset is that we only include characters that speak in a scene. While most of the important characters do speak at some point, there are some characters, such as the baby Emma and Joey and Chandler's pets Chick and Duck, who certainly interact with others in some sense, but never speak. We also assume everyone in a scene interacts with everyone else in a scene, however, a character could have left the scene by the time another character arrives, so they never actually meet.

Finally, as in the manual data collection, we do not record the length of scenes, so don't store information about the length of interactions in the **co-occurrence** dataset. This means we could have two characters having a long conversation, then a third comes along and says one word. Each of these interactions are weighted equally.

## 3.4 Overview of network datasets

Table 3.1 shows some features of the **manual** and **co-occurrence** datasets.

Notice that there are more episodes in the **manual** dataset than in the **co-occurrence** dataset. This is because the scripts used to create the **co-occurrence** networks treated some double episodes as single double-length

episodes, as they were originally aired. These episodes are sometimes split into two parts (*e.g. The One After the Superbowl: Part 1* and *The One After the Superbowl: Part 2* in Season 2) for reruns and DVD release, and are thus counted as two individual episodes in the **manual** dataset. For this reason, we cannot compare every episode in the two datasets unless we combine the double episodes in the **manual** dataset. Throughout the thesis we use the episode names and numberings from the **manual** dataset.

Also notice that there are more interactions and interacting pairs in the **co-occurrence** dataset than in the **manual** dataset, but there are fewer characters in the **co-occurrence** dataset than the **manual** dataset. We discuss these differences in Chapter 5.

## 3.5 Other data

### 3.5.1 From the script

Through the automated **co-occurrence** network extraction from the episode scripts [6], we also collected

- the number of lines in each episode, where a line is a continuous statement from a single character (or group of characters), and
- the number of words spoken in each episode.

These give indication about the length of character interactions (through the average words per line for an episode). There are 616665 words and 61026 lines in the whole series. Table 3.2 shows some summary statistics for this data.

| | Words | Lines | Words/line |
|---|---|---|---|
| Minimum | 1896.00 | 176.00 | 8.09 |
| Maximum | 4032.00 | 402.00 | 13.54 |
| Mean | 2612.99 | 258.58 | 10.16 |

Table 3.2: Table of the minimum, maximum and mean of the number of words, lines and words per line in each episode of *Friends*.

### 3.5.2 From IMDb

We use the code in GitHub to scrape IMDb [1] for the

- title,
- rating (out of 10), and
- number of people who gave ratings

for each episode. Viewers can rate each episode by logging into their IMDb account and choosing a number of stars between 1 and 10. The ratings we collect are the average of viewer ratings for each episode. The rating is a measure of how successful each episode was, so we model the success of the series with network metrics in Chapter 6. Table 3.3 shows some summary statistics for the ratings.

| | Rating | Raters |
|---|---|---|
| Minimum | 7.30 | 1864.00 |
| Maximum | 9.70 | 7740.00 |
| Mean | 8.52 | 2442.84 |

Table 3.3: Table of the minimum, maximum and mean of the average rating of, and number of people who rated (raters) each episode of *Friends* on IMDb.

## 3.6 Conclusion

We extracted two datasets that represent the social networks of *Friends* episodes; the **manual** and the **co-occurrence** dataset. While the **manual** networks show character relationships more precisely, the **co-occurrence** networks were extracted automatically. Other automated methods for extracting a social network from a narrative are available, but it is of interest to compare how different extraction techniques affect the network, and hence the narrative analysis. In the next chapter, we model and compare manual, co-occurrence and another automated extraction techniques and discuss how they affect the analysis of the narrative.

# Chapter 4

# The One With the Comparison of Network Extraction Techniques

"Pivot! Pivot! Pivot! Pivot. Pivot. Pivot."

*David Schwimmer* as *Ross Geller*
*Season 5, Episode 16*

## 4.1 Background

In Chapter 3 we discussed the difficulty of extracting a social network from a narrative. Manually extracting a social network from any narrative can be time consuming, so automatic extraction methods of varying complexity have been developed. However, the effect of different extraction methods on the analysis is unknown. In this chapter, we model and compare three extraction methods for social networks in narratives: manual extraction, automated extraction by co-occurrence, and automated extraction using machine learning.

One of the most problematic aspects of narrative social network analysis is constructing the network from an unstructured text source such as a script or novel. Extracting an interaction network from novels is challenging because the text does not always state who is speaking. Most attempts to match quoted speech in novels to the character speaking involve Natural Language Processing (NLP) and/or machine learning techniques [12, 34, 46, 62, 65, 125]. A disadvantage of these techniques is that there is either significant

manual work in identifying aliases of characters, or that the accuracy of character identification ranges from $< 50\%$ to $\approx 90\%$ [62]. The more manual work put in at the NLP stage, the more accurate the identification tends to be. Alternatively, researchers can manually identify the speakers in novels [11], but this takes substantially longer and is not practical for analysing large corpora.

Extracting social networks from film or television scripts is almost as difficult. The most accurate, but time-consuming, approach is to manually record interactions between characters [21]. A more scalable approach is to automatically create a social network from the script of the film or television show. Scripts necessarily label speakers, but not who each character is speaking to. There are examples of using NLP and machine learning techniques [37, 38, 44], but again there is a trade-off with the accuracy of identifications.

An alternative automatic method is to extract a co-occurrence network [47, 66, 89, 127], which infers interactions between characters from the number of times they appear in a scene together. We can create co-occurrence networks for novels as well, for example by counting the number of times characters are mentioned within a number of words of each other [24]. Using a co-occurrence network presumes that relationship strength can be measured by the number of times characters share a scene, as opposed to the number of times characters directly interact. While researchers have investigated the effect of different types of interactions in real life social networks [29, 71, 77, 80, 85, 86, 107, 115, 119], to the best of our knowledge, there is no research into the effect of network extraction methods in narrative analysis.

In this chapter we compare three social network extraction techniques in the context of television scripts for *Friends*:

- manual extraction (as in Bazzan [21]),
- extraction using NLP (as in Deleris *et al.* [44]), and
- co-occurence extraction using scripts.

To compare these techniques we create a model to simulate interactions in a narrative. Using the simulated interactions, we create and compare observation networks based on the three extraction techniques. This *in silico* model allows us to compare techniques with complete knowledge of the ground truth. Modelling the narrative also allows us to control and measure parameters such as the error rate for the NLP method or the number of scenes for the co-occurrence method. There are inconsistencies in the naming of characters across our datasets, so the model allows us to compare attributes of individual characters and relationships across the different

extraction techniques. We can also simulate more data, which makes our analyses more powerful. Finally, the model allows our methods to be applied to a range of narratives, not just the case study we give here.

We use standard network metrics (see Chapter 2) to compare the three different network extraction techniques, applied to the characters in the television series *Friends*.

In this chapter we find:

- Co-occurrence networks have higher edge densities than the manually extracted networks, but the densities are highly correlated between techniques (the Pearson's correlation coefficient is 0.96).
- Centrality measures (degree, betweenness, eigenvector and closeness) are highly correlated in the manually extracted networks and co-occurrence and NLP networks, but clustering is not reliable in the automated networks.
- Edge weights in the automated networks correlate moderately with the edge weights in the manually extracted networks (the median Spearman's correlation coefficient is 0.77 for the co-occurrence networks and 0.80 for the NLP networks).

We conclude that automatically extracted networks – co-occurrence and NLP networks – give reliable analyses for most global, character, and relationship metrics, so we recommend extracting narrative social networks in one of these ways for time efficiency. If clustering is of high importance in an analysis, however, manually extracted networks are required.

## 4.2 Data

Although our findings are partially based on *in silico* experiments, we use real data to inform our models and to provide final verification. Details of the real datasets are in Chapter 3.

We examine three methods to extract social networks from *Friends*. The social network describing character relationships are defined by nodes that represent characters in a chosen time frame (usually an episode or season), and edges connecting characters who interact. The precise definition of an interaction varies throughout the literature, but the assumption that characters who interact more have stronger relationships remains constant. Our goal is to model these relationships. Note that the strength of a relationship does not imply characters are good friends (despite the name of the series), as characters can have strong hostile interactions [75, 83].

The first dataset consists of manually extracted data by Bazzan [21], described in Chapter 3. While there may be human interpretation errors in this dataset, this is the most reliable method of extracting the social network. Therefore, the manual extraction method provides a "gold standard" for the social networks of the characters. We call the networks from this dataset the **manual** networks. The edge weights in the **manual** networks correspond to the number of interactions between two characters in a given timeframe. Table 4.1 shows the number of episodes, interactions, scenes and characters in each season.

| Season | Eps | Chars | Ints | Scenes | Ints/Ep | Scenes/Ep | Ints/Scene |
|---|---|---|---|---|---|---|---|
| 1 | 24 | 126 | 2492 | 364 | 103.83 | 15.17 | 6.85 |
| 2 | 24 | 107 | 1815 | 314 | 75.62 | 13.08 | 5.78 |
| 3 | 25 | 98 | 1770 | 422 | 70.80 | 16.88 | 4.19 |
| 4 | 24 | 96 | 1598 | 438 | 66.58 | 18.25 | 3.65 |
| 5 | 24 | 92 | 1786 | 378 | 74.42 | 15.75 | 4.72 |
| 6 | 25 | 99 | 1491 | 387 | 59.64 | 15.48 | 3.85 |
| 7 | 24 | 81 | 1475 | 402 | 61.46 | 16.75 | 3.67 |
| 8 | 24 | 110 | 1220 | 356 | 50.83 | 14.83 | 3.43 |
| 9 | 24 | 101 | 1454 | 345 | 60.58 | 14.38 | 4.21 |
| 10 | 18 | 88 | 1468 | 238 | 81.56 | 13.22 | 6.17 |

Table 4.1: Summary of data from **manual** dataset [21]. For each season we have the number of episodes (Eps), the number of characters (Chars), the total number of interactions (Ints), and the number of scenes (Scenes). We also calculate the number of interactions per episode (Ints/Ep), number of scenes per episode (Scenes/Ep) and average number of interactions per scene for each season (Ints/Scene).

Table 4.1 shows there are 24 episodes in most seasons, but 25 episodes in Season 3 and Season 6 and only 18 episodes in Season 10. Season 1 has notably more interactions than any other season, possibly due to the need to establish characters and relationships at the beginning of the series. We will discuss our findings on trends in network properties over all 10 seasons in Chapters 5 and 6.

The second dataset contains **co-occurrence** networks, extracted using scripts available from a fan website [6], as in Chapter 3.

An NLP network dataset for *Friends* was not available, but Deleris *et al.* [44] provide information about how they extracted the social network, making use of Chen and Choi's data [38]. Chen and Choi use NLP techniques

to identify which character is mentioned when another character says "you", "he", "they", *etc.* They estimate their model correctly identifies a character 69.21% of the time. Deleris *et al.* use "character mention" information to build a directed social network where the interactions are one of four kinds of signals:

- Direct Speech (*e.g.* A talks to B).
- Direct Reference (*e.g.* A says 'I like you' to B).
- Indirect Reference (*e.g.* A says 'I like B').
- Third-Party Reference (*e.g.* C says 'A likes B').

Each of these is an example of a directed interaction from A to B. However, for the purpose of modelling networks consistently between all three approaches, we assume all interactions are reciprocated. We call the undirected networks extracted using this approach the **NLP** networks.

## 4.3 Method

### 4.3.1 Overview

We compare the network extraction methods by simulating narrative social networks, "extracting" observed networks using the three extraction methods and comparing these observed networks. We simulate social networks using a data-driven model. Simulation allows us to generalise the problem to any narrative that has a similar underlying social network and to generate large datasets for statistical analyses. The simulation and extraction process is outlined in Figure 4.1 and the following sections.

We estimate parameters for our model using the **manual** networks data, then use the model to simulate $n_u$ underlying season networks. For each season network, we use a random walk process to simulate $n_s$ scenes. We then combine the scenes to form a simulated episode. From each simulated episode, we extract three observation networks resembling the **manual** networks, **co-occurrence** networks and **NLP** networks. We compare these simulated networks using the network metrics outlined in Section 4.3.6.

### 4.3.2 Simulate season from data

The first step described by Figure 4.1 is simulating underlying season networks from the observed data. To simulate networks we need to model the

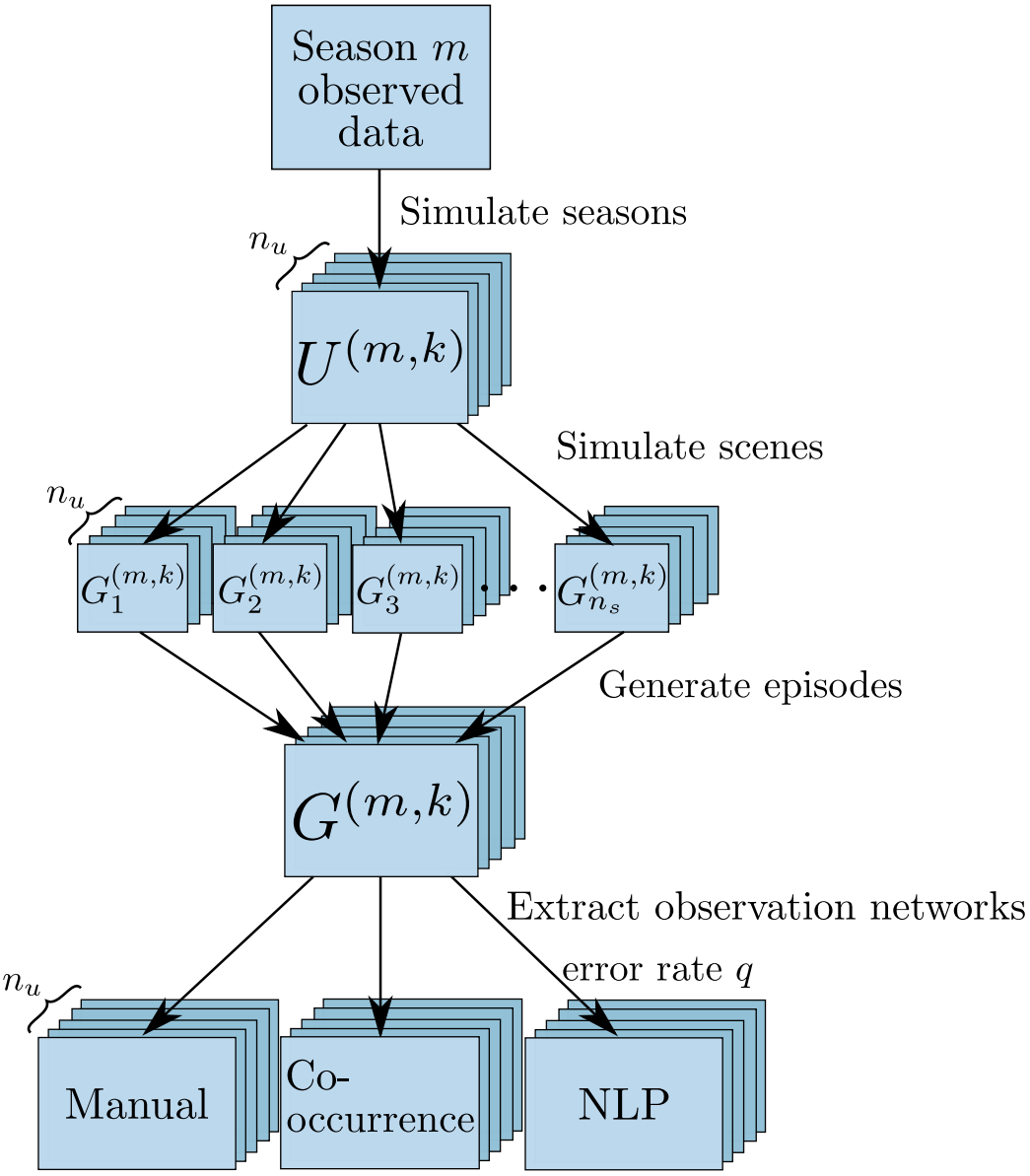



Figure 4.1: Flow chart describing the simulation process. We model each season $m$ in the observed data to simulate underlying season networks $U^{(m,k)}$, where $k = 1, \ldots, n_u$ (Section 4.3.2). From each underlying season network we simulate scene networks $G_\ell^{(m,k)}$ for $\ell = 1, \ldots, n_s$ (Section 4.3.3). The scene networks combine to generate episode networks $G^{(m,k)}$ (Section 4.3.4). From each simulated episode network we extract three observation networks; a **manual** network, a **co-occurrence** network networks and an **NLP** network (Section 4.3.5).

seasons in the **manual** networks dataset. We want, in addition to edges, to simulate edge weights, non-negative integers representing the number of character interactions. We notice there are significant differences between the way the core characters of *Friends* (Monica, Rachel, Phoebe, Ross, Chandler and Joey) interact with each other (average of 81 interactions per pair per season) and with other characters (average of 0.71 interactions per pair per season), and the way other characters interact with each other (average of 0.0093 interactions per pair per season). We therefore propose a two-class Poisson model for each season of the **manual** networks. More details of the model selection are in Chapter 6.

Let $V^{(m)} = \left\{1, \ldots, N^{(m)}\right\}$ be the set of characters in Season $m$ and $w_{ij}^{(m)} \geq 0$ be the number of interactions between character $i$ and character $j$ in Season $m$ of the **manual** networks dataset. We partition $V^{(m)}$ such that

$$V^{(m)} = V_{\text{core}} \cup V_{\text{non-core}}^{(m)},$$

where $V_{\text{core}}$ contains the 6 core characters who are constant across all seasons, and $V_{\text{non-core}}^{(m)}$ contains the $\left(N^{(m)} - 6\right)$ non-core characters for Season $m$.

For the two-class Poisson model, assume each edge weight $w_{ij}^{(m)}$ in Season $m$ of the **manual** networks dataset is a random observation of

$$W_{ij}^{(m)} \sim \text{Poi}\left(\lambda_{C_i,C_j}^{(m)}\right),$$

where

$$C_i = \begin{cases} 1 & \text{if } i \in V_{\text{core}}, \\ 0 & \text{if } i \in V_{\text{non-core}}. \end{cases}$$

We estimate $\lambda_{C_i,C_j}^{(m)}$ using maximum likelihood estimation with the **manual** network edge weights:

$$\hat{\lambda}_{C_i,C_j}^{(m)} = \begin{cases} \dfrac{\sum_{i<j} w_{ij}^{(m)} C_i C_j}{\sum_{i<j} C_i C_j} & \text{if } C_i = C_j = 1 \text{ (within core class)}, \\ \dfrac{\sum_{i<j} w_{ij}^{(m)} (1-C_i)(1-C_j)}{\sum_{i<j} (1-C_i)(1-C_j)} & \text{if } C_i = C_j = 0 \text{ (within non-core class)}, \\ \dfrac{\sum_{i<j} w_{ij}^{(m)} (C_i + C_j - 2C_iC_j)}{\sum_{i<j} (C_i + C_j - 2C_iC_j)} & \text{otherwise (between classes)}. \end{cases}$$

For each season $m$ we simulate $n_u$ season networks

$$U^{(m,k)} = \left(V^{(m)}, E^{(m,k)}\right),$$

where $k = 1, \ldots, n_u$ and

$$E^{(m,k)} = \left\{ W_{ij}^{(m,k)} \mid i, j \in V^{(m)}, i < j \right\}.$$

Note that each simulation contains all characters $V^{(m)}$ from Season $m$, but the edge weights are randomised. We generate random edge weights between each pair of nodes from the distribution

$$W_{ij}^{(m,k)} \sim \text{Poi}\left(\hat{\lambda}_{C_i,C_j}^{(m)}\right).$$

This method allows for edges with zero-weights. We take zero-weights to mean there are no interactions between the characters, which is equivalent to having no edge between the characters.

### 4.3.3 Simulate scene from season

Given an underlying season network $U^{(m,k)}$, we wish to sample an episode network, every episode being a sequence of scenes. Following Fortuin *et al.* [49], we define a scene as a story part with a constant set of characters in a constant location. This approximation allows a consistent comparison between methods. Each scene also contains a set of interactions, so we can form a social network for every scene. Interactions within a scene are dependent. For example, if Joey talks to Monica, it is likely that Monica will then talk to Joey. We capture this in the model by proposing a random walk model for interactions in each scene.

The random walk model randomly picks a starting character in $V^{(m)}$, with probability proportional to the eigenvector centrality of the character in $U^{(m,k)}$ (see Chapter 2). We use eigenvector centralities because they are the steady state probabilities of the random walk system, however other probability distributions could also be used. The starting character randomly interacts with another character with probability proportional to the edge weight between the characters. That character randomly interacts with another character, selected in the same way. The random walk process continues until we reach $n_{\text{int},\ell}$ interactions. We choose $n_{\text{int},\ell}$ based on the average number of interactions per scene in the data from Table 4.1. The next scene starts with a new random starting character so that we have a fresh set of characters in each scene.

Each scene $\ell$ consists of a set of characters $C_\ell^{(m,k)}$ and interactions $L_\ell^{(m,k)}$. Let $n_s^{(m)}$ be the rounded average number of scenes per episode in Season $m$ from our datasets. We define the network of a scene $\ell$ sampled from $U^{(m,k)}$ as

$$G_\ell^{(m,k)} = \left(C_\ell^{(m,k)}, L_\ell^{(m,k)}\right),$$

where $\ell = 1, \ldots, n_s^{(m)}$. As shown in Figure 4.1, we independently simulate $n_s$ scenes with the random walk model from each simulated season network $U^{(m,k)}$, then combine the scenes to generate a random episode.

### 4.3.4 Generate episode from simulated scenes

We generate an episode by concatenating simulated scenes. An episode sampled from $U^{(m,k)}$ is

$$G^{(m,k)} = \left(G_1^{(m,k)}, G_2^{(m,k)}, \ldots, G_{n_s}^{(m,k)}\right).$$

The set of characters in $G^{(m,k)}$ are the union of the sets of characters in the scenes, *i.e.*, $\bigcup_{\ell=1}^{n_s} C_\ell^{(m,k)}$. The edge weight between character $i$ and $j$ in $G^{(m,k)}$ is the sum of the interactions between characters $i$ and $j$ in the scene networks, which is zero if at least one of $i$ or $j$ was not in the scene.

### 4.3.5 Extract observation networks from simulated episodes

As in Figure 4.1, we extract three observed networks from each simulated episode $G^{(m,k)}$; a **manual** network, a **co-occurrence** network and an **NLP** network. We compare these simulated networks using metrics outlined in Section 4.3.6.

The **manual** network is built from the actual data so it is assumed to be 100% correct. Therefore the simulated **manual** network extracted from $G^{(m,k)}$ is $G^{(m,k)}$.

The **co-occurrence** network is obtained by creating a clique for the characters in each scene. We add clique networks so that edge weights correspond to the number of scenes two characters are in together, as they would be in the automated process.

The **NLP** network counts interactions similarly to the **manual** network, however it simulates NLP by including errors in the identification of characters. We model these errors by assuming:

1. One character (the speaking character) has been identified correctly, but the character being spoken to may be misidentified with probability $q$.

2. An incorrectly identified character is equally likely to be any character in the episode except the speaking character or correct character.

Chen and Choi [38] obtained a "purity score" of 69.21% in their analysis of *Friends*, which they describe as the effective accuracy of character identification and hence we set $q = 0.3$. The impact of $q$ as it changes is a potential

direction for future work. We call the process of incorrect character identification "rewiring".

In practice, the definition of an interaction (and hence edge weight) differs in **NLP** networks compared to **manual** networks. In the **manual** networks an interaction occurs between two characters who see, talk to or touch each other, whereas an interaction in the **NLP** networks occurs when two characters talk to, mention or refer to each other. We do not have this information in the **manual** networks, so we assume that characters seeing and touching each other is equivalent to characters mentioning and referring to each other.

### 4.3.6 Comparing observation networks

To measure how social network extraction method affects narrative analysis we compare the simulated **manual** networks with the simulated **co-occurrence** and **NLP** networks, using three types of network metrics, which are defined in Chapter 2:

1. *Global metrics*: size, total edge weight, edge density and clustering coefficient.

2. *Node/character metrics*: degree, betweenness centrality, eigenvector centrality, closeness centrality and local clustering coefficient.

3. *Edge/relationship metrics*: edge weights.

These metrics are common in narrative social network analysis, providing useful insight into social structure and important characters and relationships. The aim is not to compare the observation networks directly, but to investigate the effect the different observation types have on the narrative analysis. Consequently, we are more interested in understanding how metrics correlate rather than systematic differences in their value.

## 4.4 Results

We simulate $n_u = 10000$ seasons using the two-class Poisson model on Season $m = 6$, as the number of scenes per episode in Season 6 is close to the mean number of scenes per episode over all ten seasons. Figure 4.2 shows the social network of interactions from Season 6. From each simulation, we sample interactions from one episode using random walks for each scene. Table 4.1 shows Season 6 of *Friends* has 25 episodes, 1491 interactions and 379 scenes. Therefore the average episode in Season 6 has approximately 60 interactions and 20 scenes. We set $n_s = 15$ scenes and $n_{\text{int},\ell} = 4$ for every scene $\ell$.

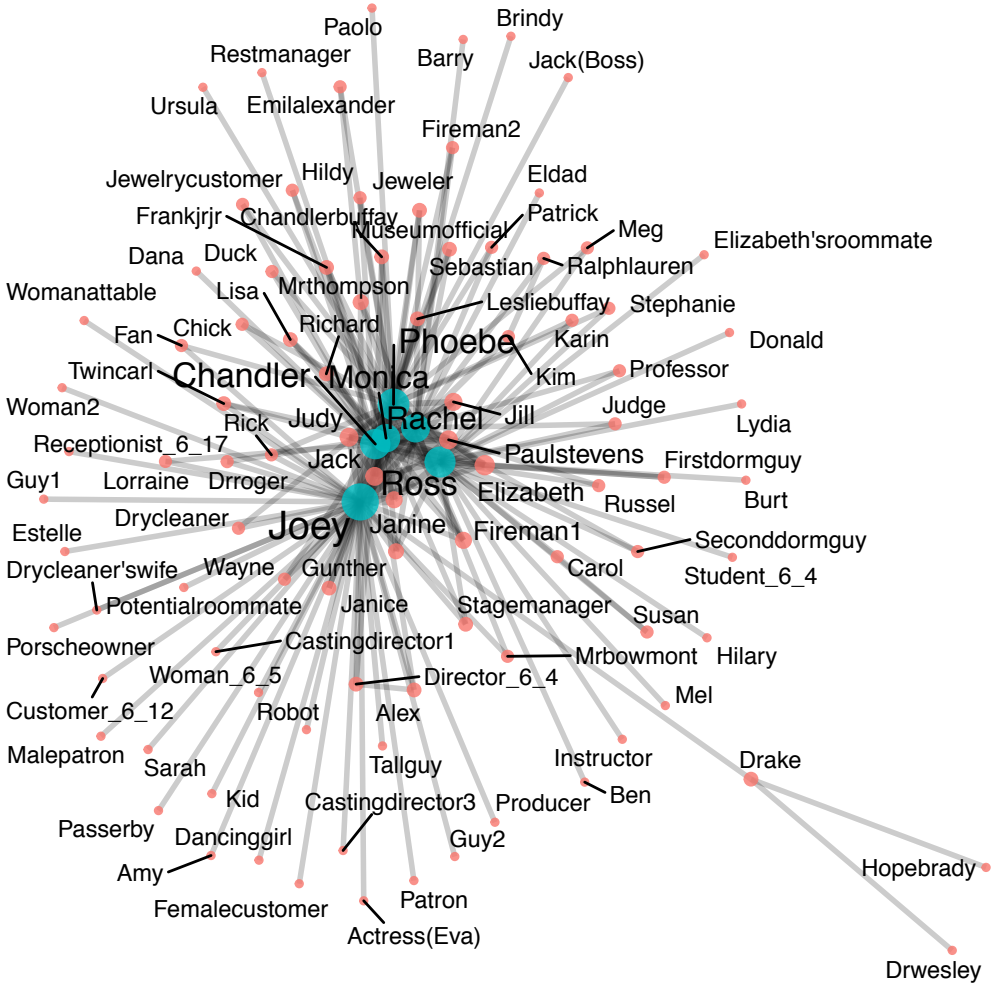



Figure 4.2: Network of Season 6 from the **manual** dataset. The core characters have blue nodes and other characters have red nodes. The width of the edges represent the edge weight and the size of the nodes represent the node degree.

From each sample episode network we "extract" the three observation networks using the methods described in Section 4.3.5 and compare using the metrics listed in Section 4.3.6. We find that there are differences in the value of the metrics across observation networks, but the errors are mostly systematic. While the exact values of metrics can vary across the different observation networks, the important features in the narrative analysis (*i.e.* the rankings and trends of metrics) would not be greatly affected. The global metrics of the simulated **co-occurrence** networks and **NLP** networks cor-

relate to those of the associated **manual** networks. The centrality metrics (degree, betweenness, eigenvector and closeness centrality) also have high correlation with the same character metrics across the simulated **manual** and random observation networks, but there is wide variance in the correlations of the local clustering coefficient of characters. The edge weights in the simulated **manual** networks also correlate reasonably highly with the edge weights in the simulated **co-occurrence** and **NLP** networks.

### 4.4.1 Global metric comparison

Global network metrics are useful for understanding characteristics of the social networks as a whole. These metrics applied to narrative social networks tell us important information about the number of characters in the narrative and how they interact on average. Popular global metrics for analysing narrative social networks are the size, total edge weight, edge density and clustering coefficient. Figure 4.3 show box plots of the normalised size, total edge weight, density and clustering for each network type. We normalise size and total edge by dividing by the maximum over the three network types.

#### Size

The size of the network in the case of the narrative networks is the number of characters in the narrative. The **manual** and **co-occurrence** networks have the same number of characters in each simulation by construction of the model. In the real data, however, it is possible for the **co-occurrence** network to miss characters if they interact but don't speak as they would not be mentioned in the script. But this is unlikely to cause problems in the analysis of the social network as characters with important roles are very likely to speak in a scene. We notice that discrepancies in the size of the networks in the two datasets were almost always due to differences in what defines a character. For example "answering machine" is counted as a character in the **co-occurrence** dataset, but not in the **manual** dataset, and Ross's monkey Marcel is counted as a character in the **manual** dataset, but not the **co-occurrence** dataset. As the "important" characters are most likely to be included in both networks, the analysis of the narrative would not be greatly affected by this.

The size of the **NLP** network is always equal to or less than the size of the other networks. This is because our model can only rewire to characters within the episode, but characters can be excluded if all the edges connected to that character are rewired away and no edges are rewired back to the character. This is more likely to happen to characters that are connected to

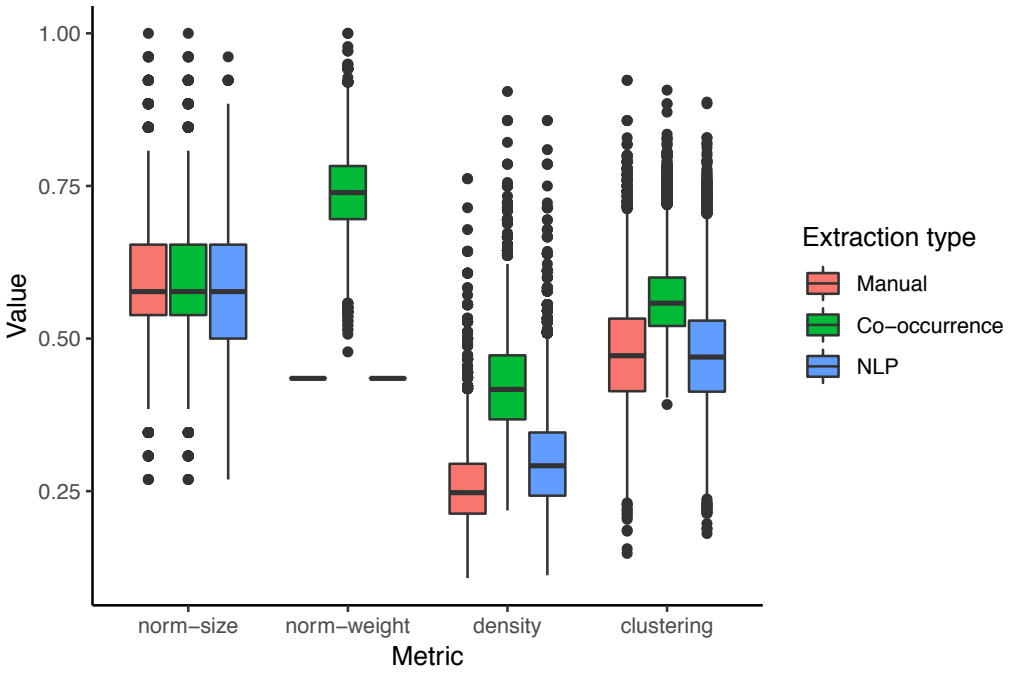



Figure 4.3: Box plots of the normalised size (norm-size), normalised total edge weight (norm-weight), edge density (density) and clustering coefficients (clustering) of the **manual**, **co-occurrence** and **NLP** networks from the 10000 simulations. The size and total edge weights are normalised by dividing by the maximum.

few edges in the first place, and so the effect of the rewiring on the analysis is minimal.

### Total Edge Weight

The total edge weight of our narrative social networks is the total number of interactions. This is an important metric for analysing how much characters are seen to interact with each other. Our model for the **NLP** network only rewires interactions, so these have the same amount of interactions as the **manual** network. In practice there might be discrepancies in total edge weights due to different definitions of interactions as discussed in Section 4.2 and Section 4.3.5.

The total edge weight for the **co-occurrence** networks, however, varies in different simulations. The simulated **co-occurrence** networks have between 10% and 100% more interactions than the **manual** and **NLP** networks on average. It makes sense that the **co-occurrence** network has more interactions

because we create a clique with all the characters from each scene. Therefore when analysing the **co-occurrence** network we should keep in mind the total edge weight will be larger than if we had the **manual** network.

Comparing the **manual** and **co-occurrence** network datasets we find that while the edge weights are generally larger for the **co-occurrence** network, the difference is not as significant. This could be due to the number of interactions, and hence characters, that occur in each scene in the model. In reality there are longer and shorter scenes with a varying number of interactions and characters, but for simplicity this is not reflected in our model.

### Edge Density

The edge density of a network indicates what proportion of character pairs directly interact. Figures 4.3 and 4.4 show that both the **co-occurrence** and **NLP** networks have a higher edge density than the **manual** networks, and the **co-occurrence** networks have higher edge densities than the **NLP** networks. Interestingly, it is rare for the simulated **NLP** network to have a lower edge density than the simulated **manual** network even though the edges are rewired with equal probability to any other character. This occurs because we only rewire one interaction, not the entire edge with its weight.

However, there is very high correlation between the **manual** network density and the other observation networks. Figure 4.4 shows a scatterplot of the two. The Pearson correlation coefficient between the simulated **manual** and **co-occurrence** network edge densities is 0.947, and between the simulated **manual** and **NLP** network edge densities is 0.950. This means that while there is some systematic bias, comparing social networks using relative edge density is not greatly affected. Importantly, the different extraction methods don't distort trends.

### Clustering Coefficient

Figure 4.3 shows that the simulated **co-occurrence** networks are more clustered than the simulated **manual** networks. This is not surprising as forming cliques for every scene creates clusters. We also notice that the clustering coefficients of the simulated **NLP** networks are distributed similarly to that of the simulated **manual** networks.

Figure 4.5 shows a scatterplot of the relationships between the clustering coefficients in the simulated **manual** networks and the simulated **co-occurrence** and **NLP** networks. The increase in clustering from the simulated **manual** to **co-occurrence** network is smaller for more highly clustered networks. This occurs because if the **manual** network is already highly clus-

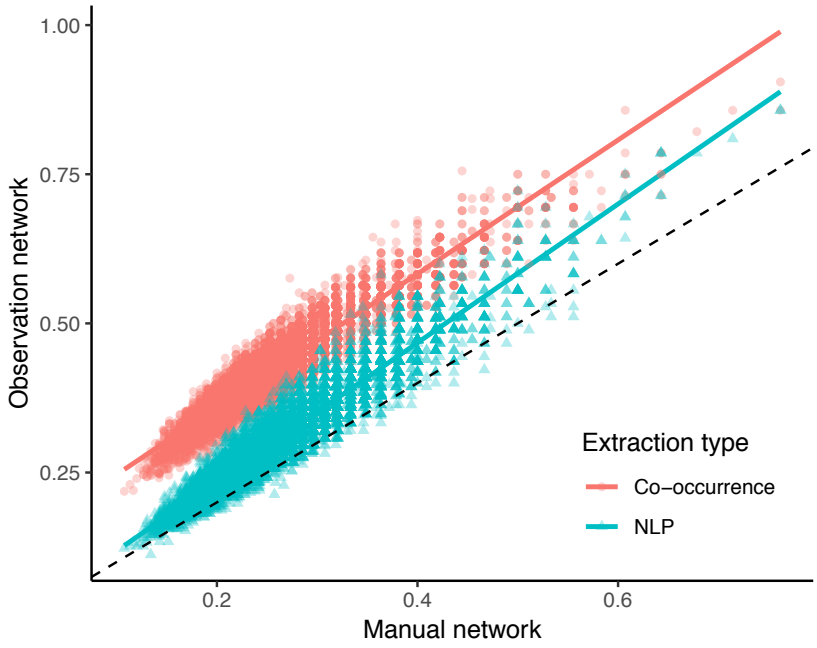



Figure 4.4: Edge density of the simulated **manual** network compared to the simulated **co-occurrence** (red) and **NLP** (blue) networks. The dashed line shows $y = x$. The $R^2$ value for the **co-occurrence** networks is 0.898, and for the **NLP** networks is 0.903.

tered, forming cliques in every scene will add fewer interactions between characters. Unclustered networks, however, will appear clustered using the co-occurrence method, so analysis of clustering is not reliable in **co-occurrence** networks. This is the largest non-systematic distortion we see across the different techniques.

The clustering coefficients of the simulated **NLP** networks are similar to those of the simulated **manual** networks (Pearson's correlation coefficient of 0.805), but there is some variation due to rewiring interactions. Therefore, when analysing clustering in the networks, **NLP** networks are reliable in general.

### 4.4.2 Character metrics

Character metrics are used in narrative social network analysis to investigate the role of each character. Generally narratives are made up of a variety of characters with different roles. For example, Agarwal *et al.* [11] determined that Alice was the main storyteller in Lewis Carroll's *Alice in Wonderland*,

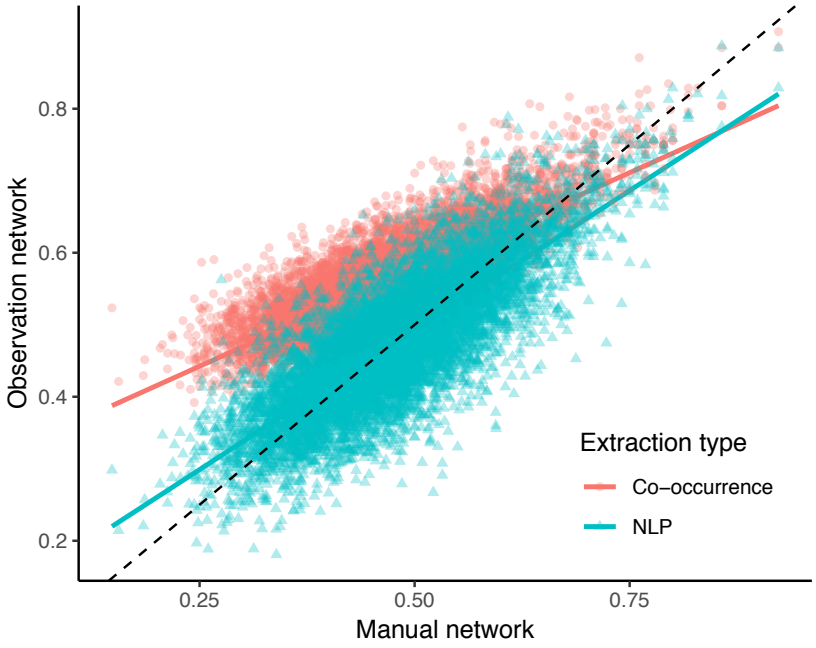



Figure 4.5: Clustering coefficient of the simulated **manual** network containing all interactions compared to the simulated **co-occurrence** (red) and simulated **NLP** (blue) networks. The dashed line shows $y = x$. The $R^2$ value for the **co-occurrence** networks is 0.659, and for the **NLP** networks is 0.648.

whereas Mouse's main role was to introduce other characters to Alice. Similarly Bazzan [21] showed that in *Friends*, while Joey connects many characters, Monica interacts the most with the five other main characters. We investigate character roles using character metrics in Chapter 6. Degree, betweenness centrality, eigenvector centrality, closeness centrality and local clustering coefficient are commonly used to assess the relative importance of the characters. We care more about comparisons between characters, *e.g.* who is the most central, so we examine the correlation between character metrics in the different networks, not the actual values. We use Spearman's correlation coefficient because we are interested in the rankings of importance of characters. Figure 4.6 shows box plots of these correlations for weighted degree, betweenness, eigenvector and closeness centrality, and local clustering coefficient.

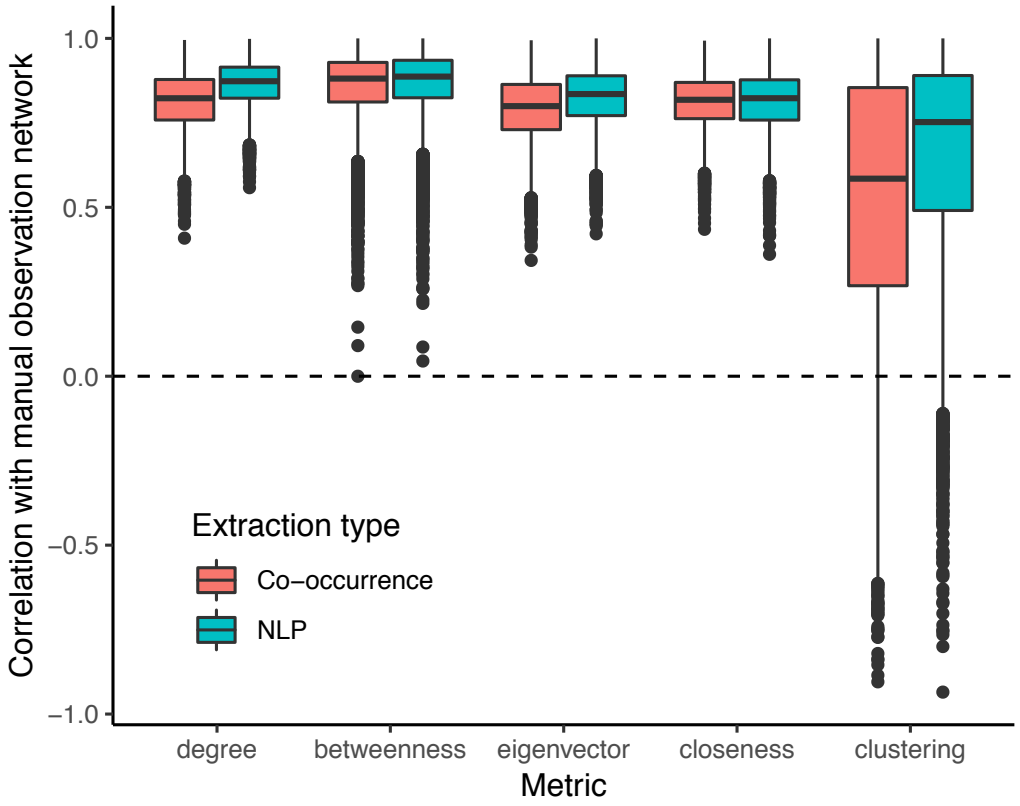



Figure 4.6: Box plots of the character metric correlations between the simulated **manual** network and simulated **co-occurrence** (red) and **NLP** (blue) networks from the 10000 simulations.

### Weighted Degree

In a narrative social network a character with a high weighted degree is involved with many interactions and hence is very social, so is likely to be a main character in the narrative. Figure 4.6 shows that there is a high correlation between the weighted degree of characters in all three observation networks, especially between the **NLP** and **manual** networks. This suggests that the observation type does not have a strong effect on the relative number of interactions each character makes, so while the actual weighted degree of characters may vary across the different observation networks, the importance of characters as measured by the degree will usually remain the same.

### Betweenness Centrality

In a narrative social network betweenness centrality measures how much of a "connector" the character is; *i.e.* to what extent the character connects characters to other characters. Our results show that on average there is a high correlation between betweenness centralities of the simulated **co-occurrence** and **NLP** network and the simulated **manual** network, but the distributions of correlations are negatively skewed, so few simulations have low and sometimes even negative correlations. Therefore our automatically extracted networks usually give indication about which characters have high betweenness scores, but we should be careful to completely trust these.

We see a similar pattern in the data. Figure 4.7 shows the rankings of betweenness centralities of Joey in the real data for the **manual** and **co-occurrence** season networks. Joey has the highest or second highest betweenness ranking in every season except Season 1 and Season 4 (and Season 10 in the **co-occurrence** network). The rankings of betweenness centrality for the other core characters are in Appendix A.1. While the exact value of the centrality may change in the different datasets, the ranking of the character is similar, so the analysis of character importance would be similar also. We analyse the metric differences between the two datasets in detail in Chapter 5.

### Eigenvector Centrality

The eigenvector centrality of a node is similar to the degree, but weights connections with more highly connected nodes higher. Figure 4.6 shows that for both the **co-occurrence** and **NLP** networks, the eigenvector centrality scores are very similar to those of the **manual** network. Similarly to degree, the importance of characters as measured by the eigenvector centrality is unlikely to change with the extraction method of the social network, so analysis results will also not change.

### Closeness Centrality

In a narrative social network closeness centrality measures how close (in terms of geodesics) a character is to all the other characters, which can indicate whether the character is central to the plot. We find that the correlation between closeness centralities for characters in the **co-occurrence** and **manual** networks, and in the **NLP** and **manual** networks are quite high. Therefore, similarly to weighted degree and eigenvector centrality, when analysing a narrative through the closeness centrality of its characters in a **co-occurrence** or **NLP** network, we can be confident in the results.

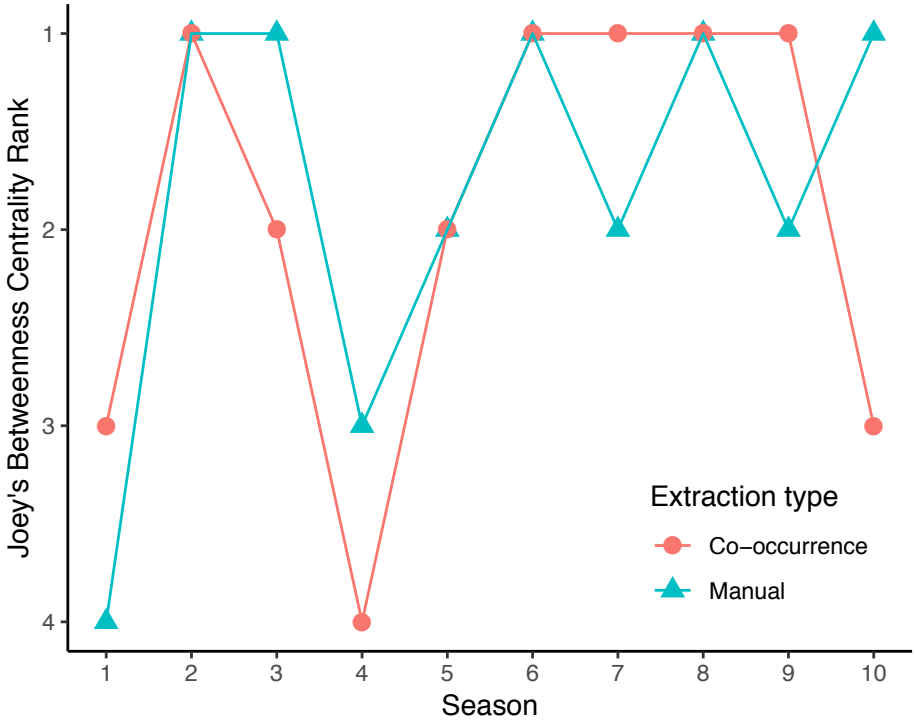



Figure 4.7: Betweenness centrality ranks of Joey over the 10 seasons of *Friends* for the **manual** (blue) and **co-occurrence** (red) network datasets.

### Local Clustering Coefficient

The local clustering coefficient of a character measures how much a character is part of a cluster. Figure 4.6 shows that the local clustering coefficient is highly variable in the simulated **co-occurrence** and **NLP** networks. The correlations between the clustering coefficients of nodes in the **co-occurrence** and **manual** networks are moderate and positive on average, but range from -0.93 to 1. The correlations between the clustering coefficients of nodes in the **NLP** and **manual** networks are centred higher than the correlations for the **co-occurrence** networks, but the range is similar.

The large range of correlations show that the random networks do not always give reliable rankings of character clusterings, so we should not trust automatically extracted networks when looking at local clustering coefficients.

## 4.4.3 Relationship metrics

Similarly to character metrics, we use edge weights to investigate the importance of relationships between characters. We correlate three sets of edge

weights to assess the accuracy of different types of relationship analyses:

1. *All weights* - including zero weights where there is no edge,

2. *Non-zero weights* - between characters that interact at least once in at least one of the networks, and

3. *Core weights* - between core characters, as these are usually the relationships we are most interested in.

Figure 4.8 shows the correlations between edge weights for the simulated **co-occurrence** and **manual** networks and simulated **NLP** and **manual** networks. Again, we use Spearman's correlation coefficient because we are interested in rank orderings rather than actual values.

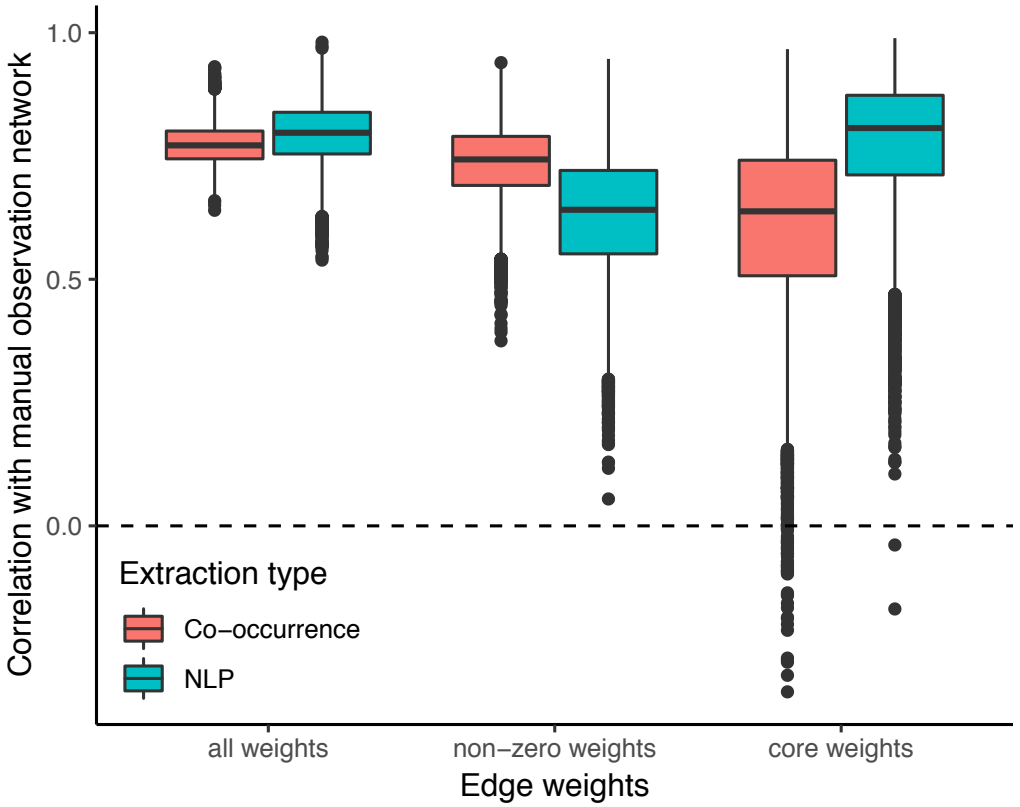



Figure 4.8: Box plots of the edge weight correlations between the simulated **manual** network and simulated **co-occurrence** (red) and **NLP** (blue) networks from the 10000 simulations.

### All Edge Weights

The correlation between all edge weights in the simulated **manual** and **co-occurrence** networks are high with little variance. There is more spread in the correlation between edge weights in the simulated **NLP** and the **manual** networks. The high correlations indicate that relationships that are important in the **manual** networks are also important in the **co-occurrence** and **NLP** networks.

### Non-zero Edge Weights

Correlations decrease, however, when we exclude edges with zero weights in both networks. This indicates that while we frequently get the correct set of interactions, the weights of those that interact are less accurate. The non-zero edge weight correlations are still high for the simulated **co-occurrence** networks, but the correlations vary greatly for the non-zero edge weights between the simulated **NLP** and **manual** networks.

### Core Edge Weights

If we only compare the edges between the six core characters, the simulated **NLP** networks are more highly correlated with the simulated **manual** networks. This is because the core characters are frequently in scenes together but do not necessarily interact. This makes inferring the relative strengths of each relationship difficult when we only observe who is in the scene (*i.e.* from the **co-occurrence** network), but **NLP** networks misdirect each interaction with the same probability, so edge weights between core characters are equally likely to be changed.

While the majority of correlations for simulated episodes are medium to high, the core edge weight correlations for both automatically extracted networks vary greatly. Therefore the analysis of relationships should be trusted more if we compare all characters than if we compare only the core characters.

## 4.5 Discussion

We modelled and compared three social network extraction methods for narrative analysis; manual extraction, co-occurrence network extraction and natural language processing (NLP) network extraction. Manual network extraction is time consuming but co-occurrence and NLP techniques introduce errors.

Our results show high correlation between the metrics of the **manual** networks and the automatically extracted networks, suggesting that for most narrative analyses we can extract the social network automatically and achieve similar results to the more time consuming manual extraction. We should, however, keep in mind that automatic extraction methods introduce some errors. For most metrics (size, total edge weight, edge density, weighted degree, betweenness centrality, eigenvector centrality and closeness centrality) these errors will have minimal effects on the *comparison* of global metrics over time and the importance of characters. A small set of metrics related to clustering (global and local clustering coefficient) are not reliable in the automatically extracted networks.

The importance of relationships in the automatically extracted networks correlate with those in the **manual** networks, however when we look more closely at relationships between core characters, the automated networks are not as reliable. The reliability of most narrative analyses using automatically extracted social networks means we can extract and analyse more social networks without the high time cost. The more narrative social networks we extract and analyse, the greater our understanding of narratives, literature and social structure.

Here we only investigated the effect of the extraction method on the television show *Friends*. With these comparison methods in place, one could check for consistent results in the other television shows and other types of narratives such as films and novels.

The core group of characters in *Friends* is intrinsic to the series, but extending the work to other narratives means we have to identify core characters in those narratives. A stochastic block model [14] could be used here to automatically identify the core group of characters. It is interesting to consider how the extraction approach might bias this identification, as some approaches to find core characters might be perturbed by distortion in clustering.

Also, our model did not take into account variation in scene lengths. Experimenting with parameters of the model, such as the length and number of characters in scenes and the error rate of the NLP extraction method could lead to further results about the quality of social network data as extracted using automated methods.

Finally, we chose the episode time frame for the social networks somewhat arbitrarily. In other narratives such a time frame may not exist, so future work could involve experimenting with different time frames in the analysis of narrative social networks.

Now that we understand the different network extraction types and their similarities and biases through simulation, we can move to a network anal-

ysis of the datasets themselves. In the next chapter, we analyse the **co-occurrence** and **manual** datasets and reinforce our findings from this chapter.

# Chapter 5

# The One With the Network Metric Analysis

> "Alright, kids, I gotta get to work. If I don't input those numbers... it doesn't make much of a difference."
>
> *Matthew Perry* as *Chandler Bing*
> *Season 1, Episode 1*

## 5.1 Background

In Chapter 4, we compared methods of extracting a social network from a narrative for the purpose of analysing the narrative. The social network of a narrative can tell us whether the social structure is similar to real social networks and who the important characters are, but also features of the narrative that are not obvious to the regular audience. In this chapter, we analyse and compare the two datasets representing the characters and relationships in *Friends* described in Chapter 3:

- the **co-occurrence** dataset, where interactions occur when characters appear in a scene together, and
- the **manual** dataset, where interactions occur when characters speak to, touch, or look at each other.

We analyse three types of network metrics; *global*, *character* and *edge*. We define these metrics in Chapter 2. Global metrics tell us attributes of the networks as a whole. Character metrics allow us to analyse the importance of characters with respect to different measures of importance. Edge weights represent the strength of relationships between characters.

We analyse the networks using three different time windows for the temporal component of the television series. A large time window will give general results about the series as a whole, whereas a small time window will allow us to analyse specific events throughout the series, and how these events affect the social network. We use the whole series, seasons and episodes as time windows in this chapter.

We perform a univariate analysis on both datasets for all time windows to understand the network structure and importance of characters and relationships. We also perform bivariate analysis to investigate how the networks change over time. We compare the temporal network with important events in *Friends*. Table 5.1 shows some of the main events that happen in each season of *Friends*, many of which can be inferred from inspecting the social network.

Our main findings are:

- the six core characters and their relationships are the essence of the series;
- while Chandler and Phoebe were originally meant to be side characters [41], Chandler becomes the most important character, yet Phoebe remains the least important; and
- Rachel and Ross's famous intermittent relationship is not as important as Chandler and Monica's relationship.

The first finding is almost trivial to fans and viewers of *Friends*, however the last two findings are rather surprising.

## 5.2 Network visualisation

### 5.2.1 Series view

The series network contains every character and every interaction throughout the series. Figures 5.1 and 5.2 show the networks of the entire series of *Friends* for the **co-occurrence** and **manual** datasets respectively. The blue nodes represent the core characters: Joey, Phoebe, Monica, Chandler, Rachel and Ross. The red nodes represent every other character – the non-core

| Season | Events |
|---|---|
| 1 | Joey and Chandler live together<br>Rachel moves in with Monica<br>Ross's son Ben is born |
| 2 | Ross dates Julie, then Rachel<br>Carol and Susan get married<br>Monica dates Richard |
| 3 | Phoebe meets her half brother, Frank Jr<br>Rachel quits waitressing<br>Rachel and Ross have a break |
| 4 | Rachel and Ross break up<br>Ross dates (and almost marries) Emily<br>Phoebe is surrogate for Frank Jr.'s triplets |
| 5 | Monica and Chandler start dating<br>Rachel is hired by Ralph Lauren<br>Ross and Rachel marry in Las Vegas |
| 6 | Chandler moves in with Monica<br>Rachel moves in with Phoebe<br>Ross dates Elizabeth |
| 7 | Rachel is promoted and dates Tag<br>Joey gets re-hired on Days of Our Lives<br>Monica and Chandler get married |
| 8 | Ross and Rachel have a one-night-stand<br>Ross dates Mona<br>Rachel gives birth to Emma |
| 9 | Chandler is relocated to Tulsa, then quits his job<br>Phoebe dates Mike<br>Joey, then Ross dates Charlie |
| 10 | Charlie breaks up with Ross<br>Phoebe and Mike get married<br>Monica and Chandler adopt twins and move to Westchester |

Table 5.1: Timeline of key events in the ten seasons of *Friends*.

characters. The radii of the nodes are proportional to the weighted node degrees. Both series networks are laid out using the Fruchterman-Reingold layout algorithm [52]. This iterative process takes each edge to be a spring and calculates the attractive forces from edges and the repulsive forces from other nearby nodes using the node positions from the current iteration, then adjust the positions accordingly. The edge weights determine the strength

of each spring, so that characters who interact many times are likely to be close together in the layout. The edge weights in the static networks are the number of times the characters interacted over the series.

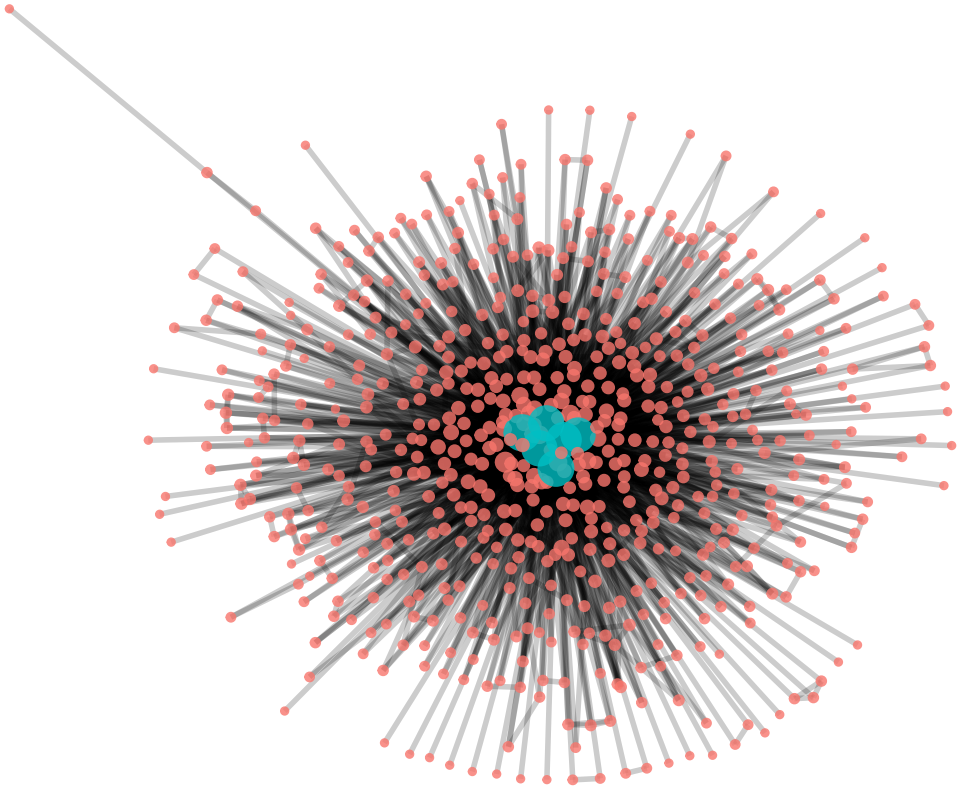


Figure 5.1: Series network for the **co-occurrence** dataset. The blue nodes represent the core characters and the red nodes represent the non-core characters. The radius of the nodes is proportional to the weighted node degree.

Comparing Figure 5.1 and Figure 5.2 shows that both series networks are centred around the core characters. This is because at least one of the core characters is part of most of the interactions that occur throughout the series, so the force network algorithm places them in the centre. The **co-occurrence** network appears denser than the **manual** network with less nodes of degree 1. We will quantify this in Section 5.3. We also notice that the **manual** network contains more non-core characters that interact only with other non-core characters. This is because it is very unlikely that non-core characters will be in a scene together without core characters. It is more likely a non-core character will interact with another non-core character, but

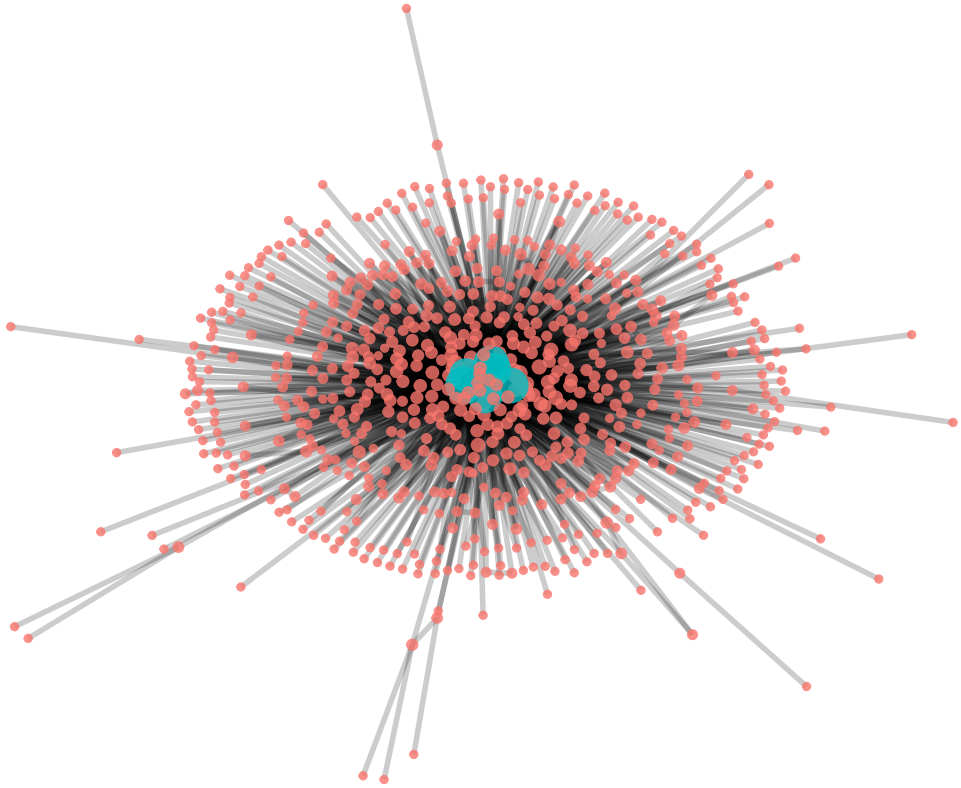


Figure 5.2: Series network for the **manual** dataset. The blue nodes represent the core characters and the red nodes represent the non-core characters. The radius of the nodes is proportional to the weighted node degree.

never interact with a core character.

### 5.2.2 Season view

There are 10 seasons of *Friends*, so the season view of the networks contains 10 networks. Each network is labelled with a season number, and contains every character and interaction that appears in that season. With the 10 distinct networks, we can start investigating temporal attributes of the *Friends* social network. We also investigate differences in metrics between the entire series and individual seasons.

Figure 5.3 shows the **co-occurrence** network for Season 1. As expected, the core characters (in blue) are central to the season network, and the non-core characters range in importance. Figure 5.4 shows the **manual** network for Season 1. Figures for Seasons 2 to 10 are in Appendix A.2. Compar-

ing Figure 5.3 and Figure 5.4, the **manual** network appears to have fewer connections between non-core characters than the **co-occurrence** network, which we quantify in Section 5.4.10, meaning non-core characters are more likely to be in a scene together than to actually talk to, look at or touch each other. Also note that there are far fewer characters in the season networks than the series networks in Figure 5.1 and Figure 5.2, as many characters don't appear in Season 1.

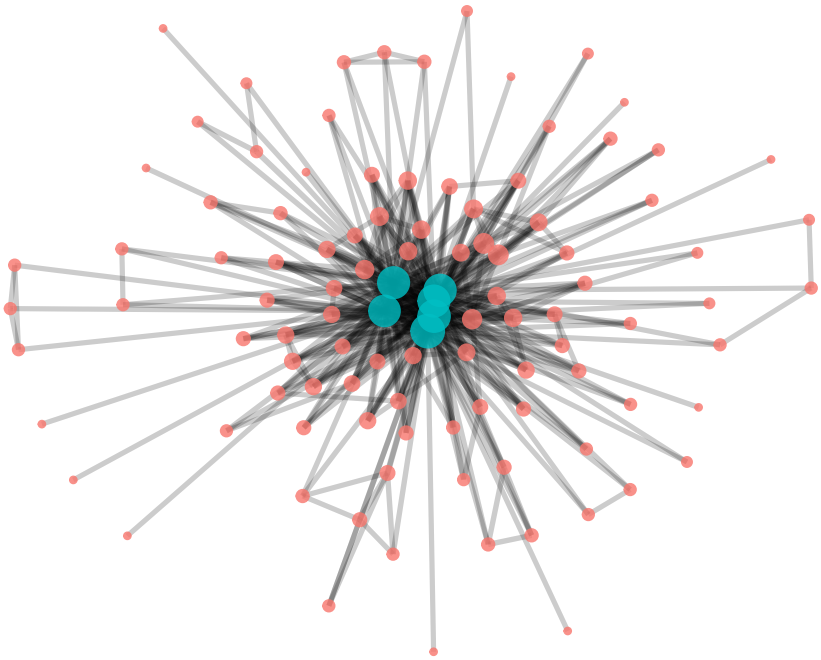


Figure 5.3: Season 1 network for the **co-occurrence** dataset.

### 5.2.3 Episode view

The episode view is the shortest timeframe we analyse. There are 227 episode networks in the **co-occurrence** dataset and 236 episode networks in the **manual** dataset. As discussed in Chapter 3, episodes that were originally aired as single, double-length episodes are counted as a single episode in the **co-occurrence** dataset, but two episodes in the **manual** dataset as they were presented for re-runs and DVDs.

Figure 5.5 shows the network for Season 9, Episode 13 – *The One Where Monica Sings* – for both the **co-occurrence** and **manual** networks. Figure 5.5a and Figure 5.5b are quite similar, but there are inconsistencies in

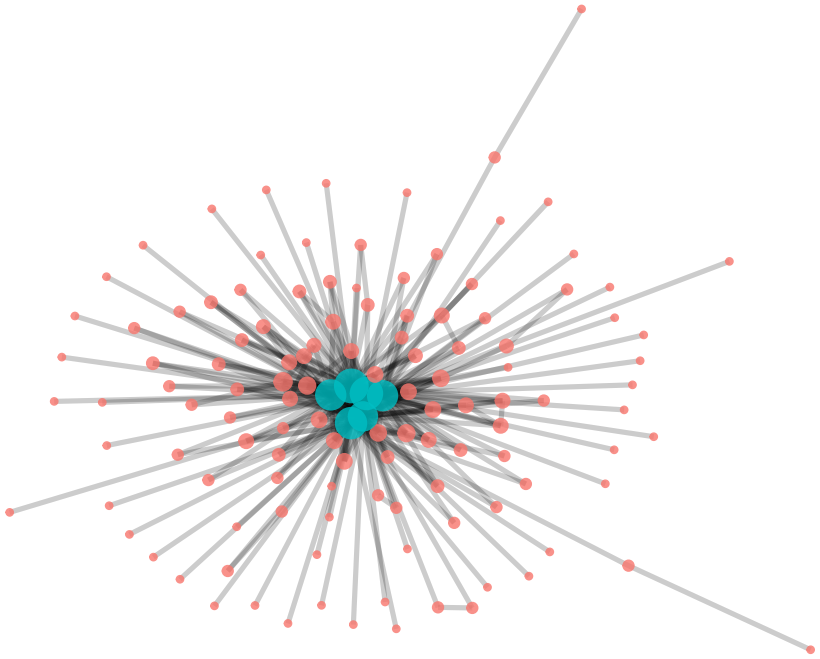


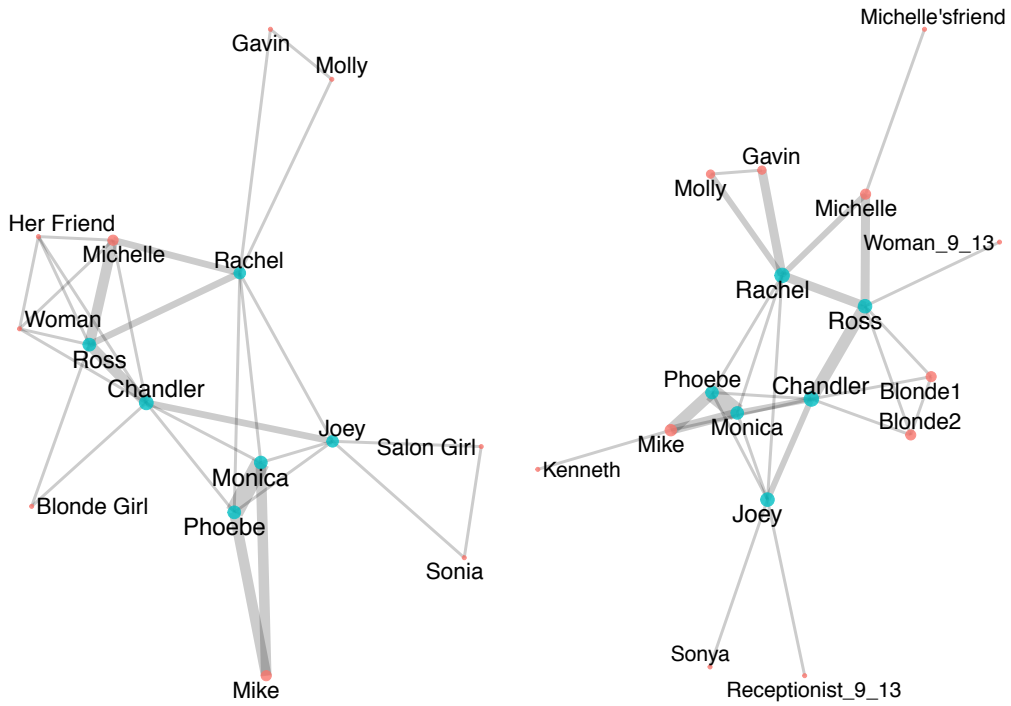
Figure 5.4: Season 1 network for the **manual** dataset.

character names, (*e.g.* Michelle's friend is called "Her Friend" in the **co-occurrence** dataset, but "Michelle'sfriend" in the **manual** dataset). There are also differences in edges where characters share a scene but don't talk to, touch or look at each other (*e.g.* "Sonia" and "Salon Girl" interact in the **co-occurrence** dataset, but their equivalent nodes "Sonya" and "Receptionist_9_13" do not interact in the **manual** network). Finally, there are differences in some edge weights between the two datasets. For example, in the **co-occurrence** network, Rachel, Molly and Gavin share one scene, so they form a clique with all edge weights 1. In the **manual** network, however, Rachel interacts with Gavin and Molly more than they interact with each other. We do not yet know how important these differences will be, so we analyse both datasets.

Visualisations of all **manual** and **co-occurrence** episode networks are presented at `https://friends-network.shinyapps.io/ingenuity_app/`.

## 5.3 Global metric univariate analysis

Global metrics give us an understanding of the "big picture" of the social network. Here we examine the global metrics of the **co-occurrence** and




(a) **co-occurrence** dataset (b) **manual** dataset

Figure 5.5: Networks of Season 9, Episode 13 – *The One Where Monica Sings* for the **co-occurrence** and **manual** datasets. This is the 207th episode in the **manual** dataset and 200th in the **co-occurrence** dataset.

**manual** datasets for three different timeframes; series view (Table 5.2), season view (Figure 5.6) and episode view (Figure 5.7). We examine 10 metrics, which we define in Chapter 2: size, total edge weight, total number of edges, average edge weight, density, average degree, average path length, diameter, clustering coefficient and size of the largest clique.

### 5.3.1 Size

The **manual** series network has more characters than the **co-occurrence** series network because it includes characters that do not speak, such as Joey and Chandler's pets Duck and Chick. The **manual** network also distinguishes between unnamed characters, which is discussed further in Section 5.4.10.

The size of some of the **manual** season networks is also larger than the size of the **co-occurrence** season networks (Figure 5.6), also due to not distinguishing between unnamed characters in the **co-occurrence** networks.

| | Co-occurrence | Manual |
|---|---|---|
| size | 671 | 746 |
| totalEW | 18574 | 16569 |
| totalE | 2695 | 1609 |
| avEW | 6.89 | 10.30 |
| density | 0.00599 | 0.00290 |
| avDeg | 8.03 | 4.31 |
| avPath | 2.27 | 2.59 |
| diameter | 4 | 5 |
| clustering | 0.0541 | 0.0335 |
| clique | 13 | 10 |

Table 5.2: Table of global metrics: size, total edge weight (totalEW), number of edges (totalE), average edge weight (avEW), density, average degree (avDeg), average path length (avPath), diameter, clustering coefficient (clustering) and size of the largest clique (clique) for the **co-occurrence** and **manual** series networks.

However, the median season network size is smaller in the **manual** dataset.

At the episode level, the size of the **co-occurrence** and **manual** networks are similar (Figure 5.7), with 11 characters in the median episode in both datasets, *i.e.* the six core characters (Chandler, Joey, Monica, Phoebe, Rachel and Ross), with five others, as the core characters are in every episode.

### 5.3.2 Total Edges and Edge Weights

The total edge weight is larger for the **co-occurrence** series network than the **manual** series network, but only by 13% (Table 5.2). The total number of edges is larger in the **co-occurrence** series network by 60%, meaning many more character pairs interact at least once in the **co-occurrence** network than in the **manual** network over the series. This is not surprising as the requirement for "interaction" is more difficult to achieve in the **manual** network, as most of the time when characters talk to, touch or look at each other, they are in the same scene anyway. The total edge weight is not so greatly impacted as several interactions can happen in a single scene, but not necessarily between all pairs of characters.

Similarly, Figure 5.6 shows a large difference in total edges between the two datasets, and a smaller difference in total edge weights. Figure 5.7 shows that the distribution of total edges for each episode is centred higher for the **co-occurrence** dataset, and there are select episodes that predominantly add to the total number of edges overall.

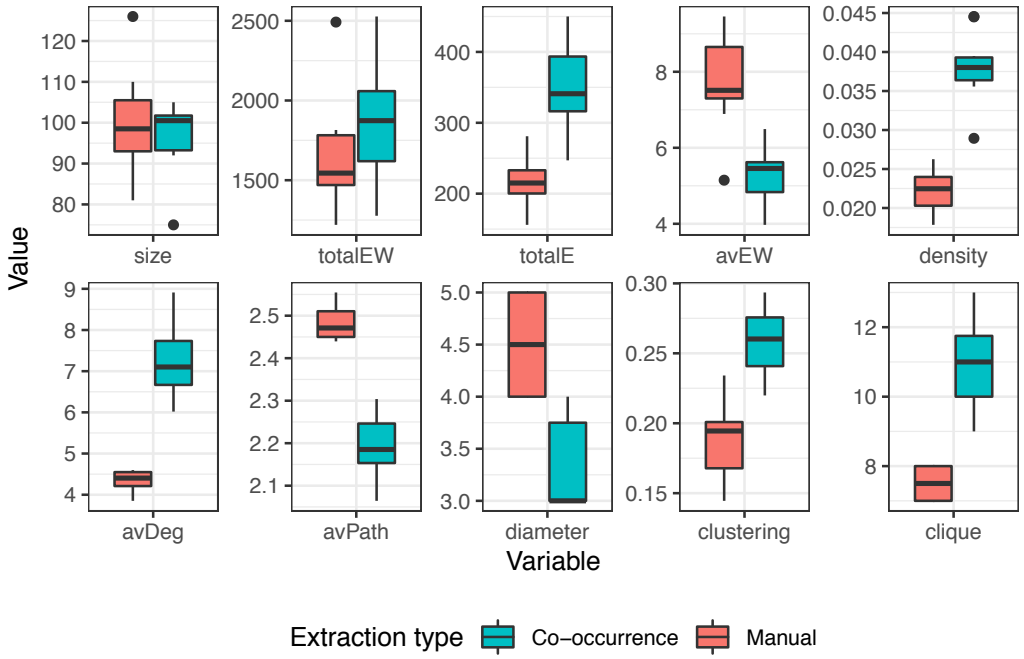



Figure 5.6: Box plots of global metrics: size, total edge weight (totalEW), number of edges (totalE), average edge weight (avEW), density, average degree (avDeg), average path length (avPath), diameter, clustering coefficient (clustering) and size of the largest clique (clique) for the **co-occurrence** (blue) and **manual** (red) season networks.

On the other hand, the average edge weight is larger in the **manual** series network compared to the **co-occurrence** series network. Less character pairs interact in the **co-occurrence** network, but of the characters that interact at some point, their relationships are stronger, with more interactions on average.

We see this clearly in Figure 5.6, where, apart from one season (Season 8, which had the lowest average edge weight in both datasets), the average edge weights of the **manual** season networks are larger than any of the average edge weights of the **co-occurrence** season networks.

Interestingly, the difference is less notable in the episode view (Figure 5.7), but the average edge weight of the **manual** episode networks are still larger than for the **co-occurrence** networks.

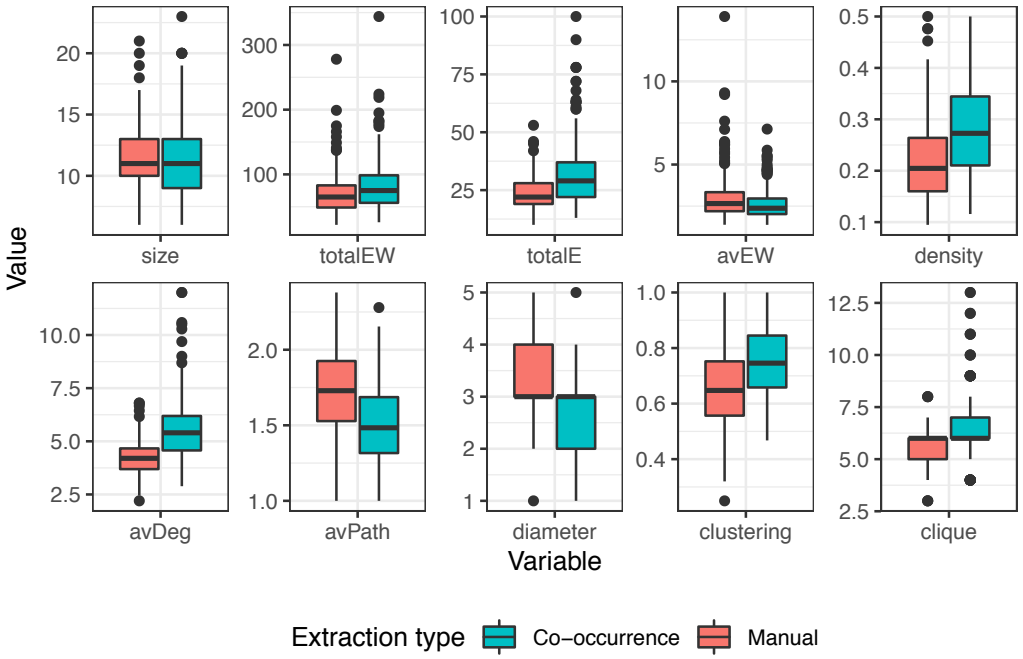



Figure 5.7: Box plots of global metrics: size, total edge weight (totalEW), number of edges (totalE), average edge weight (avEW), density, average degree (avDeg), average path length (avPath), diameter, clustering coefficient (clustering) and size of the largest clique (clique) for the **co-occurrence** (blue) and **manual** (red) episode networks.

### 5.3.3 Density

The density of the **co-occurrence** series network is more than double the density of the **manual** static network (Table 5.2) because many characters that do not directly interact are in scenes together. Both densities are small, so many character pairs never interact throughout the entire series. We see in Figure 5.1 and Figure 5.2 that there are many 'extra' characters in both series networks who only interact with one or two characters - leaving possible connections with every other character absent. This sparsity is normal in large social networks [91], and especially expected in the series network of a television show, as many characters appear in only one episode, so never get the chance to interact with anyone not in the episode.

We also see greater density in the **co-occurrence** season networks in Figure 5.6 for the same reason, however the densities in the season networks are much larger than in the series networks. As the time windows for the season networks are smaller, there are fewer extras who interact with few

others, and fewer opportunities for characters to only appear in one episode.

Figure 5.7 further confirms this, as the densities for both datasets at the episode view are an order of magnitude larger than for the season view. The difference between edge densities in the **co-occurrence** episode networks and **manual** episode networks is not as large as in the series and season view case, but the density is still larger on average in the **co-occurrence** dataset.

### 5.3.4 Average degree

Similarly to density, the average (unweighted) degree of the **co-occurrence** series network is almost double the average degree of the **manual** network. The average character in the **co-occurrence** dataset interacts with approximately 8 other characters, compared to the 4.31 characters the average character in the **manual** dataset interacts with.

Figure 5.6 shows that the range of average degrees in the **co-occurrence** season networks is much wider than the range for the **manual** network.

At the episode level, however, the average degrees in both datasets have a wide range (Figure 5.7), and while it appears the average degree of the **co-occurrence** networks are generally larger than those of the **manual** episode networks, the differences are much smaller than for the season and series views.

We notice that as the time window shrinks, the average degree of the **co-occurrence** networks decrease, *i.e.* as we merge **co-occurrence** episode networks together to form season and series networks, the average degree increases. The average degree in the **manual** networks, however, remain reasonably constant in the series, season and episode networks. Figure 5.8 shows box plots of the average degree in the **co-occurrence** and **manual** networks at the series, season and episode views, and also the average degrees of the core and non-core characters for these time windows.

The average degree of the core characters increases from episode to season to series view in both datasets as over more episodes, the core characters are able to interact with more non-core characters than in a single episode. In the **co-occurrence** networks, we also see an increase in average degrees of non-core characters as we increase the time window, but the average degrees of non-core characters in the **manual** networks remains constant. The pattern of non-core character average degrees is similar to the average degree overall because there are many more non-core characters than core characters. The **co-occurrence** networks overestimate the degree of non-core characters because many non-core characters are in scenes with several other characters, but only interact with a single character.

Figure 5.8 also provides evidence that the core characters are much more

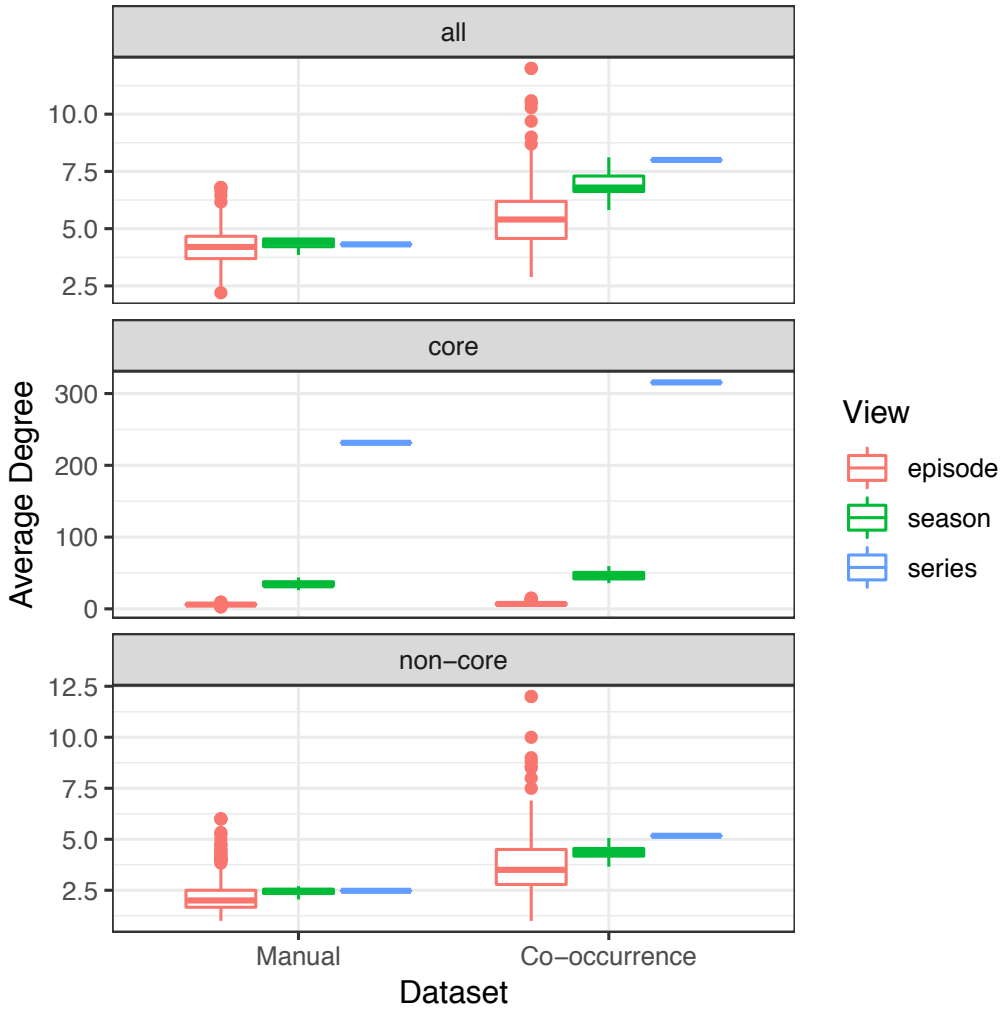



Figure 5.8: Box plots of the average (unweighted) degree in the **co-occurrence** and **manual** networks at the episode, season and series views for all characters (all), the core characters (core) and the non-core characters (non-core).

important than any of the non-core characters, as the average degree of the core characters is more than that of the non-core characters in both datasets for all time windows.

### 5.3.5 Average path length

The average geodesic path length of the **co-occurrence** series network (2.27) is slightly smaller than the average path length of the **manual** series net-

work (2.59), but both are smaller than many real-world social networks. For example, Watts and Strogatz [124] find the network of film actors co-starring in a film has an average path length of 3.65.

Likewise in narrative networks, paths tend to be larger. For example, Mac Carron and Kenna [76] showed that networks representing the characters in Icelandic Viking Sagas have mean path lengths between 3.4 and 5.7. The average path lengths in the social networks of the iconic narratives *Beowulf*, *Táin Bó Cuailnge* and *Iliad* are 2.37, 2.76 and 3.54 respectively [75]. The average path length in the **co-occurrence** series network is smaller than all of these average path lengths, which suggests the characters appear closer than we would expect from a social network. The average path length for the **manual** social network is similar to that of *Beowulf* and *Táin Bó Cuailnge* social networks, which are on the small side compared to other narrative social networks. Small average path lengths in narratives that are focused on a core group of characters, such as *Friends*, *Beowulf* and *Táin Bó Cuailnge*, could be because most characters are connected through the core group, and the core group is fully connected.

One technique to asses the suitability of a small world network model (as described by Watts and Strogatz [124]) is to compare the average path length of the network with the average path length of a random network with the same number of nodes and edges. We also compare the clustering of the original and random network and find that the path length is slightly larger in the random network, but the clustering coefficient is much lower. We find that out of 100 Gilbert-Erdös-Rényi (GER) [48, 54] random networks with the same number of nodes and edges as the **co-occurrence** network, the average of the average path lengths is 3.35, which is slightly larger than the 2.27 from the original **co-occurrence** network. The average of the average path lengths out of 100 random networks with the same number of nodes and edges as the **manual** network is 4.66, which is 1.8 times the 2.59 from the original **manual** network.

The average path lengths in the season networks are similar to the average path lengths in the series network (Figure 5.6), but at the episode view, the is a wide range of average path lengths for both datasets, ranging from 1, to over 2.25 (Figure 5.7). The average path length is generally smaller in the episode networks because there is usually a single story for an episode, so most characters are closely related. Characters in different episodes are usually linked through one or two core characters at most.

### 5.3.6 Diameter

The diameters of the **co-occurrence** and **manual** series networks are 4 and 5 respectively (Table 5.2). These agree with previous results in social networks, known as "six degrees of separation" [68, 82], which says that any two individuals are connected by at most five others. This concept gives rise to similar concepts in different social circles, such as the "six degrees of Kevin Bacon" [40], which says that nearly every movie actor is no more than six steps from appearing in a film with Kevin Bacon, and the "Erdős number" [63], which is the number of co-authorships someone is away from co-authoring a paper with Paul Erdős. The *Friends* networks have diameters smaller than 6 because many characters are connected through at least one of the six core characters. The show rarely shows characters interacting without at least one of the core characters being present.

In comparison, the diameters in the social networks of *Beowulf*, *Táin Bó Cuailnge* and *Iliad* are 6, 7 and 11 respectively [75], even though they have 74, 404 and 716 characters respectively, so the *Friends* networks have smaller diameters than other narratives. This alludes to the six core characters being the main focus of *Friends*, as most characters are connected through the core group.

The diameters of the series networks form the maximum of the diameters for the season and episode networks (Figure 5.6 and Figure 5.7), except for one episode in the **co-occurrence** dataset with a diameter of 5. The **co-occurrence** episode with diameter 5 is Season 7, Episode 7, and is in Figure 5.9. Notice that many of the core characters don't interact, so the longest path in this episode doesn't contribute to the longest path in the Season 7 **co-occurrence** network, where all the core characters interact.

Excluding Season 7, Episode 7, the episode network diameters range from 1 to 4 for the **co-occurrence** networks and 1 to 5 for the **manual** networks. There is a smaller range in the diameters of the season networks, where the **co-occurrence** network diameters are either 3 or 4, and the **manual** network diameters are either 4 or 5.

### 5.3.7 Clustering coefficient

The clustering coefficient in the **co-occurrence** series network, 0.0541, is larger than the clustering coefficient in the **manual** network, 0.0335 (Table 5.2), but compared to many real-world social networks and other narrative social networks, both clustering coefficients are small. For example, Watts and Strogatz [124] found that the network of film actors co-starring in a film has a clustering coefficient of 0.79.

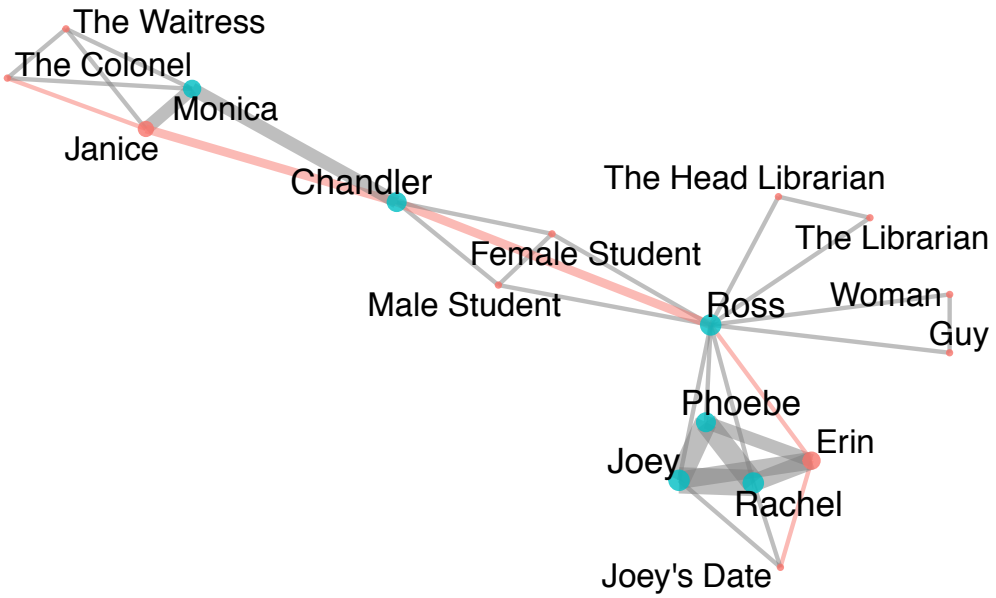



Figure 5.9: **Co-occurrence** network for Season 7, Episode 7: *The One with Ross's Library Book*. The longest geodesic path has length 5. One path of length 5 is between The Colonel and Joey's Date, which is highlighted in red.

Mac Carron and Kenna [76] showed that networks representing the characters in Icelandic Viking Sagas have clustering coefficients between 0.4 and 0.6. The clustering coefficients in the networks for *Beowulf*, *Táin Bó Cuailnge* and *Iliad* are 0.69, 0.82 and 0.57 [75]. Bersini *et al.* [125] extracted social networks from popular books and found the clustering coefficients ranged from 0.21 to 0.54.

The *Friends* networks have much lower clustering coefficients because they are structured differently. Alberich *et al.* [17] showed the clustering coefficient in the Marvel superhero collaboration network is 0.012. This network is more comparable to the *Friends* networks because it has characters that are significantly more 'core' to the network, and many other characters are connected through the core characters, as opposed to several groups, or clusters, which come together to form the whole social network.

We simulate 100 GER random networks with the same number of nodes and edges as the **co-occurrence** series network and calculate the mean of the clustering coefficients, 0.0122 (compared to 0.0541). We do the same for the **manual** networks and get a mean clustering coefficient of 0.0058 (compared to 0.0335). These random clustering coefficients are significantly less than the clustering coefficients of the original networks (see t-tests in Appendix B.2), which suggests some small-world network attributes, but the

differences are not as drastic as one would expect in small-world networks.

There is more clustering in the season networks (Figure 5.6) and the episode networks (Figure 5.7). As we expect, in the season networks there is more clustering in the **co-occurrence** dataset than the **manual** dataset, but the difference is less prominent in the episode view. The minimum episode clustering coefficient for the **co-occurrence** dataset is 0.47, which is large for a random network, but in the reasonable range for other narrative networks. The minimum episode clustering coefficient for the **manual** dataset is 0.25, which is also in a reasonable range for a narrative, based on Bersini *et al.*'s analysis of narrative social networks in novels [125]. The maximum episode clustering coefficient in both datasets is 1, which corresponds to a fully connected network. This is large for a social network, but not unexpected in a small time window such as an episode.

### 5.3.8 Clique size

The largest clique contains three more characters in the **co-occurrence** series network than in the **manual** social network. There is only one clique with 10 characters in the **manual** network. The characters in the largest clique are the six core characters (Chandler, Joey, Monica, Phoebe, Rachel and Ross), along with Jack, Judy, Emily and "registrar". Most of these characters are central to the series, except the registrar who is included in the largest clique as he interacts with everyone at Ross and Emily's wedding in Season 4, Episode 24.

There are two cliques with 13 characters in the **co-occurrence** network. Both contain the six core characters. The extra seven characters in one clique are Uncle Dan, Parker, "man", "woman", Jack, Judy and Aunt Lisa. The other clique contains Judy, Gunther, "guy", "girl 1", "girl's voice", Mr. Greene and Mrs. Greene. There is a notable effect of not being able to distinguish between unnamed characters such as "man" and "woman" here, but interestingly Judy is the only non-core character in all three largest cliques.

Figure 5.6 and Figure 5.7 show that the maximum clique size in the **co-occurrence** dataset remains the same in the series, season and episode views. This means that the clique of 13 characters was formed in a single episode. Based on the characters in the clique, it is not surprising that the clique of 13 characters was formed in a scene at Ross and Emily's wedding in Season 4, Episode 24. In each season of the **manual** dataset, there are either 7 or 8 characters in the largest clique. In the episode view, the range of characters in the largest clique goes from 3 to 8 in the **manual** dataset, but 4 to 13 in the **co-occurrence** dataset.

## 5.4 Character metric univariate analysis

### 5.4.1 Core character metrics

Figure 5.10 shows the character metrics for the six core characters in the **co-occurrence** and **manual** series networks. Many of the character metrics across the two datasets have systematic differences even though the metric values in each network can differ. While the values of the metrics are interesting, for many character metrics we are more interested in how the characters score in these metrics compared to the other characters (*i.e.* to analyse their "importance"). To investigate the importance of characters, and to test whether there are also differences in the relative metrics of each character compared to the other characters, we plot the ranks of each character for each of these metrics. Figure 5.11 shows scatterplots of the character metric rank of each of the core characters in both the **co-occurrence** and **manual** series networks. Trends are visually similar for both methods of network construction.

We also look at the character metrics for the core characters using the season and episode time windows. Figure 5.12 shows box plots of the character metrics for the six core characters in the **co-occurrence** season networks, and Figure 5.13 shows box plots of the character metrics for the core characters in the **co-occurrence** episode networks. The season and episode box plots for the core characters in the **manual** networks are in Appendix A.3.

### 5.4.2 Degree

The degree of a character in the series network is the number of characters that interact with at some point in the series. Figure 5.10 shows the core characters in the **co-occurrence** networks systematically interact with more characters than their corresponding characters in the **manual** networks. This is not surprising as characters are likely to be in scenes with other characters that they never talk to, touch or look at, so they form an edge in the **co-occurrence** network but not the **manual** network.

Figure 5.11 shows that the degree ranks of every core character except for Ross and Joey, who are switched, are the same in the **co-occurrence** and **manual** series networks. Ross co-appears in scenes with more characters than any of the other core characters, however, Joey directly interacts with more characters than any other core character. In both datasets, Phoebe is the least "important" core character, as measured by the degree.

Figure 5.12 shows Ross has the highest degree out of any characters for a single season, but Figure 5.12 shows Ross and Rachel share the highest

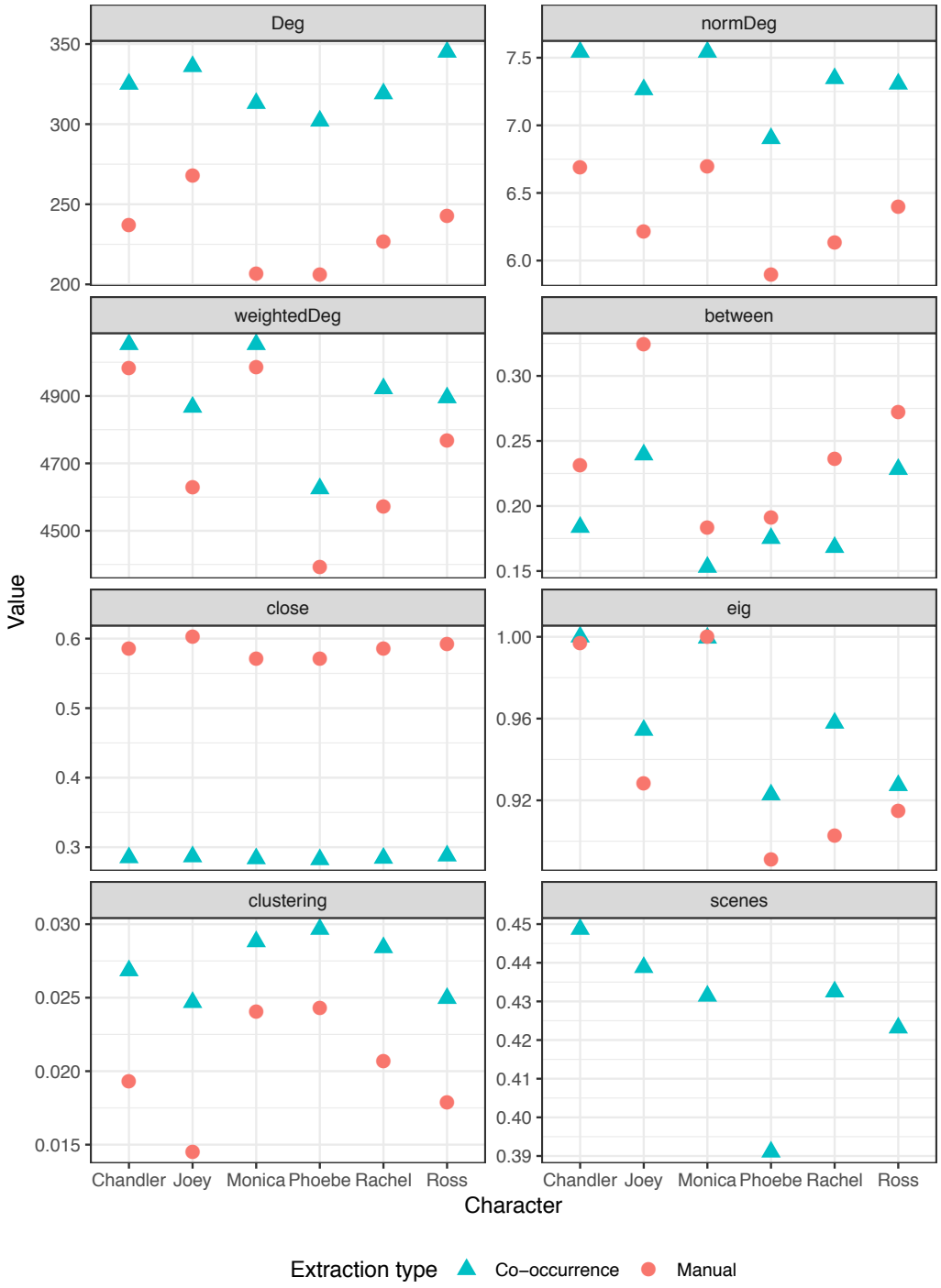



Figure 5.10: Scatterplots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen), local clustering coefficient (clustering) and proportion of scenes (scenes) for the six core characters in the **co-occurrence** (blue) and **manual** (red) series network.

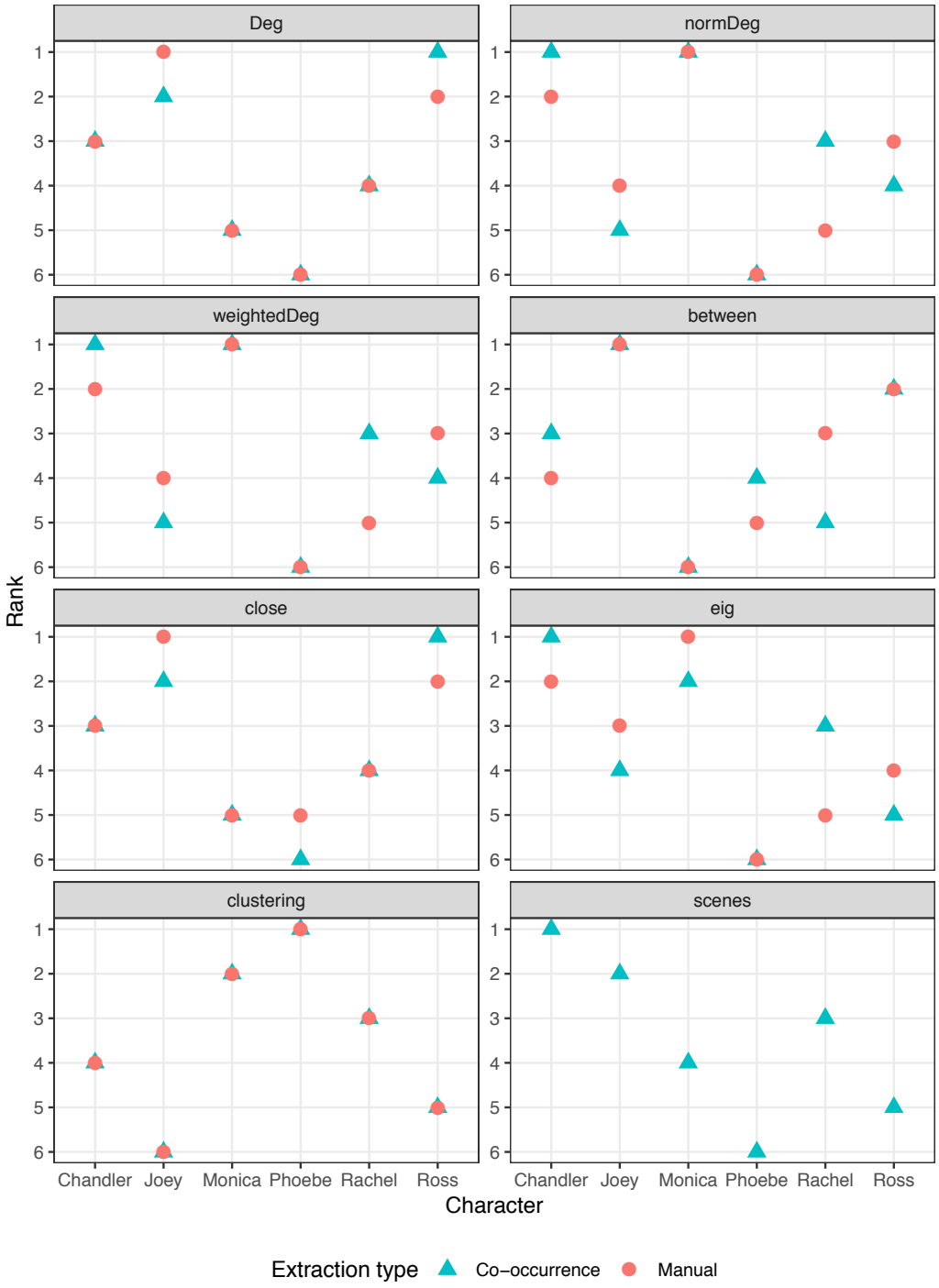



Figure 5.11: Scatterplots of ranks for character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen), local clustering coefficient (clustering) and proportion of scenes (scenes) for the six core characters in the **co-occurrence** (blue) and **manual** (red) series network.

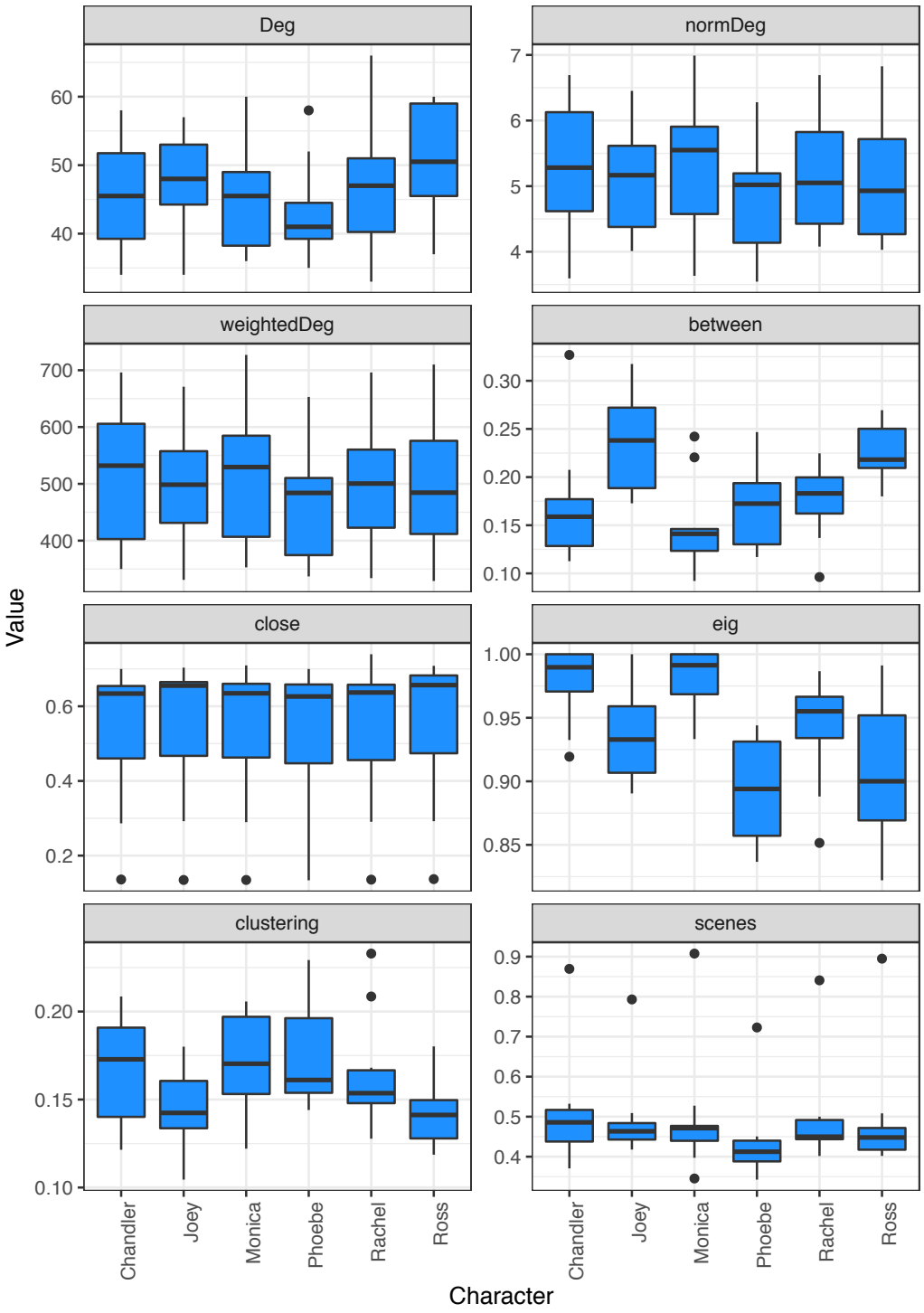



Figure 5.12: Box plots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen), local clustering coefficient (clustering) and proportion of scenes (scenes) for the six core characters in the **co-occurrence** season networks.

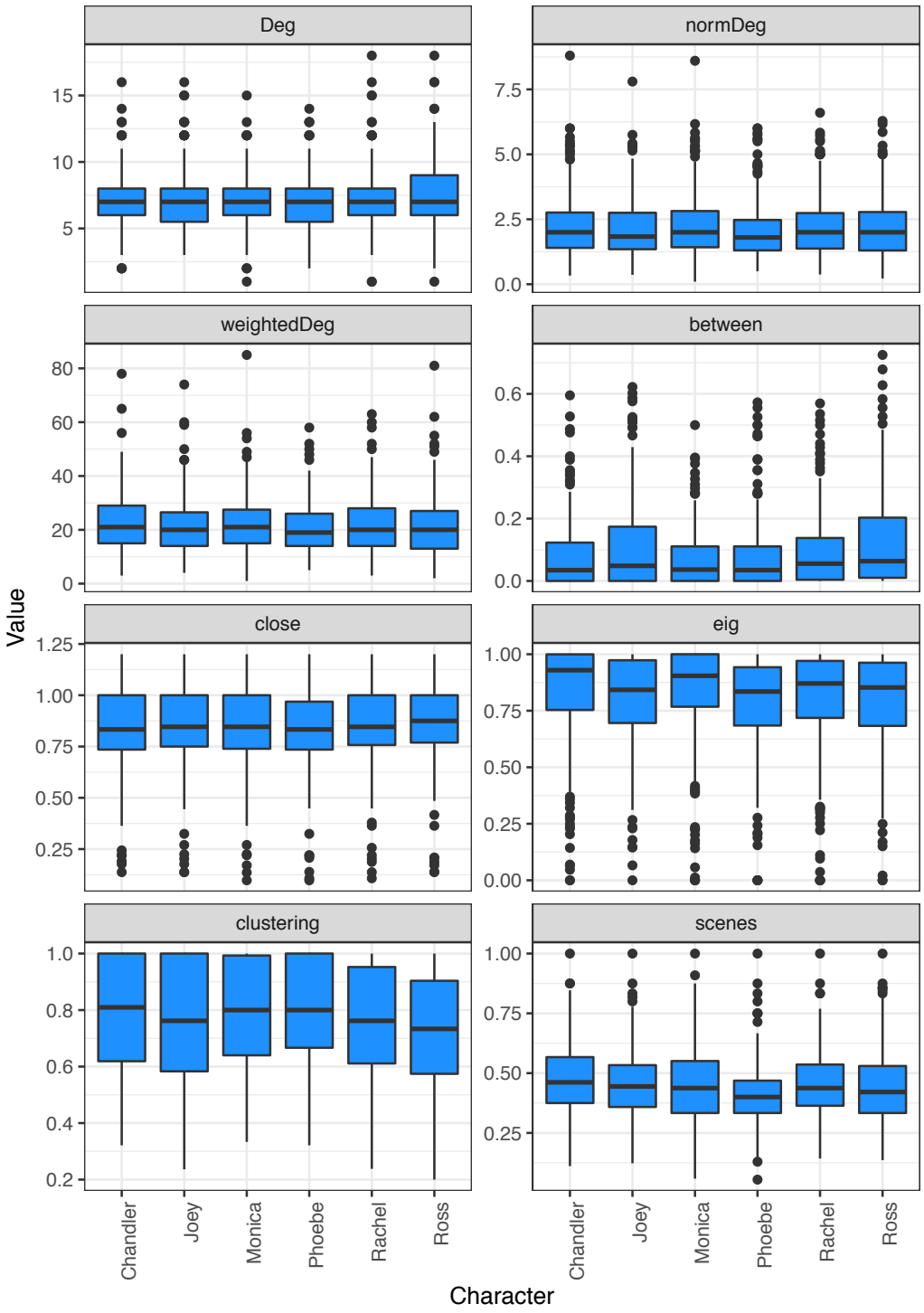



Figure 5.13: Box plots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen), local clustering coefficient (clustering) and proportion of scenes (scenes) for the six core characters in the **co-occurrence** episode networks.

degree for a single episode, but the distribution of degrees for every episode between the six core characters is quite even.

### 5.4.3 Normalised weighted degree

The normalised weighted degree of a character in the series network is the total number of interactions a character makes, divided by the total number of characters they could interact with $(n-1)$. In Figure 5.10, we see that both Chandler and Monica appear in a scene with each character more than 7.5 times on average, whereas Phoebe appears in a scene with each character less than 7 times on average. The normalised weighted degrees are lower for the **manual** network, where Chandler and Monica talk to, touch or look at every other character about 6.75 times on average, and Phoebe talks to, touches or looks at every other character less than 6 times on average.

Chandler and Monica have exactly the same normalised weighted degree in the **co-occurrence** series network and are both rank 1 (Figure 5.11), but Monica's normalised weighted degree is slightly higher than Chandler's in the **manual** series network. As in the degree case, Phoebe has the lowest normalised weighted degree out of the core characters in both datasets. Interestingly, while Ross and Joey interacted with more characters (higher degree), Monica and Chandler interact with each character more times (higher normalised weighted degree).

In the **co-occurrence** season networks, Monica has higher normalised weighted degree than Chandler on average (Figure 5.12), but both characters still rank the highest and Phoebe remains having the lowest normalised weighted degree.

Figure 5.13 shows that at an episode level, there is not much difference in normalised weighted degrees between the core characters, except for a handful of episodes marked as outliers, where Chandler, Joey and Monica have much higher normalised weighted degrees than any other characters in any episode.

### 5.4.4 Weighted degree

The weighted degree of a character in the series network is the total number of interactions the character makes in the series. The weighted degree character ranks are the same as in the normalised case, however Figure 5.10 shows the gap between metrics in the **co-occurrence** and **manual** networks is smaller in the unnormalised case. This is because we normalise by dividing (roughly) by the size of the network and there are more characters in the **manual** series network.

The weighted degree of the characters in the **co-occurrence** season networks are also very similar to the normalised case (Figure 5.12), but in the episode networks, there are more outlier episodes where Chandler, Joey, Monica, and now Ross have unusually high weighted degrees (Figure 5.13). The stand-out episode for Ross's weighted degree did not stand out for his normalised weighted degree because the size of the episode network is large.

### 5.4.5 Betweenness centrality

The betweenness centrality of a character measures the degree to which the character lies on the shortest unweighted path between two other characters. Figure 5.10 shows that the betweenness centralities for the core characters in the **manual** network are larger than the betweenness centralities for the same characters in the **co-occurrence** network. This is due to the added interactions in the **co-occurrence** network. There are more shortest paths between characters so either the core character is no longer on the shortest path, or is on a smaller proportion of the shortest paths in the **co-occurrence** series network.

Figure 5.11 shows that the betweenness ranks for the two highest ranked and the lowest ranked core characters are the same in both the **co-occurrence** and **manual** series networks. Joey lies on the shortest path between two characters most often, possibly because of all the characters from his acting career that only interact with him. Ross interacts at least once with many characters, as shown by his high degree, but Ross also has the second highest betweenness centrality score, so is also a connector of characters. The other core characters are rarely shown interacting with many characters from their jobs. Monica, for example, has the lowest betweenness centrality score out of the core group as we rarely see her interacting with characters outside the core group unless another core character is also present. This agrees with Bazzan's argument [21] that Monica is the "mother hen" of the group, as she interacts many times (as evidenced by her high weighted degree), but rarely to non-core characters.

Joey's betweenness centrality is also the highest in the **co-occurrence** season networks (Figure 5.12). While Monica has a low betweenness centrality for most season networks, there is one outlier season where she has a high betweenness centrality. Figure 5.13 shows that every character usually has a low betweenness centrality for every episode in the **co-occurrence** dataset, there are significant episodes where some or all of the core characters have much higher betweenness centralities. Joey and Ross are more likely to have high betweenness centralities in episodes than the other characters.

### 5.4.6 Closeness centrality

The closeness centralities of the core characters in the **manual** networks are approximately double the closeness centralities of the characters in the **co-occurrence** networks (Figure 5.11). This is the largest systematic difference between the two datasets for the character metrics. The small closeness centralities in the **co-occurrence** network are due to the series network having more than one component. The shortest path to characters in a different component is the size of the network (as defined in Chapter 2), so the closeness centrality, which takes the reciprocal of the shortest paths, will be much lower. A scatterplot of the closeness centralities of core characters in the largest connected component is in Appendix A.4.

The rankings of characters' closeness centralities are almost identical to the rankings of characters' degrees (Figure 5.11), except for Phoebe's rank in the **manual** network, which is equal last with Monica for the closeness centrality, but exclusively last for the degree. Therefore the analysis of closeness centrality rank is identical to the analysis of degree rank in the core characters for the series view.

Figure 5.12 shows that the rankings for the closeness centralities of the core characters in the **co-occurrence** season networks are similar to the rankings for the series network, but the distributions over the 10 seasons are very left-skewed. Most closeness centralities are above 0.5 like the closeness centralities for the **manual** series network, but there seasons with more than one component, which decreases the closeness centrality.

Similarly, Figure 5.13 shows that for most episodes, the core characters have high closeness centralities, but for some episodes (the episodes with more than one component), the closeness centralities are low.

### 5.4.7 Eigenvector centrality

The eigenvector centrality, as defined in Chapter 2, measures the importance of characters based on their interactions to other important characters. Figure 5.10 shows that Chandler and Monica have the highest eigenvector centralities in both the **co-occurrence** and **manual** series network, similarly to the weighted degree. This means Chandler and Monica frequently interact with other characters, especially other important characters such as the core characters. As Chandler and Monica are a couple from Season 5 until the end of the series (Table 5.1), it is likely that they interact with each other often, which could explain their high eigenvector centralities. The eigenvector centralities of the other four core characters are smaller for the **manual** network than for the **co-occurrence** network, so there is more of a difference

in their centralities compared to Chandler and Monica's.

Figure 5.11 shows that Phoebe has the lowest eigenvector centrality out of the six core characters in both series networks. We combine this results with the ranks for degree, weighted degree and closeness centrality and find that Phoebe seems to be the least important of the six core characters.

Chandler and Monica also have the highest eigenvector centralities in the **co-occurrence** season and episode networks (Figure 5.12 and Figure 5.13). At the season level, Ross and Phoebe both have low eigenvector centralities on average, but all characters have eigenvector centralities above 0.8 in every season, so are still highly central. At the episode level, the core characters usually have high eigenvector centralities, but there are many episodes where the core characters have low eigenvector centralities (less than 0.5). This is because every core character might not be central to every episode, but over an entire season, or series, the core characters certainly are the most central.

### 5.4.8 Clustering

The local clustering coefficient of a character measures the degree to which the character is part of a cluster. Figure 5.10 shows the local clustering coefficients are systematically larger for the characters in the **co-occurrence** network compared to the **manual** network. This is because we form clusters when we create cliques out of all the characters in a scene in the **co-occurrence** network.

Surprisingly, the local clustering coefficient ranks of the core characters are exactly the same for the **co-occurrence** and **manual** series networks (Figure 5.11). The local clustering coefficient ranks are exactly the opposite of the degree ranks for the core characters in the **manual** series network, which suggests that the characters that have less friends outside of group fit into clusters better.

In the **co-occurrence** season networks, Joey and Ross have the lowest local clustering coefficients. Notice that Joey and Ross were also the characters with high betweenness centralities as they interact with many characters that do not interact with anyone else. These sparse neighbourhoods of Joey and Ross mean that their clustering coefficients are low.

The clustering coefficients for every character are an order of magnitude larger than in the series networks (Figure 5.12). Similarly in the **co-occurrence** episode networks, every character has generally high local clustering coefficients (Figure 5.13), meaning the characters that interact with core characters in each episode often interact with each other too. As we make cliques out of every scene in the episode, the core characters, who appear in a high proportion of scenes, are often part of cliques, which contribute

to their high clustering coefficients.

### 5.4.9 Scenes

The proportion of scenes is only collected for the **co-occurrence** dataset. Figure 5.10 and Figure 5.11 show Chandler is in the most scenes (44.8%) and Phoebe is in the least scenes (31.1%) out of the core characters, which alludes to Chandler being the most important and Phoebe being the least important core character. The proportion of scenes the characters are in has similar trends to the eigenvector centralities of characters, but Monica and Joey swap ranks.

We see similar patterns in the proportion of scenes in the season networks (Figure 5.12), but overall each character is in a similar proportion of scenes, with the exception of Phoebe, who is in the smallest proportion of scenes. For most seasons, each character is in less than half of the scenes.

Figure 5.13 shows that the proportion of scenes the core characters are in range from less than 0.1 to 1. Ross and Chandler are in every scene of Season 3, Episode 2: *The One Where No One's Ready*. Joey, Monica, Phoebe, and Rachel are in every scene of Season 8, Episode 9: *The One with the Rumor*. These episodes only have 4 and 5 scenes respectively, and every scene takes place in Monica's apartment. Every character's proportion of scenes is centred between 0.4 and 0.5 for the episode view networks, so apart from the variation from episode to episode, the core characters get reasonably equal screen-time.

### 5.4.10 Non-core character metrics

The core characters are the most important characters in the series by all measures considered here, but the character metrics for the non-core characters are still of interest. Here, we look at the degree and weighted degree distributions of the non-core characters for both the **co-occurrence** series network and the **manual** series network.

Figure 5.14 shows the complementary cumulative degree distribution for the **co-occurrence** and **manual** series networks on the log-scale. Note that for both datasets, the degree distribution is far from the power laws commonly found in social networks. The degree of a character is the number of other characters they interact with over the series. The core characters' degrees for the series are much larger than the non-core characters'. Both degree distributions are heavily right skew, but the mode degree in the **co-occurrence** network is 3, whereas the mode degree in the the **manual** network is 1. As suggested in Section 5.2, there are more characters with degree

1 in the **manual** network than in the **co-occurrence** network.

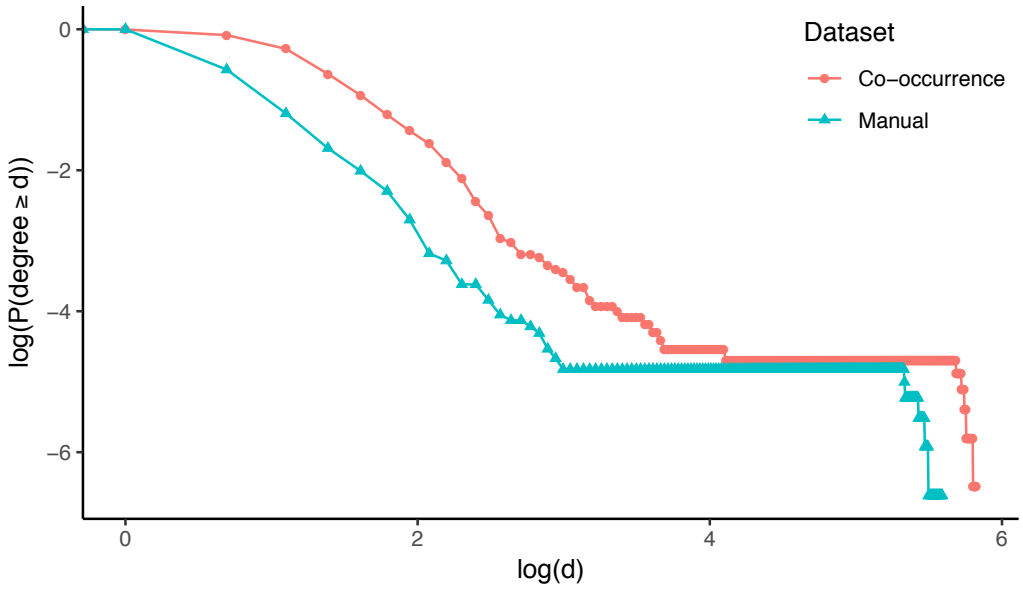



Figure 5.14: Complementary cumulative degree distribution of *Friends* characters in the series networks for the **co-occurrence** dataset and the **manual** dataset on the log-scale.

In the **co-occurrence** series network there is a non-core character with degree 60. Such a high degree character is not present in the **manual** network. The character with degree 60 is labelled "Woman". Many unnamed women interact with other characters over the series, but the **co-occurrence** algorithm doesn't distinguish between different women. In the **manual** network there are characters such as "Woman_7_7", who was in Season 7, Episode 7. We could label each of these non-recurring characters in the **co-occurrence** networks similarly, however it is difficult to recognise whether a character really is recurring. For example, how do we know the "Director" in Season 1, Episode 21 is not the same as the "Director" in Season 2, Episode 13, whereas the "Waiter" in Season 8, Episode 16 is the same as the "Waiter" in Season 9, Episode 14? Only by manually watching the series. Similarly, the non-core character with second highest degree (36) in the **co-occurrence** network is labelled "Man".

The highest ranked named characters in both datasets are the same; Gunther, Jack/Mr. Geller, Judy/Mrs. Geller, and Mike. The high degrees in the **co-occurrence** network, however, are higher than in the **manual** network.

Figure 5.14 shows the degree distribution of both datasets on the log-scale. Again we see that the degrees of characters in **manual** network is smaller than in the **co-occurrence** networks. We also see that the cumulative probability flattens out for the large gap between degrees of core and non-core characters. The cumulative probability is smaller for the **manual** network at this point is smaller than for the **co-occurrence** because the six core characters represent a smaller proportion of all characters (*i.e.*, the **manual** network has fewer characters).

Figure 5.15 shows the complementary cumulative distribution of weighted degrees of all non-core characters in the **co-occurrence** and **manual** series networks on the log-scale. The weighted degree distribution is similar to the degree distribution, but is even more right-skew. The log-scale cumulative distribution appears linear for the non-core characters, which suggests a power-law weighted degree distribution. However, as in Figure 5.14, there is a large gap between weighted degrees of core and non-core characters, as the core characters have much larger weighted degrees than any other characters. This is similar to the pattern observed by Chen *et al.* [36] in their analysis of the social network of Cao Xueqin's *Dream of the Red Chamber* .

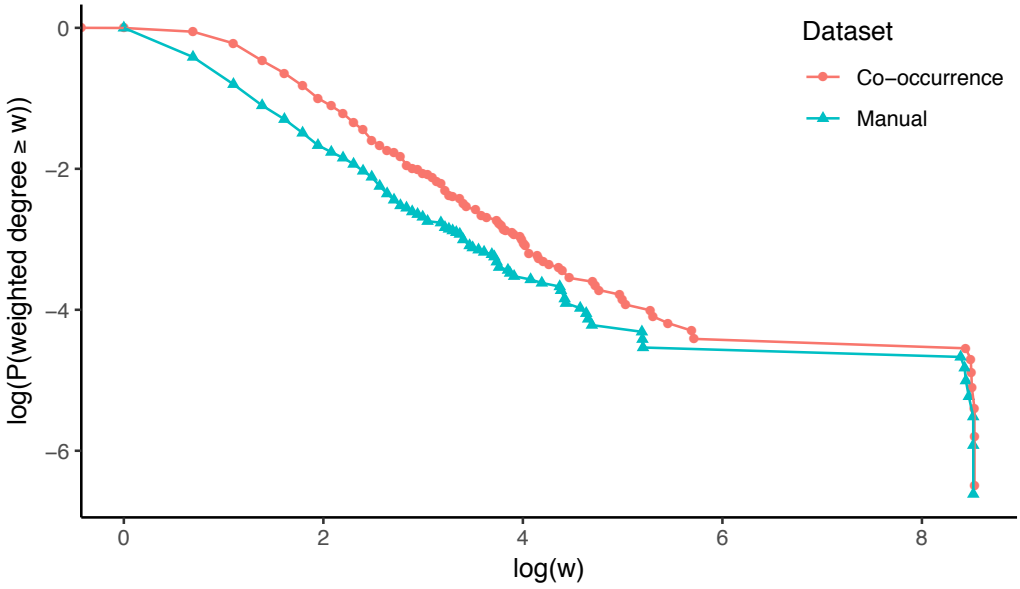



Figure 5.15: Complementary cumulative weighted degree distribution of *Friends* characters in the series networks for the **co-occurrence** dataset and the **manual** dataset on the log-scale.

Table 5.3 shows the non-core characters with the highest weighted degrees in the series networks. Note that Mrs. Geller and Judy are the same

character, and Mr. Geller and Jack are the same character. Most of the high degree characters in the **manual** network match with the high degree characters in the **co-occurrence** network, even though the weighted degree for each of these characters is higher in the **co-occurrence** network than the **manual** network. The exceptions to this are "Woman", who interacts many times in the **co-occurrence** network, and "Ben" and "Emma", who interact many times in the **manual** network.

| | Co-occurrence | weights | Manual | weights |
|---|---|---|---|---|
| 1 | Mrs. Geller | 303 | Judy | 182 |
| 2 | Mr. Geller | 296 | Mike | 181 |
| 3 | Gunther | 234 | Jack | 180 |
| 4 | Woman | 201 | Gunther | 109 |
| 5 | Mike | 196 | Janice | 105 |
| 6 | Emily | 153 | Ben | 103 |
| 7 | Carol | 148 | Carol | 97 |
| 8 | Janice | 144 | Emma | 84 |

Table 5.3: Table of weighted degrees of the 8 most highly weighted non-core characters in the co-occurrence and manual static networks for the social network of *Friends*.

As discussed, the character "Woman" is usually a different, unnamed character every appearance. The **co-occurrence** doesn't distinguish between these characters and so this node in the **co-occurrence** series network has a high weighted degree. The unweighted degree of "Woman" is significantly higher than that of any other non-core character, but she only has the 5th highest weighted degree. This difference shows that a "Woman" interacts with an unusually large range of characters compared to the number of interactions she has with each of those characters. This is not surprising as the node represent different characters and hence doesn't have any strong connections with any other characters. Removing "Woman" from the analysis also removes many single interaction edges, however much of the analysis is unchanged. Figures and tables of network metrics for the **co-occurrence** network with the "Woman" node removed are in Appendix A.5.

Ben and Emma have high weighted degrees in the **manual** network, but not the **co-occurrence** network because they are both babies or young children throughout the series. Ben is Ross and Carol's child, born at the end of Season 1, and Emma is Ross and Rachel's child, born at the end of Season 8 (Table 5.1). Recall the **manual** network defines an interaction as two characters talking to, looking at or touching each other. The **co-occurrence**

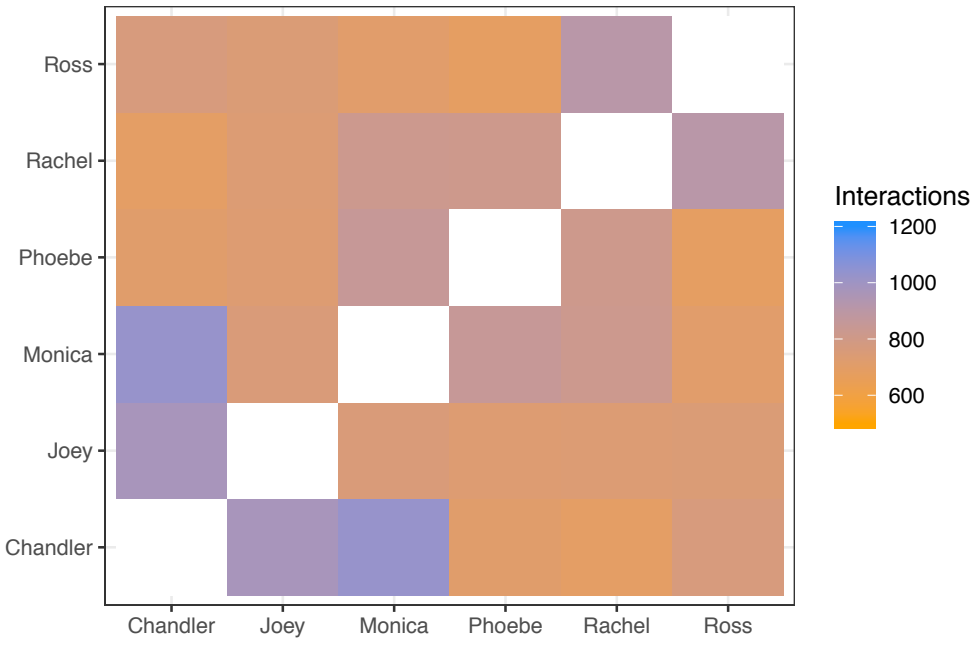
network only observes characters that speak in a scene. Naturally, babies are the centre of attention for looking at and touching, but they rarely speak, so are rarely picked up in the **co-occurrence** networks.

## 5.5 Edge univariate analysis

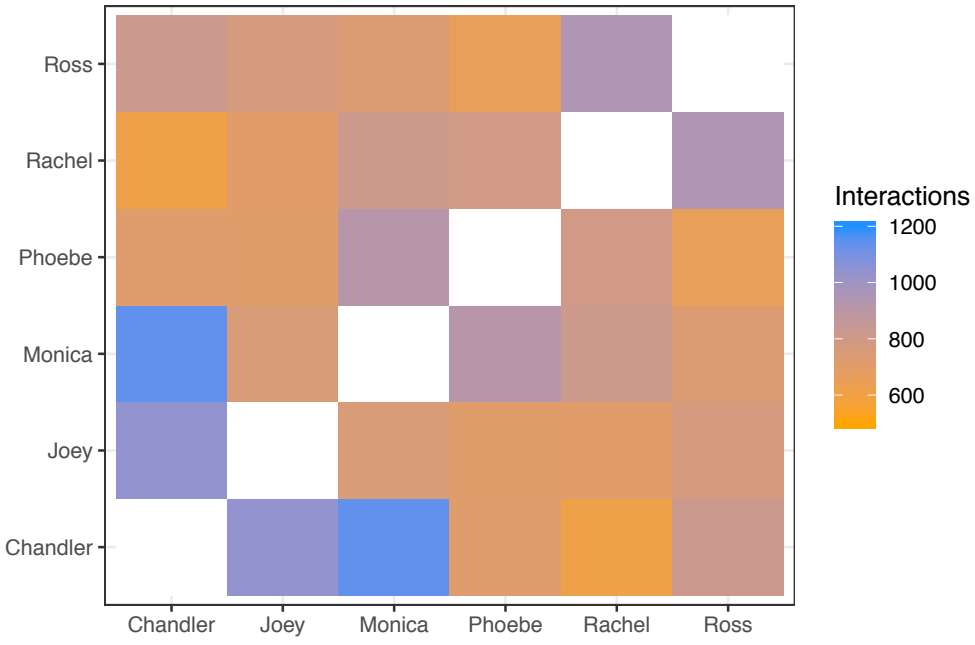
The edge weights of the series network allow us to analyse the prominence of each relationship in *Friends*. We are particularly interested in the relationships between the six core characters as the show revolves around them. Figure 5.16 shows heat maps of the edge weights between the core characters in the **co-occurrence** and **manual** series networks. Note that the heat maps are symmetric because the networks are undirected. Figure 5.16a and Figure 5.16b show that the pair with the most interactions for the series is Monica and Chandler, who are closely followed by Joey and Chandler. The famous intermittent relationship between Ross and Rachel ranks third in both datasets in terms of the number of interactions. The least number of interactions between core characters occur between Chandler and Rachel and between Phoebe and Ross. We also notice that the **manual** dataset has the highest number of interactions in a pair, as well as the least.

Figure 5.17 shows a scatterplot of the edge weights between the core characters for the **co-occurrence** and **manual** series networks. For the pairs of characters with the largest edge weights; Chandler and Monica, Chandler and Joey, Ross and Rachel and Monica and Phoebe, there are more interactions in the **manual** network than the **co-occurrence** network. For the pairs of characters with the smallest edge weights; Chandler and Rachel and Phoebe and Ross, however, there are more interactions in the **co-occurrence** network than the **manual** network. This could be because characters who are closer in relationship tend to talk to, touch or look at each other several times in a scene, which counts as only one interaction in the **co-occurrence**, but several interactions in the **manual** network. On the other hand, characters who are not as close in relationship may be in a scene with each other without talking to, touching or looking at each other.

Interestingly, while Joey and Chandler shared an apartment for as long as Rachel and Monica did, Joey and Chandler shared many more interactions than Rachel and Monica. It is common for narratives to feature males more than females [7, 19, 51, 104]. In fact, Allison Bechdel [22] popularised the term "Bechdel Test" in her comic strip *Dykes to Watch Out For* in 1985, for a test measuring the presence of women in films. A film passes the Bechdel test if it has at least two women in it who talk to each other about something other than a man. A surprising number of films do not pass this test. The




(a) **co-occurrence** dataset




(b) **manual** dataset

Figure 5.16: Heat map of edge weights between core characters (Chandler, Joey, Monica, Phoebe, Rachel and Ross) for the **co-occurrence** series network and the **manual** series network. The edge weight is equivalent to the number of interactions between the characters throughout the series.

**manual** dataset does not contain information about the topic of conversation between characters, but we know that women in *Friends* talk to each other.

| | Co-occurrence | Manual | Equal |
|---|---|---|---|
| Male-Male (%) | 20.83 | 21.66 | 20.00 |
| Female-Female (%) | 20.88 | 20.84 | 20.00 |
| Male-Female (%) | 58.29 | 57.50 | 60.00 |
| Couples (%) | 16.22 | 17.14 | 13.33 |

Table 5.4: Table of percentages of male-male, female-female and male-female interactions between core characters in the co-occurrence and manual static networks of *Friends*, and the percentages if each pair interacted an equal number of times (Equal). Couple (%) is the percentage of interactions between either Chandler and Monica or Rachel and Ross.

Table 5.4 shows, however, that the interactions are distributed evenly amongst males and females in *Friends* in the **co-occurrence** dataset, and only slightly favour males in the **manual** dataset. In both datasets, the majority of interactions between the core characters occur between a male and female character. Compared to the percentage of interactions if all character pairs interacted equally, the males prefer to interact with males, and females with females. This is called homophily, and is analysed in depth in Chapter 6. We also notice that the percentage of male to female interactions is slightly less than it would be if all pairs interacted equally, but the couples (Chandler and Monica, and Rachel and Ross) interact more, so unsurprisingly, they dominate the male to female interactions.

Similarly to the analysis of character metrics, we are interested in the ranking of character relationships over the actual values. Figure 5.18 shows a scatterplot of the rank of the edge weights of each pair of core characters in both the **co-occurrence** and **manual** series networks. The four highest ranked relationships are the same for both datasets, and most other differences in ranks across the datasets are only 1. The exceptions are Chandler and Ross with a difference in rank of 2, and Joey and Rachel and Monica and Ross, both with difference in ranks of 3, however from Figure 5.17, the differences in number of interactions between the datasets are not notably different.

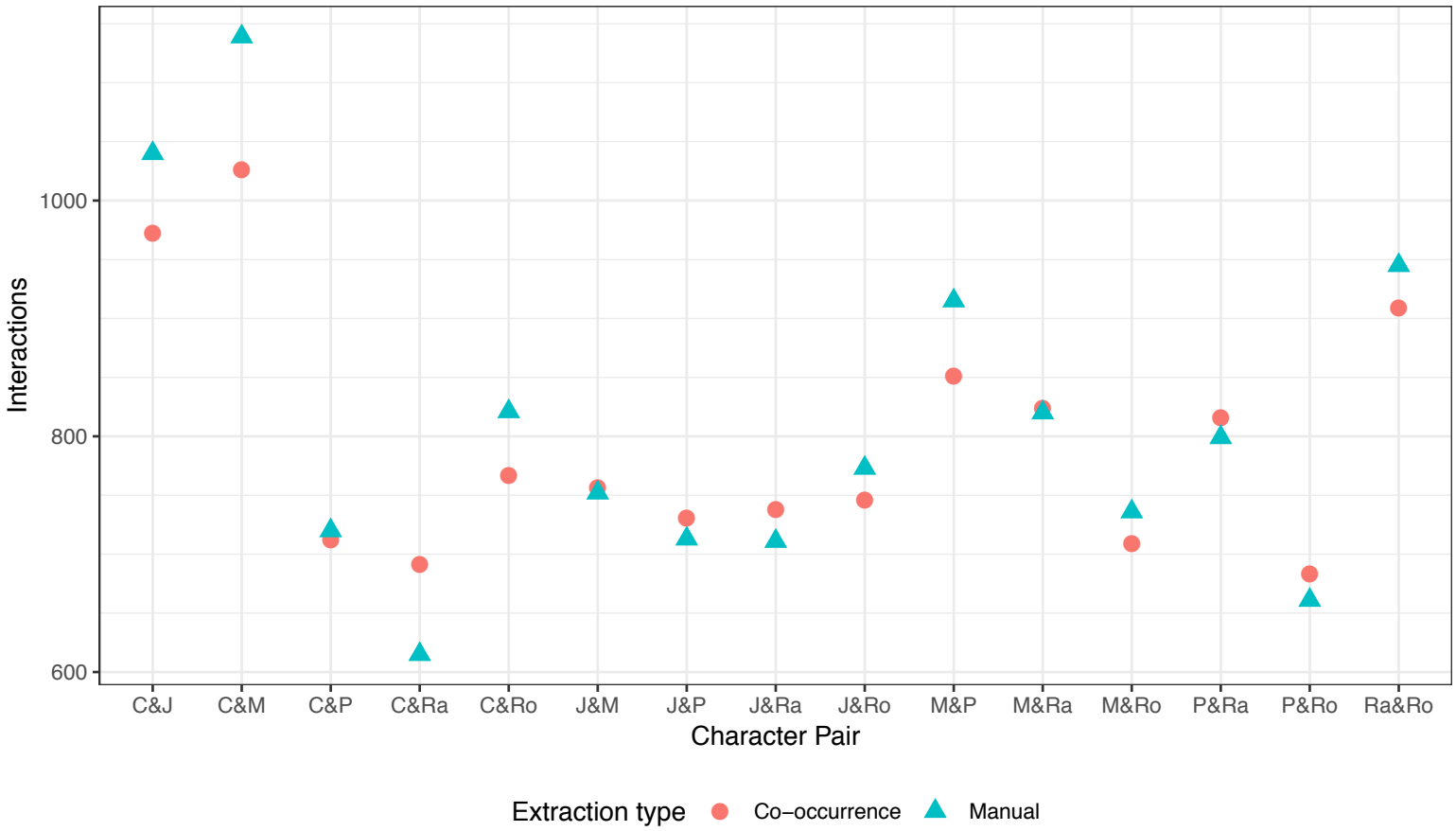



Figure 5.17: Scatterplot of edge weights between core characters (C = Chandler, J = Joey, M = Monica, P = Phoebe, Ra = Rachel and Ro = Ross) for the **co-occurrence** (red) and **manual** (blue) series networks. The edge weight is equivalent to the number of interactions between the characters throughout the series.

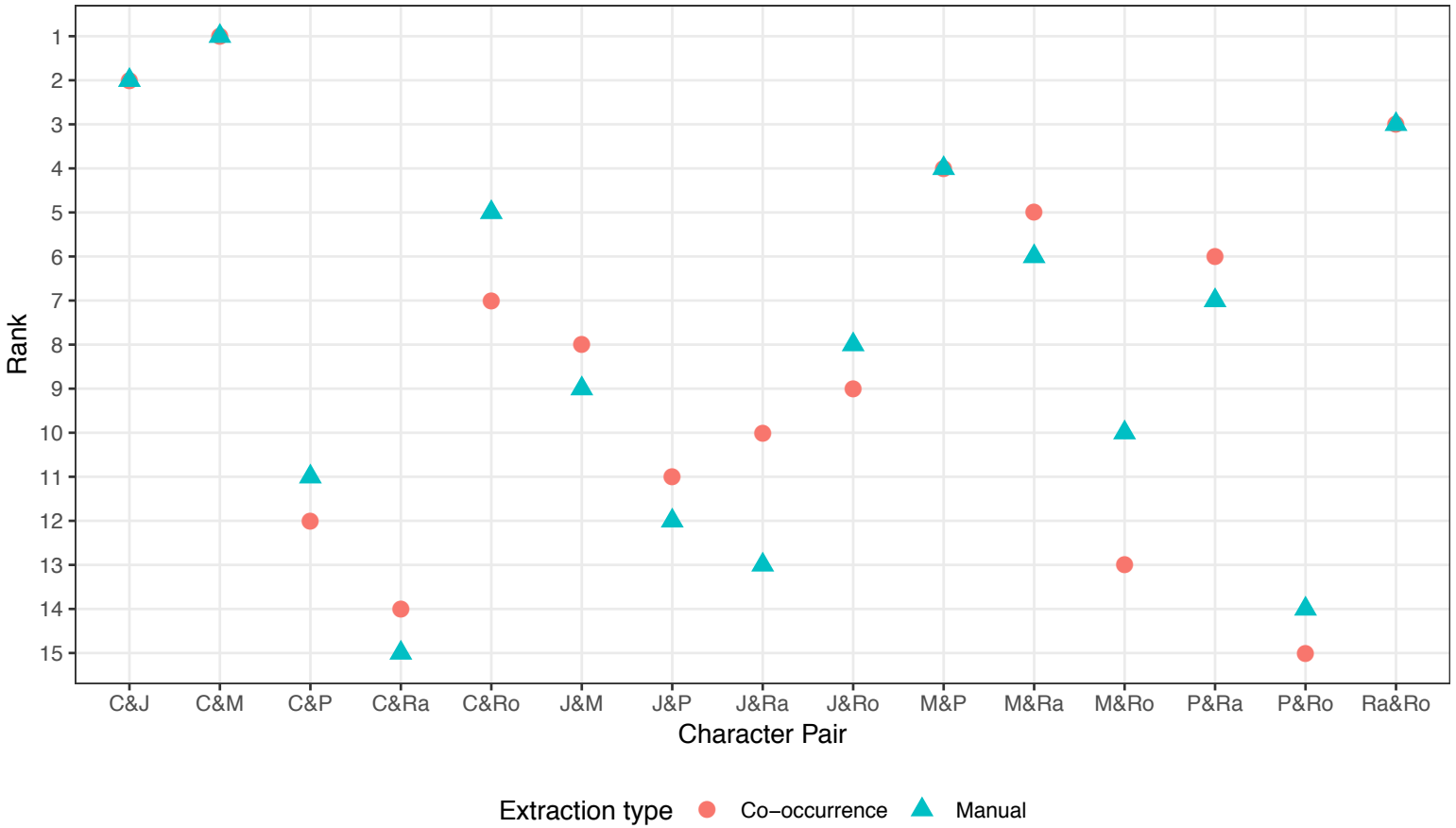



Figure 5.18: Scatterplot of the rank of edge weights between core characters (C = Chandler, J = Joey, M = Monica, P = Phoebe, Ra = Rachel and Ro = Ross) for the **co-occurrence** (red) and **manual** (blue) series networks.

Figure 5.19 shows box plots of edge weights between every core character pair for the **co-occurrence** season networks. The edge weights for the **manual** season networks are in Appendix A.6. Again, Chandler and Joey and Chandler and Monica seem to be the most prominent pairs in the show, closely followed by Rachel and Ross. Monica and Rachel interact 138 times in Season 1, which is the second highest maximum for any pair in a single season, but they don't interact as much in the other seasons.

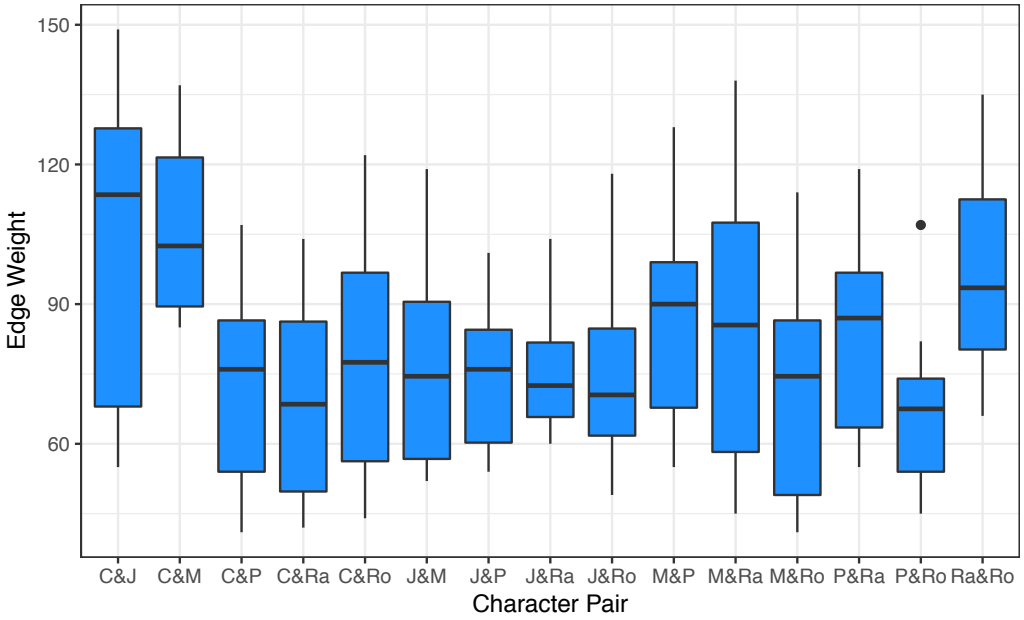



Figure 5.19: Box plots of edge weights between core characters (C = Chandler, J = Joey, M = Monica, P = Phoebe, Ra = Rachel and Ro = Ross) for the **co-occurrence** season networks. The edge weight is equivalent to the number of interactions between the characters in the season.

Figure 5.20 shows box plots of edge weights between every core character pair for the **co-occurrence** episode networks. The edge weights for the **manual** episode networks are in Appendix A.7. The distributions of character pair edge weights across the episodes are relatively similar, but there are many unusually high edge weights for most character pairs, with the exception of Chandler and Phoebe, and Monica and Phoebe. This suggests that there are various episodes that focus on almost all of the character pairs. The highest edge weight for a single episode occurs between Rachel and Ross, which is in line with their intermittent relationship.

We are also interested in relationships between core characters and non-core characters, as well as between non-core characters. Figure 5.21 shows

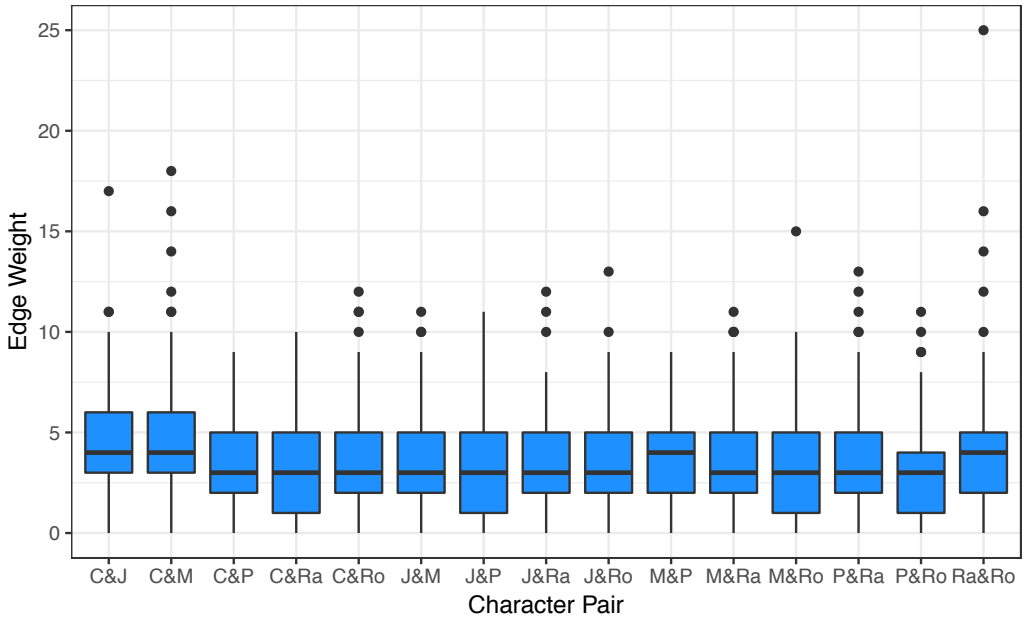



Figure 5.20: Box plots of edge weights between core characters (C = Chandler, J = Joey, M = Monica, P = Phoebe, Ra = Rachel and Ro = Ross) for the **co-occurrence** episode networks. The edge weight is equivalent to the number of interactions between the characters in the episode.

the complementary cumulative distribution of the edge weights on the log scale for the **co-occurrence** and **manual** series networks. Note that the edges between core characters are much larger than the other edges, which motivates our modelling choices of a two-class model in Chapter 6. We find that the edge weights are very right-skew in both datasets as there are many pairs of characters that only interact once or twice, but few pairs that interact more than 10 times. Similarly to the weighted degree distribution, the edge weight distribution appears to have a power law. The characters pairs that do interact more than 10 times often include one core character, with the exception of Jack and Judy (who interact 39 times in the **co-occurrence** dataset and 29 times in the **manual** dataset), and Carol and Susan (who interact 25 times in the **co-occurrence** dataset and 17 times in the **manual** dataset). These relationships are unsurprising as both character pairs are couples that are close to the core group.

The highest edge weights between a character in the core group and a character in the non-core group are between Phoebe and Mike, and Ross and Monica and their parents Jack and Judy. Despite these relationships which are vital to the show, there is a large difference in the edge weights between

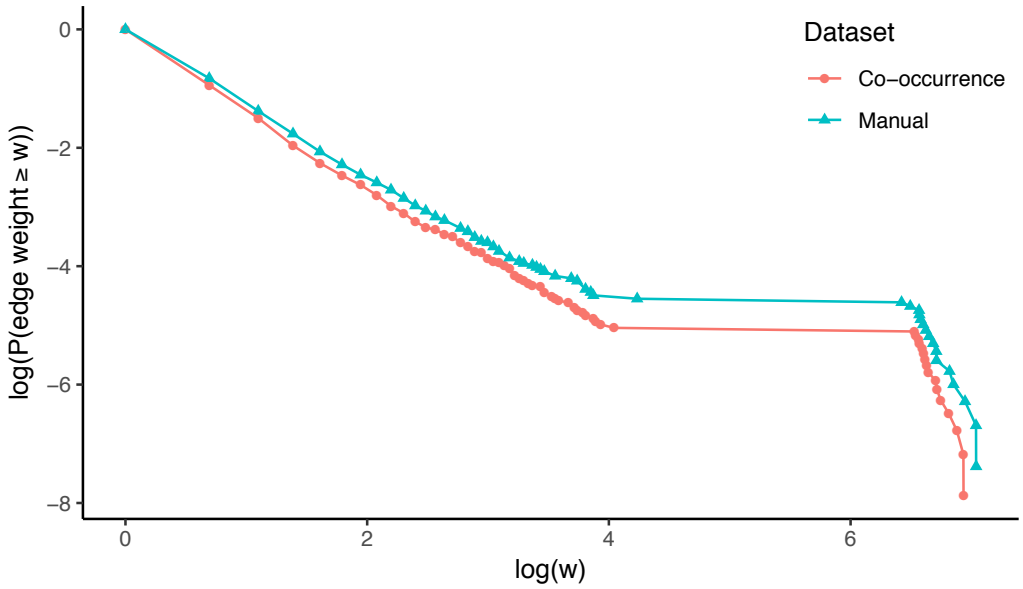



Figure 5.21: Complementary cumulative edge weight distribution of *Friends* character pairs in the series networks for the **co-occurrence** dataset and the **manual** dataset on the log-scale.

these pairs of characters and the edge weight between any pair of characters in the core group. This supports the claim that the core group is the essence of the series.

## 5.6 Global metric bivariate analysis

The season and episode view of the metrics allow us to analyse the metrics as they change over time. Bivariate analysis with episode time windows for the networks is noisy, so we analyse the metrics using season time windows. Figure 5.22 shows scatterplots of the global metrics for the **co-occurrence** and **manual** season networks over time.

The size of the network is similar in both datasets for most seasons, but Season 1 of the **manual** dataset has the most characters. There is a drop in the size of the networks in Season 7 and Season 10. Season 10 only has 18 episodes, compared to the 24 or 25 episodes in every other season, so we expect there will be fewer "extra" characters who only appear once. This could also explain the peak in the density in both datasets in Season 10. The drop in size in Season 7 corresponds to a drop in the total number of edges and a peak in the average edge weight in the **manual** network.

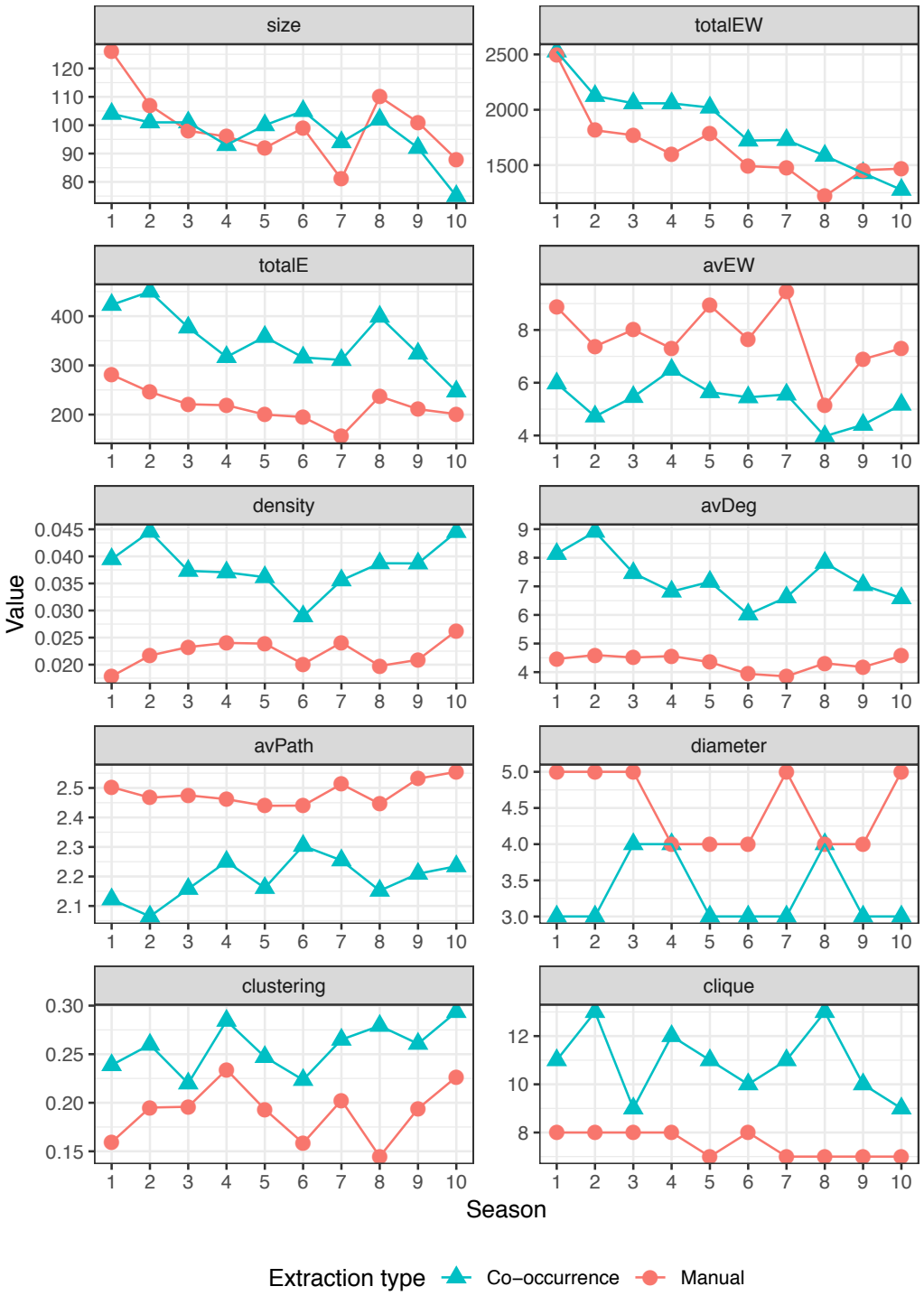



Figure 5.22: Scatterplots of network size, total edge weight (totalEW), number of edges (totalE), average edge weight (avEW), density, average degree (avDeg), average path length (avPath), diameter, clustering coefficient (clustering) and size of the largest clique (clique) for the **co-occurrence** and **manual** season networks over time.

There is also a downward trend in the total edge weight in both datasets. This means characters interact less as the series goes on. This trend is discussed further in Chapter 6.

The average degree varies more in the **co-occurrence** dataset than the **manual** dataset. In the **co-occurrence** networks there is a peak in the average degree in Season 2 and Season 8. These are the same seasons with the highest number of characters in the largest clique in the **co-occurrence** networks. There seems to be a drop in the average degree in Season 6, however, which lines up with a drop in density and clustering, and a peak in average path length.

## 5.7 Character metric bivariate analysis

Analysing character metrics over time allows us to identify changes in character importance and character attributes as the series develops. Figure 5.23 shows the character metrics for the six core characters in the **co-occurrence** season networks over time. Scatterplots for the **manual** season networks over time are in Appendix A.8.

Figure 5.23 shows that the degree, normalised weighted degree and weighted degree are similar for every character in each season, so although there are some changes in character rankings for these metrics throughout the series, many of the differences are not notable. There is a downward trend in the weighted degrees of the characters, similar to the trend in the total edge weights in Figure 5.22. It seems as though the core characters interact less as the series develops. We discuss this trend further in Chapter 6.

The betweenness centrality of the core characters vary widely over the 10 seasons. Joey seems to dominate the betweenness centrality of the networks from Season 5 to Season 10, except for Season 9 where Chandler becomes more important by this metric. This happens because Season 5 is when Joey's acting career kicked off, so he has many interactions with characters from his work from then on. Season 9, however, focusses on Chandler's work as he quits his job in Season 9, Episode 10, then looks for more work in the rest of the season. Chandler interacts with many characters that none of the other core characters interact with through this season, so his betweenness centrality peaks. Therefore betweenness centrality predominantly only tells us about who the core characters interact with when they are not with the other core characters.

Closeness centralities are similar for each character over the ten seasons. We see drops in closeness centralities in Season 2, Season 3 and Season 5. These are the three seasons with more than one component, so we expect

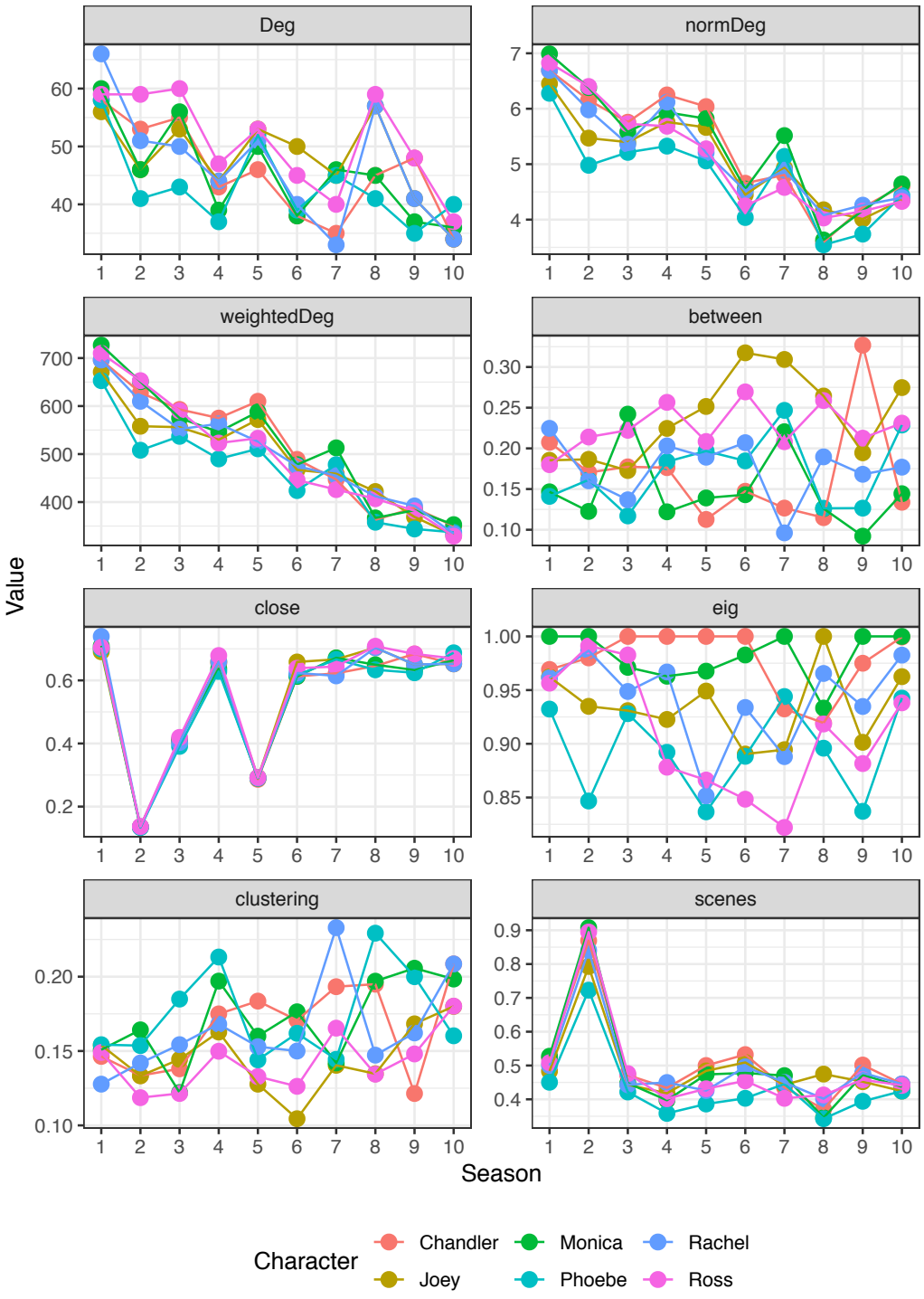



Figure 5.23: Scatterplots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen), local clustering coefficient (clustering) and proportion of scenes (scenes) for the six core characters in the **co-occurrence** season networks over time.

closeness centralities to decrease. A scatterplot of closeness centralities for each character in the largest connected component of the 10 season networks is in Appendix A.4.

Chandler and Monica have the highest eigenvector centralities in every season except for Season 8, where Joey's eigenvector centrality peaks. Monica starts and ends the series as the most important character by this metric, but Chandler dominates the eigenvector centralities in Season 2 to Season 6. In Season 7 we notice a drop in Ross's eigenvector centrality. Ross also has the least interactions and number of scenes in this season, even though his degree is not particularly low. This could be because the season is focussed on Monica and Chandler in their preparation for their wedding (Table 5.1), so there is little time for Ross's character development.

The clustering coefficients of the core characters vary widely over the 10 seasons, but the clustering coefficient peaks and drops correspond to many of the drops and peaks of the betweenness centralities, respectively. For example, Joey's clustering coefficient drops in Season 6, which is the same season he has his highest betweenness centrality. Similarly, Rachel's clustering coefficient peaks in Season 7 and her betweenness centrality drops in Season 7. We see this pattern because characters have high betweenness centralities when they interact with many characters that no-one else interacts with. On the other hand, characters have high clustering coefficients when they interact with characters that do interact with others. For example, if a character only interacts with the core group, all of the character's connections will also interact, leaving the character with a high clustering coefficient. This may not mean the character is particularly important, so clustering tells us little about the importance of characters.

Phoebe is in the smallest proportion of scenes in most seasons of *Friends*. As discussed, Ross is in the smallest proportion of scenes in Season 7, which lines up with when he has the smallest eigenvector centrality, and similarly to eigenvector centrality, Monica is in the most scenes in the season. In Season 8, Joey is in the largest proportion of scenes, which also lines up with his peak in eigenvector centrality. The proportion of scenes each character is in is rarely above half of the scenes in a season.

## 5.8 Edge bivariate analysis

To analyse the edge weights over time, for simplicity we look at Chandler's edge weights with the other core characters over the 10 seasons as Chandler featured in the two highest ranked relationships overall (Figure 5.18). Figure 5.24 shows a scatterplot of the edge weights between Chandler and

the other core characters over time in the **co-occurrence** dataset. Similar scatterplots for the other characters are in Appendix A.9. We notice that Chandler was closest to Joey in the first four seasons, but Monica's relationship with Chandler takes over in the remaining six seasons. With some knowledge of the series, this is unsurprising because Chandler and Joey are roommates at the beginning of the series, and hence they are close friends, but in Season 5, Chandler and Monica get together and eventually get married in Season 7 (Table 5.1). Before they get together, there were no hints of this happening based on the season view edge weights, as the edge weights between Monica and Chandler were not outstanding in Seasons 1 to 4, except having the second highest edge weight with Chandler in Season 4, equal with Rachel.

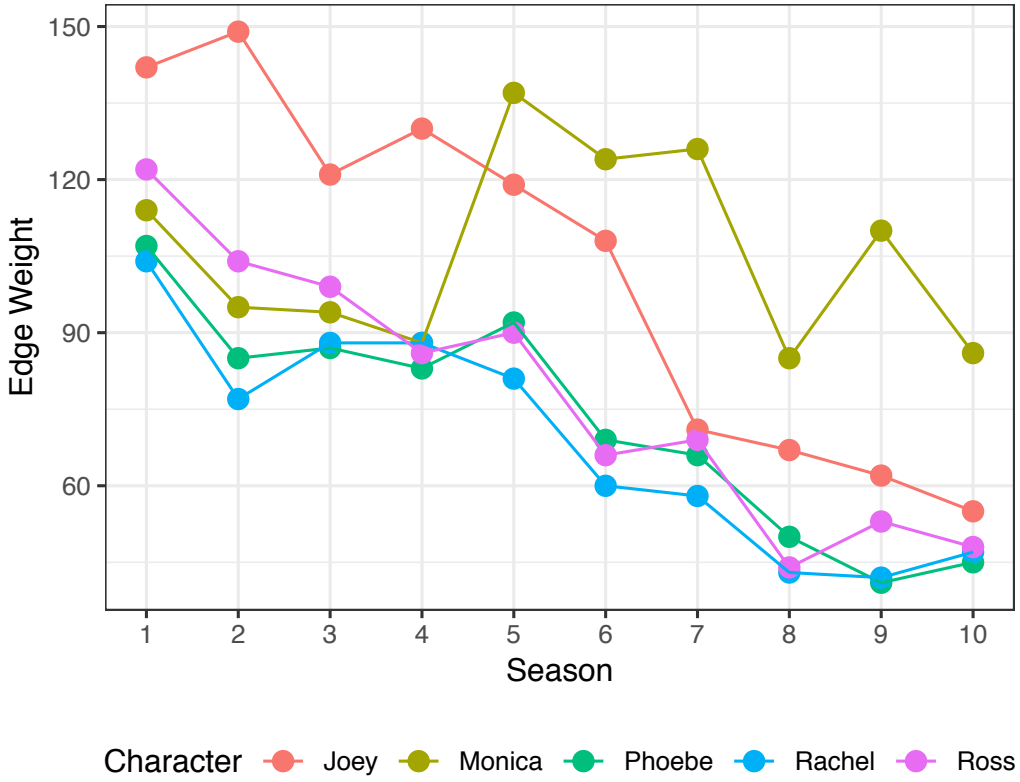



Figure 5.24: Scatterplot of the edge weight of Chandler with the five other core characters in the **co-occurrence** season networks over time.

Chandler was also close to Ross in the first three seasons, but from Season 4 onwards, interactions with Ross were as frequent as interactions with Phoebe and Rachel. Chandler's interactions with Monica, Phoebe and

Rachel in Seasons 1 to 4 and with Ross, Phoebe and Rachel in Seasons 4 to 10 roughly follow the trend of decreasing edge weights in Figure 5.22 and Figure 5.23.

Figure 5.25 shows a scatterplot of the proportion of Chandler's interactions that are with each of the other core characters. While Chandler and Joey interacted many times in the first four seasons, as a proportion of his interactions, Chandler interacts with Monica more in Seasons 5 to 10 than he did with Joey in Seasons 1 to 4.

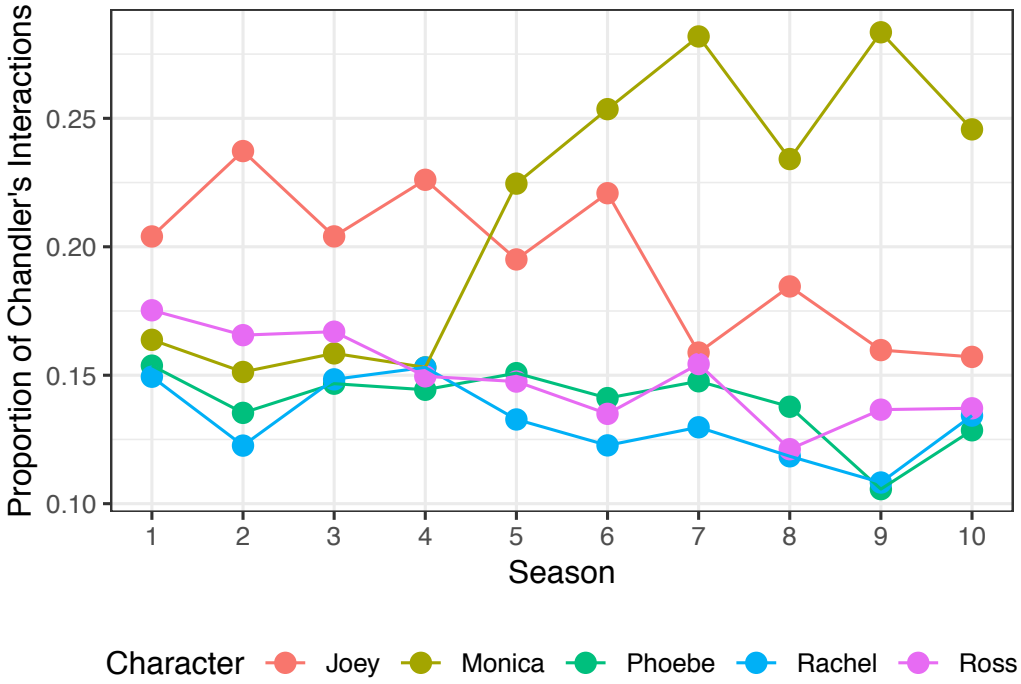



Figure 5.25: Scatterplot of the proportion of Chandler's interactions that are with the five other core characters in the **co-occurrence** season networks over time.

Figure 5.26 shows box plots of the edge weights between Chandler and the other core characters in each episode of every season in the **co-occurrence** network. In Season 1, Chandler interacts with Joey much more than the other characters. In fact, Chandler interacts with Joey at least once in every episode until Season 3.

In Season 4, there are episode edge weights between Chandler and Joey and Chandler and Monica that are unusually high. The high edge weight between Chandler and Monica could hint towards their potential as a couple.

In Season 5, when Chandler and Monica get together, the edge weights between Chandler and Joey remain predominantly between 4 and 6, but Chandler and Monica usually interact more than 5 times. There are two outlying episodes in which Chandler and Monica interact 11 and 14 times.

The highest number of interactions for Chandler with a character in a single episode is 18. This is with Monica in Season 7, Episode 21, *The One with the Vows*. In this episode Chandler and Monica attempt to write their wedding vows, but don't know what to say, so they reflect on all their times together in the past. This episode shows flashbacks of previous interactions between Chandler and Monica, so we would expect to see them interact frequently here.

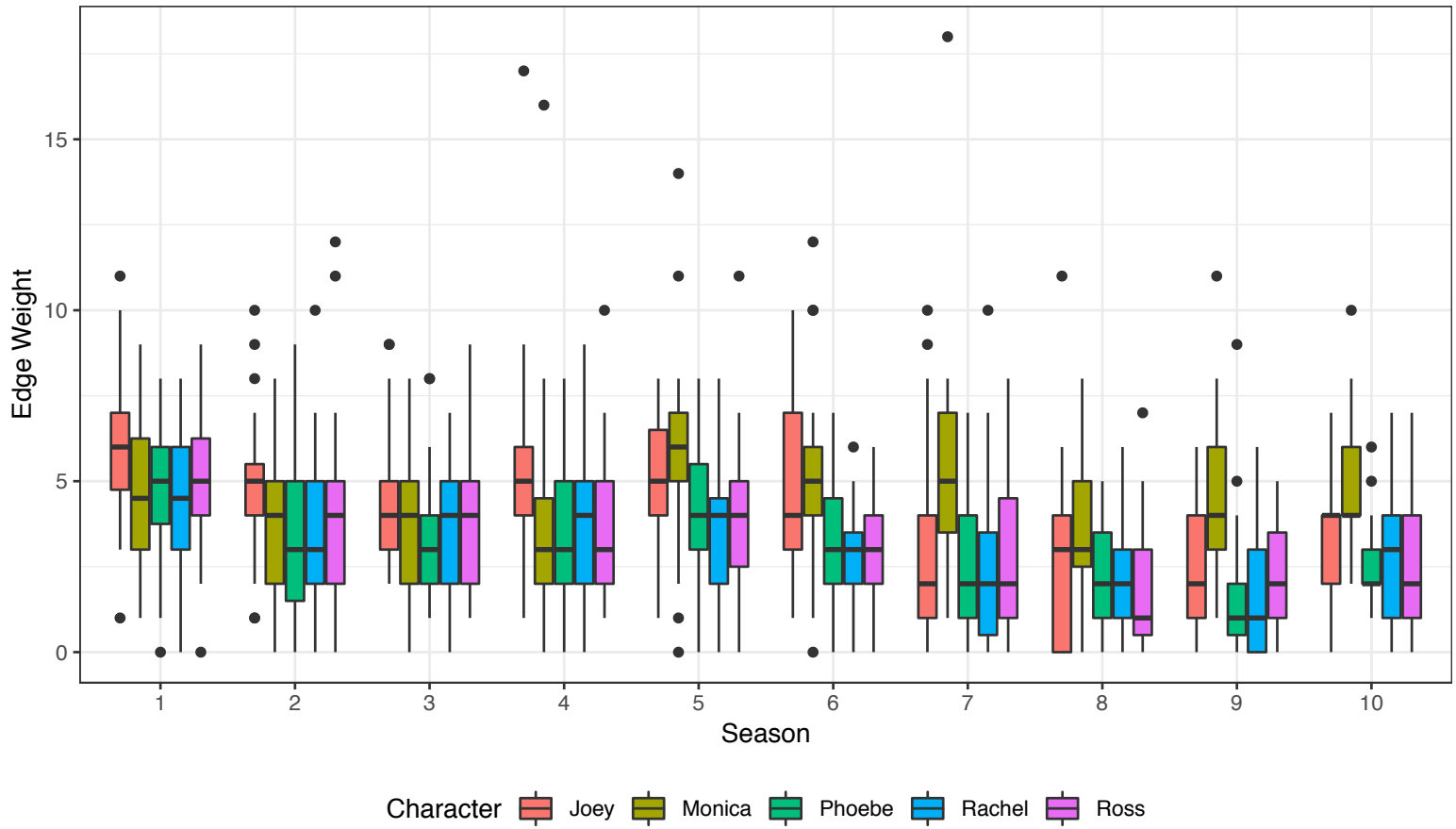



Figure 5.26: Box plots of the edge weights of Chandler with the other five core characters in the **co-occurrence** episode networks for each season.

## 5.9 Summary

Through the analysis of *global*, *character* and *edge* metrics of the **co-occurrence** and **manual** networks, we found that Chandler and Monica's relationship is the most prominent in the series, but it only started in Season 5. Before Season 5, Chandler and Joey's friendship was the strongest relationship. Using the edge weights, the famous intermittent relationship between Rachel and Ross is of less importance than these. This is because the intermittent nature of their relationship means that for much of the series Rachel and Ross are not a couple.

While Chandler's relationships with Monica and Joey have the highest edge weights, there is more evidence that Chandler is the most important character in *Friends*. Chandler frequently has the highest eigenvector centrality, meaning he interacts the most with other important characters. Chandler is also in the highest proportion of scenes throughout the series. Phoebe, on the other hand, is in the smallest proportion of scenes and is the least important character by almost all character metrics. Interestingly, Chandler and Phoebe were originally meant to be secondary characters, providing humour when needed [41], but the producers quickly realised they would be more central to the show. Chandler certainly became central to the show, but Phoebe was closer to remaining as a secondary character. Despite this, Phoebe was certainly more central than any non-core character.

We also noticed that interactions between the core characters in *Friends* are not biased towards either gender. In both datasets the interactions between two males or two females make up slightly more than 20% of the interactions each. The proportion of interactions between males and females is slightly less than 60%, but over a quarter of these interactions are between the famous couples; Monica and Chandler, and Rachel and Ross.

All six core characters are much more important than any of the other characters. This is not surprising as the show is intentionally about these six characters, however it is interesting that none of the more important non-core characters come close to being part of the core group. We can use this information to model the network.

# Chapter 6

# The One With the Model of the Network, its Metrics and Success

"Come on Ross, you're a paleontologist, dig a little deeper."

*Lisa Kudrow* as *Phoebe Buffay*
*Season 6, Episode 23*

## 6.1 Introduction

Analysis of network metrics is useful for understanding certain features of networks, but finding a suitable model for the network can tell us more about its structure and formation. In this chapter, we model the *Friends* networks, and describe the results we can infer from the models. We also model features of the network over time, and features of the network with ratings. These bivariate models allow us to gain insight into, and make predictions about *Friends* and narratives in general.

It is important to note that the *Friends* networks have undirected edges with weights. As we found in Chapter 5, the weights are essential to the network, and so the model should take these into account. Also recall that the networks can represent episodes, seasons or the entire series. The best model for the episode networks may be different to the best model for the season networks. Similarly, we have **manual** and **co-occurrence** networks.

Although many aspects of the networks are similar, they may be fundamentally different, and hence the same model is not necessarily the best model for both datasets. In this chapter, we fit each suggested model to both datasets, and compare the fits.

These are the main contributions of this chapter:

- We demonstrate the suitability of stochastic block models for narrative social network analysis.
- We show that "the *Friends* get less friendly" over the series.
- We show that number of words spoken in each episode increases throughout the series, which is similar to some other television series, but opposite to others.
- We find the network characteristics that predict higher ratings. In particular, high edge density and low clustering coefficient is optimal, which is the combination that appears in "bottle episodes".

## 6.2 Supervised network modelling

### 6.2.1 Initial model

A natural starting point for modelling social networks is a Gilbert-Erdős-Rényi (GER) random network [48, 54]. A GER network has $n$ nodes, and probability $p$ of an edge between pairs of nodes. This means that each edge is a Bernoulli random variable with probability of success $p$. This models a simple, unweighted networks. Both *Friends* network datasets, however, have weighted edges. We saw in Chapter 5 that the weights are essential to the network, so it is important that the model allows for these.

We therefore propose a Poisson model for the graph, where we assume that each edge is an independent observation of the Poisson distribution with rate $\lambda$ interactions per time frame, *i.e.*

$$W_{ij} \sim \text{Poi}(\lambda),$$

for edge weights $W_{ij}$, $i, j = 1, \ldots, n$, $i < j$. The Poisson distribution counts the number of times an event occurs in an interval, so in our context, it counts the number of interactions between a pair of characters in an episode, season or series. Note that this means there could be edges with weight 0. We say an edge with weight 0 is the same as no edge, as it means the characters do not interact in the time frame.

An attribute of the Poisson distribution is that the mean is equal to the variance, so we can use this in determining whether the model is reasonable for the data. To fit the model, we use Maximum Likelihood Estimation (MLE). The likelihood for a particular graph $G = (V, E)$, with nodes

$$V = \{1, \ldots, n\}$$

and edges

$$E = \{w_{ij} \in \mathbb{N} | i, j \in V, i < j\},$$

is

$$L(\lambda; E, n) = \prod_{i,j \in V,\ i<j} \frac{\lambda^{w_{ij}} e^{-\lambda}}{w_{ij}!}.$$

The log-likelihood is then

$$\ell(\lambda; E, n) = \sum_{i,j \in V,\ i<j} w_{ij} \log \lambda - \binom{n}{2} \lambda - \sum_{i,j \in V,\ i<j} \log w_{ij}!.$$

To find the maximum likelihood estimator, $\hat{\lambda}$, we differentiate with respect to $\lambda$ and set the derivative equal to zero. We find

$$\hat{\lambda} = \frac{\sum_{i<j} w_{ij}}{\binom{n}{2}} := \bar{w}.$$

Hence the Poisson parameter estimate is the mean of the edge weights. We use the fact that the mean of a Poisson distribution is the same as the variance to check how appropriate this simple model is for the data. Define the dispersion parameter

$$D = \frac{\sigma^2}{\mu},$$

where $\sigma^2$ is the variance and $\mu = \bar{w}$ is the mean of the edge weights. For a true Poisson distribution, $D = 1$. If $D > 1$, we say the model is overdispersed and if $D < 1$, the model is underdispersed. Figure 6.1 shows a histogram of the calculated dispersion parameters for every episode of the **co-occurrence** and **manual** networks of *Friends*. In both datasets many episodes are very overdispersed, meaning the model doesn't fit the data well. Similarly in the season networks, Figure 6.2 shows that the dispersion parameters in every season of both datasets is much larger than 1.

Recall the large difference between the edge weights between core characters (Rachel, Phoebe, Chandler, Joey, Monica and Ross) and all other edge weights as discussed in Chapter 5. As the model assumes all edge weights come from the same distribution, the difference in these "classes" of edge weights could contribute to the high variance. Therefore we suggest a two-class Poisson model.

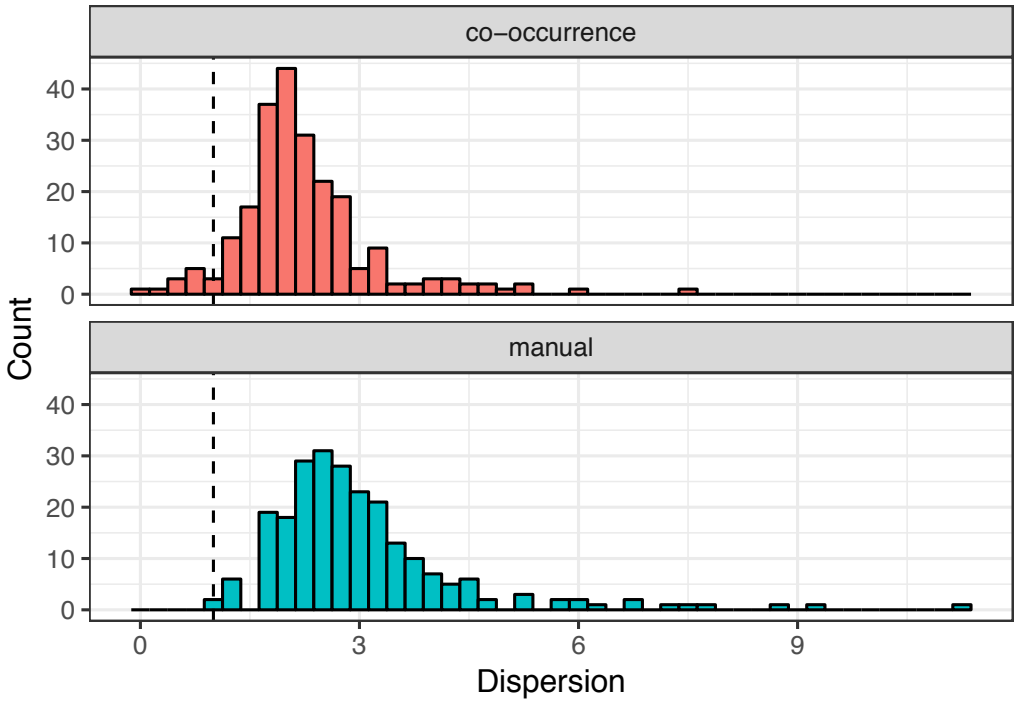



Figure 6.1: Histogram of dispersion for each *episode* in the **co-occurrence** and **manual** episode networks using the simple Poisson model. The dotted line at 1 indicates the dispersion of a true Poisson distribution.

### 6.2.2 Two-class Poisson model

We define the two-class Poisson model as follows. Partition the nodes into two classes; core and non-core, such that

$$V = V_{\text{core}} \cup V_{\text{non-core}}$$

and

$$V_{\text{core}} \cap V_{\text{non-core}} = \emptyset.$$

Here, $V_{\text{core}}$ contains the 6 core characters, and $V_{\text{non-core}}$ contains the remaining $n - 6$ non-core characters. There are now three types of edges;

1. edges between characters within the core group,

2. edges between characters within the non-core group, and

3. edges between characters from each group.

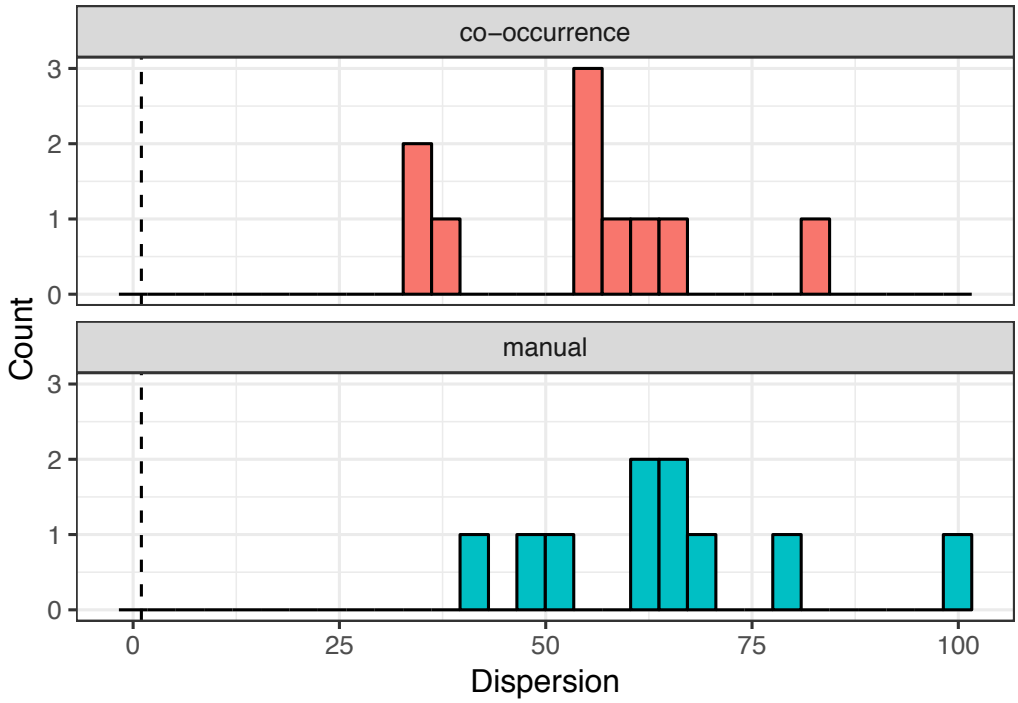



Figure 6.2: Histogram of dispersion for each *season* in the **co-occurrence** and **manual** season networks using the simple Poisson model. The dotted line at 1 indicates the dispersion of a true Poisson distribution.

Let

$$W_{ij} \sim \text{Poi}\left(\lambda_{C_i,C_j}\right),$$

where

$$C_i = \begin{cases} 1 & \text{if } i \in V_{\text{core}}, \\ 0 & \text{if } i \in V_{\text{non-core}}, \end{cases}$$

as in Chapter 4. Recall the networks are undirected, so $\lambda_{C_i,C_j} = \lambda_{C_j,C_i}$. Also define $n_{\text{core}} = |V_{\text{core}}|$ and $n_{\text{non-core}} = |V_{\text{non-core}}|$.

We estimate $\lambda_{C_i,C_j}$ for the **co-occurrence** and **manual** network edge

weights at both the episode and season view using MLE and find:

$$\hat{\lambda}_{C_i,C_j} = \begin{cases} \dfrac{\sum_{i<j} w_{ij} C_i C_j}{\sum_{i<j} C_i C_j} & \text{if } C_i = C_j = 1, \\ \dfrac{\sum_{i<j} w_{ij}(1-C_i)(1-C_j)}{\sum_{i<j}(1-C_i)(1-C_j)} & \text{if } C_i = C_j = 0, \\ \dfrac{\sum_{i<j} w_{ij}(C_i + C_j - 2C_iC_j)}{\sum_{i<j}(C_i + C_j - 2C_iC_j)} & \text{otherwise.} \end{cases}$$

$$= \begin{cases} \dfrac{\sum_{i<j} w_{ij}}{\binom{n_{\text{core}}}{2}} & \text{if } i,j \in V_{\text{core}}, \\ \dfrac{\sum_{i<j} w_{ij}}{\binom{n_{\text{non-core}}}{2}} & \text{if } i,j \in V_{\text{non-core}}, \\ \dfrac{\sum_{i<j} w_{ij}}{n_{\text{core}} n_{\text{non-core}}} & \text{otherwise.} \end{cases}$$

This means the maximum likelihood estimator for the Poisson parameter of an edge type is the average of the edges of that edge type.

As in the one-class model, we calculate the dispersion for each parameter in each episode and season to check the model is reasonable. Figure 6.3 shows histograms of the dispersion for each of the edge types in the co-occurrence and manual episode networks. In some episodes there are no non-core characters, so we cannot estimate $\lambda_{0,0}$ or $\lambda_{0,1}$. In other episodes there is only one non-core character, so we cannot estimate $\lambda_{0,0}$. In these circumstances we ignore the dispersion parameters that we cannot calculate.

Most of the dispersion parameters in the two-class model are much closer to 1 than in the one-class model, giving some evidence that the two-class model is better. Figure 6.4 shows box plots of the dispersion parameters for each edge type of each season of the **co-occurrence** and **manual** networks. While these are closer to 1 than in the one-class model, the model is still overdispersed for the season networks.

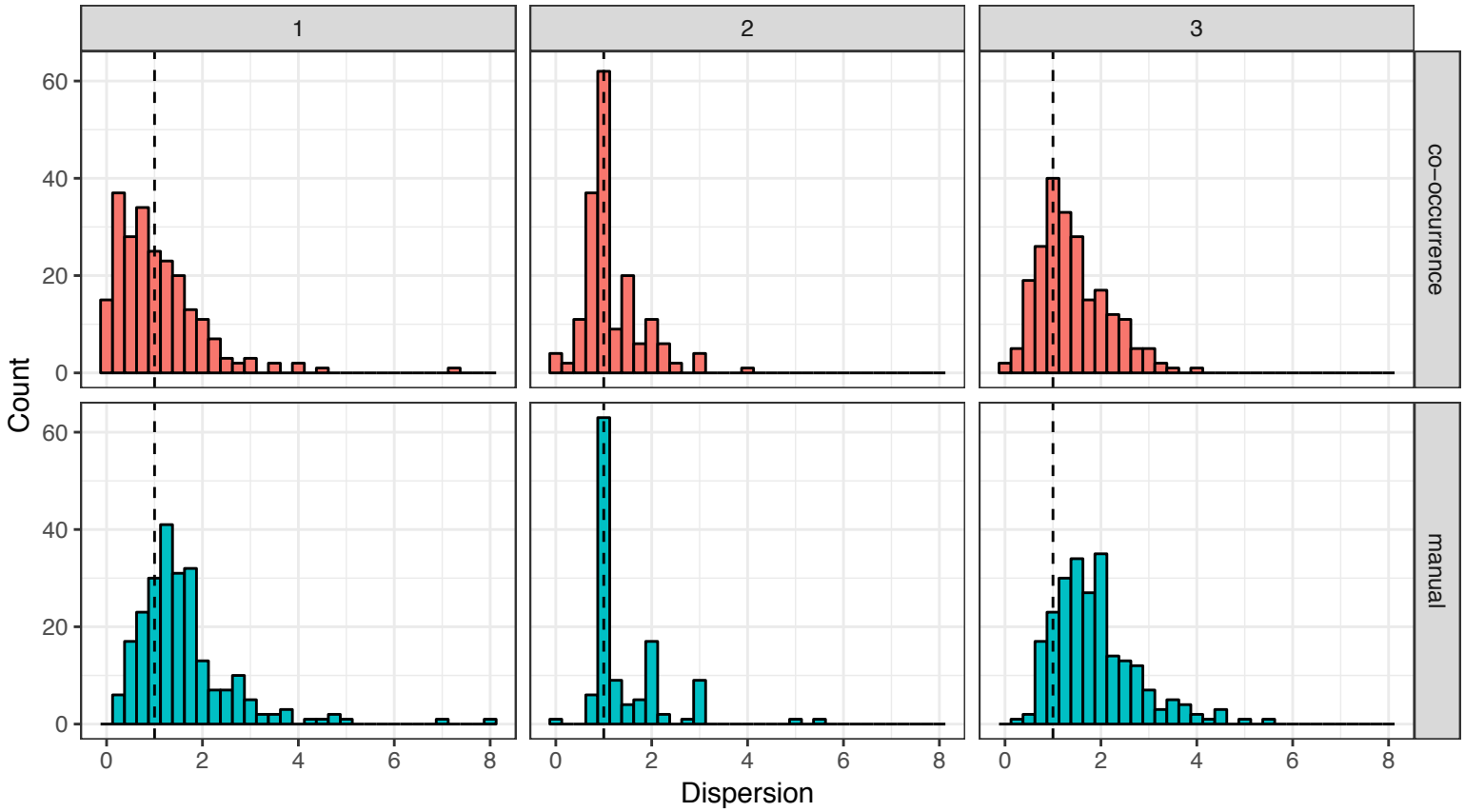



Figure 6.3: Histograms of dispersion for each *episode* using the two-class Poisson model for the **co-occurrence** and **manual** datasets. The dispersion has been calculated separately for each type of edge: core group to core group (1), non-core group to non-core group (2), and between groups (3). The dotted line at 1 indicates the dispersion of a true Poisson distribution.

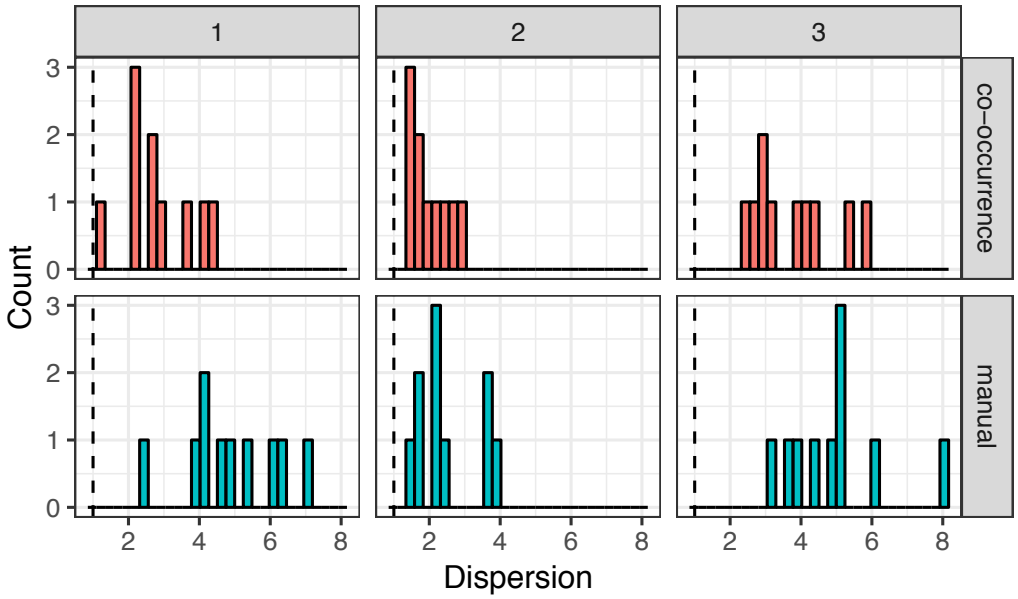



Figure 6.4: Box plots of dispersion for each *season* using the two-class Poisson model for the **co-occurrence** (red) and **manual** (blue) datasets. The dispersion has been calculated separately for each type of edge: core group to core group (1), non-core group to non-core group (2), and between groups (3). The dotted line at 1 indicates the dispersion of a true Poisson distribution.

### Model selection

We now formally compare models using Akaike's information criterion (AIC) [15, 16]. The AIC for a model is

$$AIC_{\text{model}} = 2k_{\text{model}} - 2l(\hat{\lambda})_{\text{model}}$$

where $k_{\text{model}}$ is the number of parameters estimated in the model, and $l(\hat{\lambda})_{\text{model}}$ is the log-likelihood of the model at its maximum. A smaller AIC roughly means the model fits the data better, without overcomplicating it with too many parameters. To favour the simpler model, we use the rule of thumb that if Model 2 has more parameters estimated than Model 1 (*i.e.* $k_2 > k_1$), then we choose Model 2 over Model 1 if

$$AIC_2 < AIC_1 - 2.$$

Figure 6.5 is a cumulative density function of the difference in AIC scores for each episode in the co-occurrence and manual datasets. The dotted line is

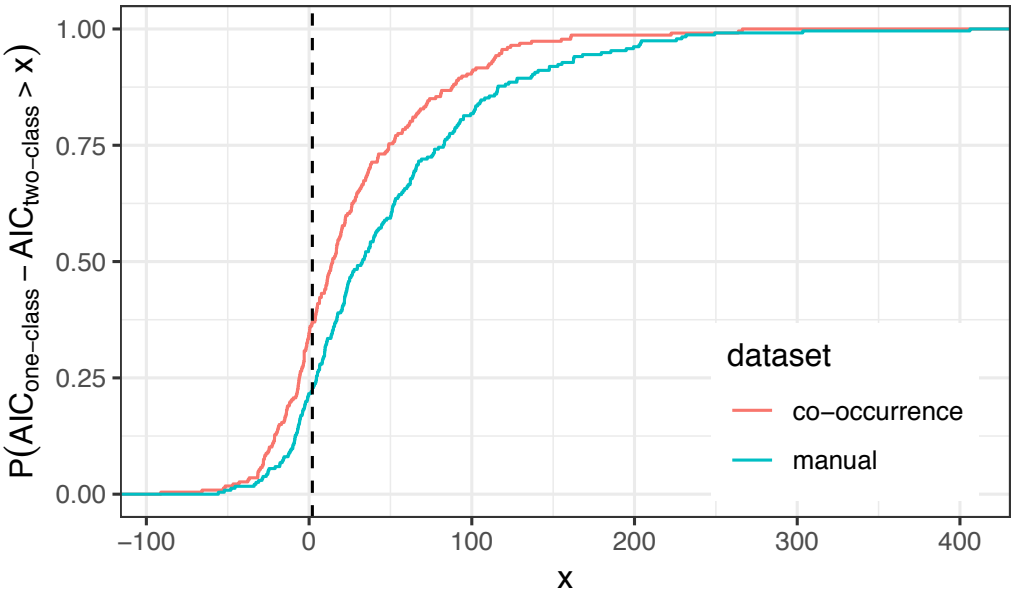



Figure 6.5: Cumulative distribution of the difference in AICs for the one-class and two-class model for the episode networks in the **co-occurrence** (red) and **manual** (blue) datasets. The dotted line represents is at $x = 2$, which is the lower bound threshold for accepting the two-class model over the one-class model.

the AIC threshold (where the difference is 2) for accepting the more complicated model. The more complicated model is the two-class model, which has 3 parameters, as opposed to 1 in the one-class model. In the **co-occurrence** dataset, we would select the two-class model for 63.4% of episodes, and in the **manual** dataset, we would select the two-class model for 76.7% of episodes. Therefore, in general, the two-class model is worth the extra parameters, particularly in the **manual** dataset.

One could argue that the sample sizes are small and so the AIC could overfit [39]. To address this issue we also calculate the corrected AIC (AICc) [33]. The AICc of a model further penalises for extra parameters, and is given by

$$AICc_{\text{model}} = AIC_{\text{model}} + \frac{2k^2_{\text{model}} + 2k_{\text{model}}}{n - k_{\text{model}} - 1}.$$

Using the AICc over the AIC, we would select the two-class model for 63.0% of the co-occurrence episodes and 76.3% of the manual networks.

Therefore, for both datasets, the majority of networks are better modelled by the two-class Poisson model than the one-class model. In the season

networks, the difference in AICs in the two-class model and the one-class model are much greater than 2 for both datasets, so the two-class model is much better here too.

#### Verification against simulations

To check whether the model fits the data well, we simulate networks using out model and compare the global metrics of the simulated networks to those of the observed networks. Figure 6.6 shows histograms of metrics for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 1, Episode 9: *The One Where Underdog Gets Away*. We chose this episode as it has the median number of characters, six core and five non-core. The dotted black line represents the metric of the actual network for this episode.

Notice that the total number of edges, density and average degree are the same, but on different scales. This is because we simulate with a fixed number of nodes. Most of the simulated network metrics are centred around the value of the real network, however the clustering coefficient and clique size are underestimated by the model. We obtain similar results with simulations based on other **co-occurrence** episode networks (see Appendix A.10).

We also simulate 1000 networks using our model based on **co-occurrence** season networks. Figure 6.7 shows histograms of metrics for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 1. See Appendix A.10 for histograms of metrics for simulations based on the **co-occurrence** network for Season 2.

At the season view, the model overestimates the number of edges (and hence density and average degree), and underestimates the average edge weight, average path length, clustering coefficient and number of characters in the largest clique. The total edge weight and diameter are simulated well by the model.

The model fits the data at the episode view better than at the season view, so we could simply use it for episode networks and aggregate these to form a season network. However, the model overestimates the clustering and size of the largest clique even at the episode view, so it may be beneficial to consider other models.

### 6.2.3 Other possible models

While the two-class Poisson model does not fit every aspect of the *Friends* episode data well, the need to split the characters into core and non-core

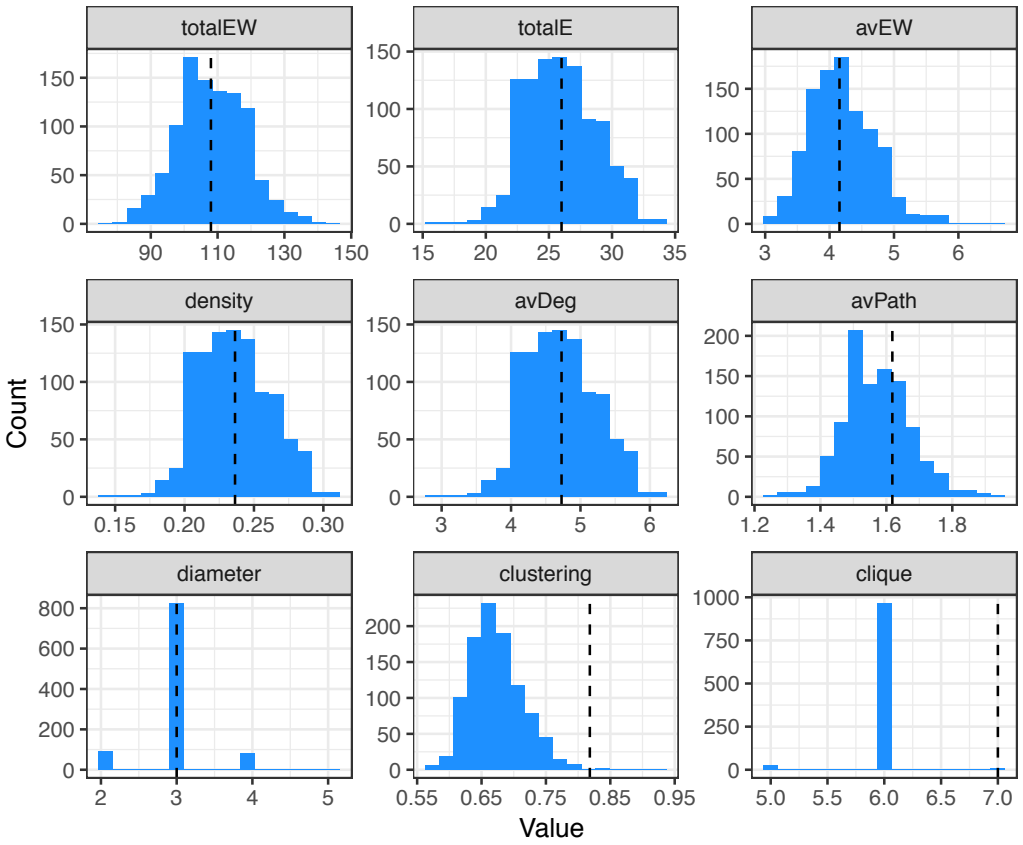



Figure 6.6: Histograms of metrics (total edge weight: totalEW, total number of edges: totalE, average edge weight: avEW, density, average degree: avDeg, average path length: avPath, diameter, clustering coefficient: clustering, and size of the largest clique: clique) for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 1, Episode 9: *The One Where Underdog Gets Away*. The dotted black line represents the metric of the actual network for episode.

groups still applies. Here we discuss different network models and how one could apply them to the *Friends* networks.

### Preferential attachment model

The preferential attachment model, or Barabási-Albert model generates scale-free networks, where there are few nodes with unusually high degrees, and many nodes with low degrees [20]. A unique properly of the preferential attachment model is that the resulting degree distribution follows a power law.

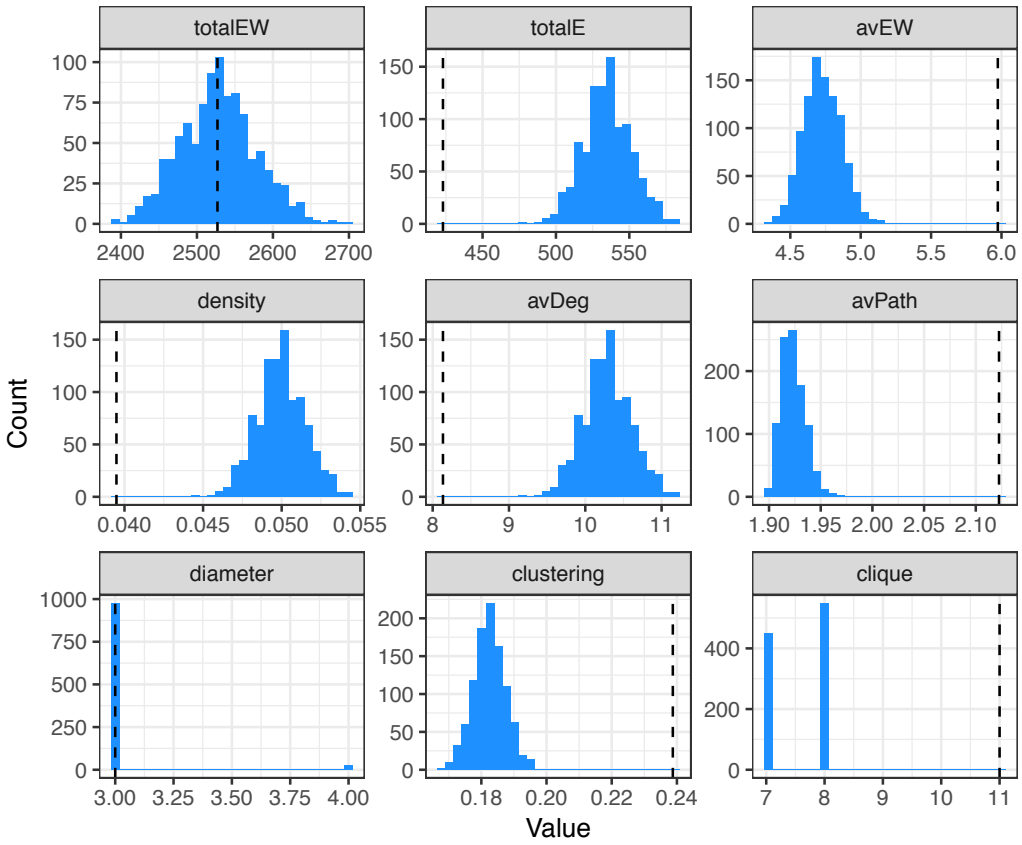



Figure 6.7: Histograms of metrics (total edge weight: totalEW, total number of edges: totalE, average edge weight: avEW, density, average degree: avDeg, average path length: avPath, diameter, clustering coefficient: clustering, and size of the largest clique: clique) for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 1. The dotted black line represents the metric of the actual network for episode.

One indication of a power law is a linear pattern in the cumulative degree distribution on the log-log scale.

In Chapter 5 we noticed that the the cumulative weighted degree distributions of the non-core characters in the global networks appeared linear on the log-log scale. As the network is weighted and has the two-class structure, the standard preferential attachment model is not suitable. However, an adapted preferential attachment model may be suitable, such as the multiclass preferential attachment model suggested by Shakkottai *et al.* [109] to model the evolution of the internet.

### Small-world model

The small-word model, or Watts-Strogatz model generates networks with "small-world" properties such as short average path lengths and high clustering [124]. The two-class Poisson model did not capture the high clustering of the episode or season networks, so a small-world model may be more suitable for this aspect of the model. An attribute of the small-world model, however, is degree regularity, *i.e.*, the degree is similar across all nodes. We saw in Chapter 5 that this is not evident in the *Friends* networks, so the small-world model would have to be adapted to fit these networks.

### More classes

An alternative to changing the basic model assumptions to be more like the preferential attachment or small-world model is to change the number of classes. More classes could allow for clustering within classes, and hence would increase the clustering in the model simulations. It is not clear, however, what the third class would represent, and which characters would be in it. This motivates the application of a stochastic block model to these datasets.

## 6.3 Unsupervised block model

In the two-class model we used information about the characters in the core group to split the network into the two classes. However, the model might be more suitable if we have more classes, for example, a three-class model with the core characters, recurring non-core characters and non-recurring non-core characters as the three classes. This raises the question of how many classes is best, and how can we determine which characters are in each class? Given an adequate method for determining each character's class, we could apply the model to social networks of narratives where the core characters are not as intrinsic to the narrative as Chandler, Joey, Monica, Phoebe, Ross and Rachel are to *Friends*.

There are various methods for community detection in networks, such as multilievel [28], spinglass [98], leading eigenvector [90], label propagation [96], infomap [103], walktrap [93] and edge betweenness [92]. Bazzan compared these methods of community detection on the *Friends* **manual** networks [21]. Here, however, we are not interested in finding communities (where there are more edges within communities than between communities). We wish to find classes, or hierarchies, where the parameters that describe how characters interact depend on which class the characters are in, but it may

be that characters in the same class are less likely to interact with each other than with characters outside of their class. Therefore we use a stochastic block model.

### 6.3.1 Stochastic block models

A stochastic block model assumes each node belongs to a class, and interactions between characters depend on the class in which the character is in. We observe the interactions, and must infer the class structure. Here we define the stochastic block model as described by Mariadassou *et al.* [78].

#### Model

Let $G = (V, E)$ be a network with nodes $V = \{1, \dots, n\}$ and $E = \{W_{ij} \in \mathbb{G} | i, j \in V\}$. For the weighted interaction networks, $\mathbb{G} = \mathbb{N}$. Our networks are undirected, so $W_{ij} = W_{ji}$, and hence we only define edges for $i < j$.

Assume each node is a member of a class $q = 1, \dots, Q$. We estimate the number of classes, $Q$ using a process described later in this section.

Also define the $n \times q$ membership matrix $Z$ by

$$Z_{iq} = \begin{cases} 1 & \text{if node } i \text{ is in class } q \\ 0 & \text{otherwise.} \end{cases}$$

The latent layer of the model is the class membership, for which we assume

$$\boldsymbol{Z}_i \sim \mathcal{M}(1, \boldsymbol{\alpha}),$$

independently, where $\mathcal{M}$ is the multinomial distribution and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_Q) \in \mathbb{R}_+$ is the probability vector for class membership, *i.e.*

$$P(Z_{iq} = 1) = \alpha_q$$

and

$$\sum_{q=1}^{Q} \alpha_q = 1.$$

The observed layer of the model is the set of edges, $W_{ij}$ for $i, j \in V$. We assume

$$W_{ij} | (Z_{iq} Z_{j\ell} = 1) \sim \mathcal{F}_{q\ell},$$

where $\mathcal{F}_{q\ell}$ is a distribution that depends on the classes $q$ and $\ell$. For an unweighted network we use the Bernoulli distribution:

$$\mathcal{F}_{q\ell} = \mathcal{B}(\pi_{q\ell}) = \begin{cases} 1 & \text{with probability } \pi_{q\ell} \\ 0 & \text{with probability } 1 - \pi_{q\ell}. \end{cases}$$

The *Friends* networks are weighted, so we use the Poisson distribution:

$$\mathcal{F}_{q\ell} = \text{Poi}(\lambda_{q\ell}).$$

### Estimation

We use Leger's R-package `Blockmodels` [64] for estimation, which implements Mariadassou *et al.*'s method [78]. Here we describe the methods used in the the package.

Define the parameter $\boldsymbol{\theta} := (\boldsymbol{\alpha}, \boldsymbol{\lambda})$ where

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1Q} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{Q1} & \lambda_{Q2} & \dots & \lambda_{QQ} \end{bmatrix}.$$

Given the number of classes $Q$, we estimate $\boldsymbol{\theta}$ by maximising the likelihood of the observed data,

$$\mathcal{L}(E; \boldsymbol{Z}, \boldsymbol{\theta}) = P(E|\boldsymbol{Z}, \boldsymbol{\theta}).$$

Now,

$$\begin{aligned} P(E|\boldsymbol{Z}, \boldsymbol{\theta}) &= \sum_{\boldsymbol{Z}} P(\boldsymbol{Z}, E|\boldsymbol{\theta}) \\ &= \sum_{\boldsymbol{Z}} P(\boldsymbol{Z}|\boldsymbol{\theta}) P(E|\boldsymbol{Z}, \boldsymbol{\theta}) \end{aligned}$$

But

$$P(\boldsymbol{Z}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{q=1}^{Q} \alpha_q^{Z_{iq}},$$

and

$$P(E|\boldsymbol{Z}, \boldsymbol{\theta}) = \prod_{i<j} \prod_{q,\ell} f_{q\ell}\left(W_{ij}\right)^{Z_{iq} Z_{j\ell}},$$

where

$$f_{q\ell}\left(W_{ij}\right) = \frac{\lambda_{q\ell}^{W_{ij}} e^{-\lambda_{q\ell}}}{W_{ij}!}$$

for the Poisson edge weights.

So

$$P(E|\boldsymbol{Z}, \boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} \left( \prod_{i=1}^{n} \prod_{q=1}^{Q} \alpha_q^{Z_{iq}} \right) \left( \prod_{i<j} \prod_{q,\ell} f_{q\ell}\left(W_{ij}\right)^{Z_{iq} Z_{j\ell}} \right).$$

This summation involves $Q^n$ terms, so quickly becomes intractable. Therefore we use the Expectation-Maximisation (EM) algorithm [45]. The EM algorithm iterates two steps;

1. adjusting the class memberships $\boldsymbol{Z}$ based on parameters $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\lambda})$ (expectation step),

2. estimating parameter $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\lambda})$ given the edges $E$ and classes $\boldsymbol{Z}$ (maximisation step).

To initialise the class membership of the nodes, we use the Absolute Eigenvalues Spectral Clustering method, as in Rohe *et al.* [100]. In the expectation step we calculate

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right) = \mathbb{E}_{\boldsymbol{Z}|E,\boldsymbol{\theta}^{(t)}}\left[\log \mathcal{L}(\boldsymbol{\theta}|E, \boldsymbol{Z})\right],$$

for iteration $t$, and in the maximisation step we calculate

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right).$$

However, $\mathcal{L}(\boldsymbol{\theta}|E, \boldsymbol{Z})$ (and hence $Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right)$) depends on $P(\boldsymbol{Z}|E)$, which is intractable due to the network's strong dependency between edges. Instead, we use variational EM to approximate the likelihood, as outlined by Mariadarrou *et al.* [78].

The variational EM algorithm outputs the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ for a model with $Q$ classes, which we denote as $m_Q$. In practice, we also wish to estimate $Q$. We use the Integrated Classification Likelihood (ICL), as proposed by Biernacki *et al.* [25];

$$\mathrm{ICL}_{m_Q} = \max_{\boldsymbol{\theta}} \log(P(E, \tilde{\boldsymbol{Z}}|\boldsymbol{\theta}, m_Q)) - \mathrm{pen}(m_Q),$$

where the penalty term $\mathrm{pen}(m_Q) = \frac{1}{2}\left(P_Q \log(n(n-1)) - (Q-1)\log n\right)$. Here $\tilde{\boldsymbol{Z}}$ is the prediction for $\boldsymbol{Z}$, and $P_Q$ is the number of parameters estimated by the model, which is $\frac{Q(Q+1)}{2}$ for the undirected Poisson model. The best model is the one with the highest ICL.

We run the model for $Q = 1, 2, 3$ and 4. If the model with the maximum ICL has $Q_{\max}$ classes, we also run the model up to $Q = 1.5Q_{\max}$. Out of these models, we select the number of classes that achieves the highest ICL.

### 6.3.2 Results

We fit the stochastic block model to every episode, season and series network in the **co-occurrence** and **manual** dataset using the `Blockmodels` package

in R. The algorithm classifies the core characters in their own class in every season and series network, but in some episode networks core characters are split up and combined with non-core characters.

### Series-level model

Figure 6.8 shows the series **co-occurence** network with the nodes coloured by classes. The **co-occurrence** series network has 9 classes with 6, 427, 30, 37, 2, 9, 55, 11 and 81 nodes in the classes. Figure 6.9 shows a heatmap of the $\boldsymbol{\lambda}$ parameters for the 9 classes.

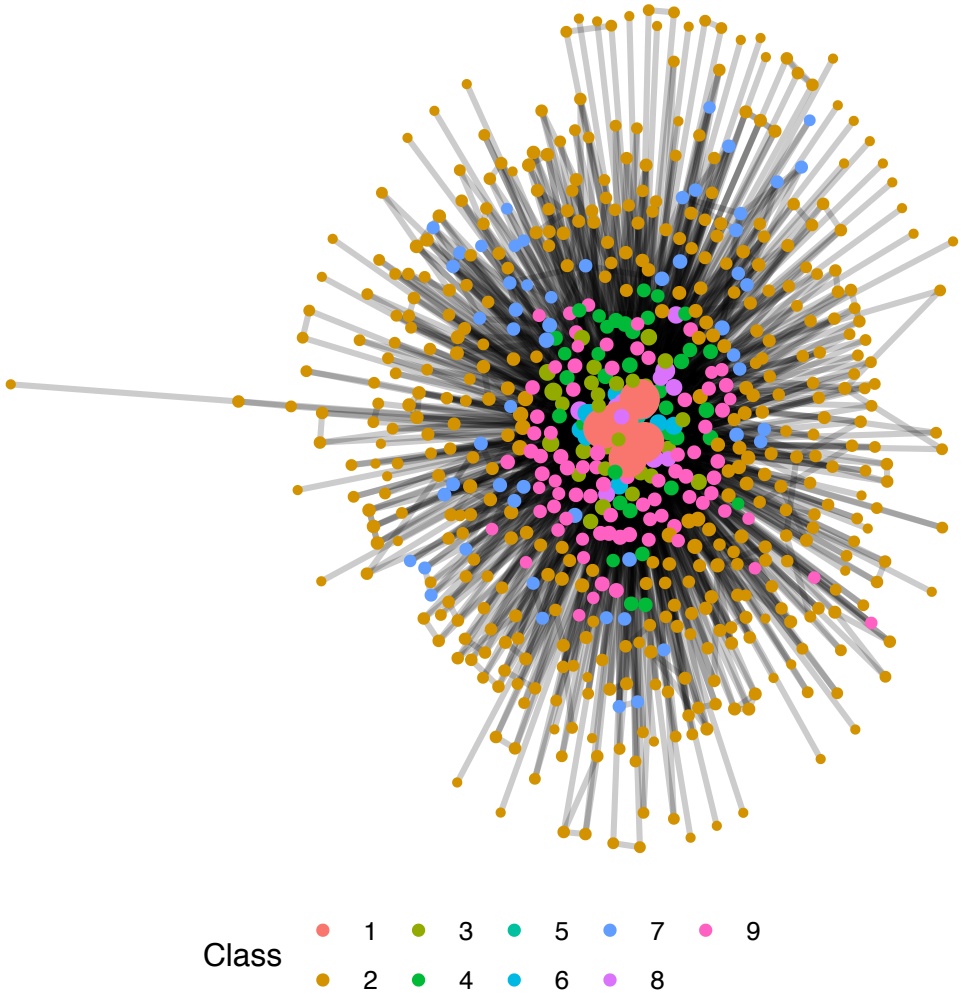



Figure 6.8: **Co-occurrence** series network with nodes coloured by classes determined by stochastic block model. There are 9 classes in total.

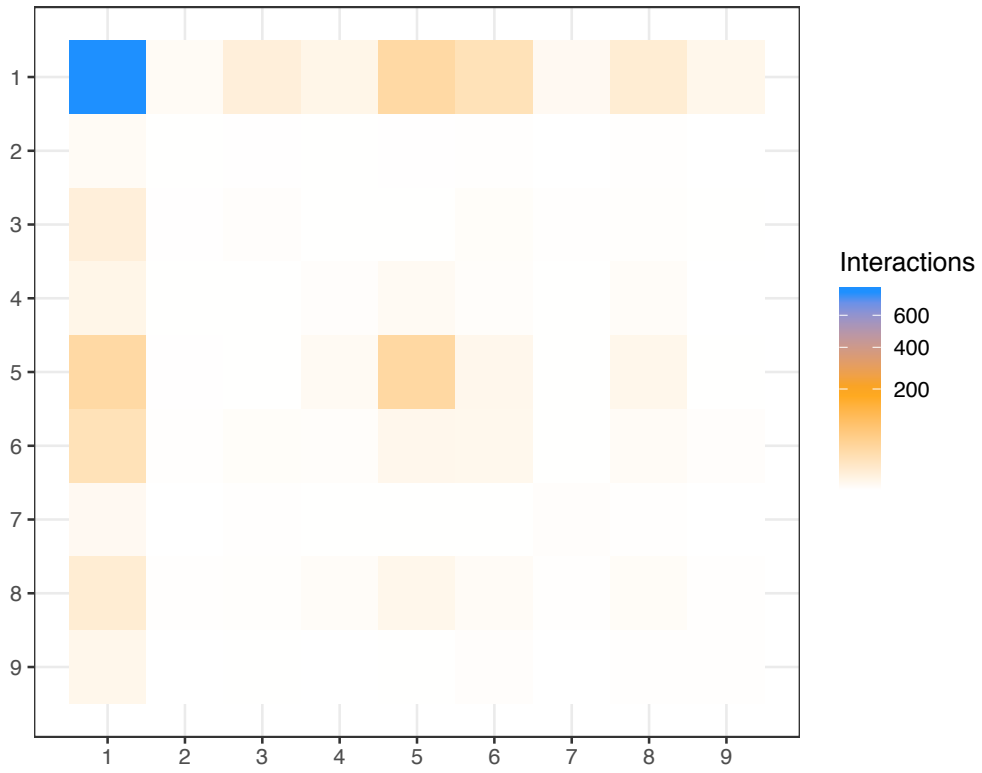



Figure 6.9: Heatmap of the **co-occurrence** series stochastic block model $\boldsymbol{\lambda}$ parameters. The legend shows that the darker shades correspond to larger parameters (*i.e.*, larger mean number of interactions).

Class 1 contains the six core characters, so $\lambda_{11}$ is larger than any other parameters by far. The dark row and column for Class 1 show that most characters mainly interact with the core characters. We also see high interaction rate within Class 5. Class 5 contains only Mr. Geller and Mrs. Geller, so it is not surprising that they frequently interact with each other and with the core group.

Characters in Class 6 also interact frequently with the core characters. Class 6 contains Charlie, Emily, Mike, Richard, Gunther, "woman", Janice, Susan and Carol. These characters are those who have long term relationships with the core characters, as well as characters that are regularly around the core characters: Gunther, "woman" and Carol. Class 3 contains some of the characters that had a shorter relationship with the core characters, such as Tag, Joshua, Kathy, Paolo and David. Class 8 contains characters

that are also close to the core six characters, however they interact within their own class less. This class contains the frequently occurring non-specific characters such as "waiter", "man", "guy" and "nurse".

The remaining classes – Classes 2, 4, 7 and 9 – contain the characters with smaller roles in the series. The four classes differ by how much the characters in them interact with the core characters and other important characters, such as Mr. and Mrs. Geller. Characters in Classes 4 and 9 interact the most with core characters, and characters in Class 2 interact the least with core characters.

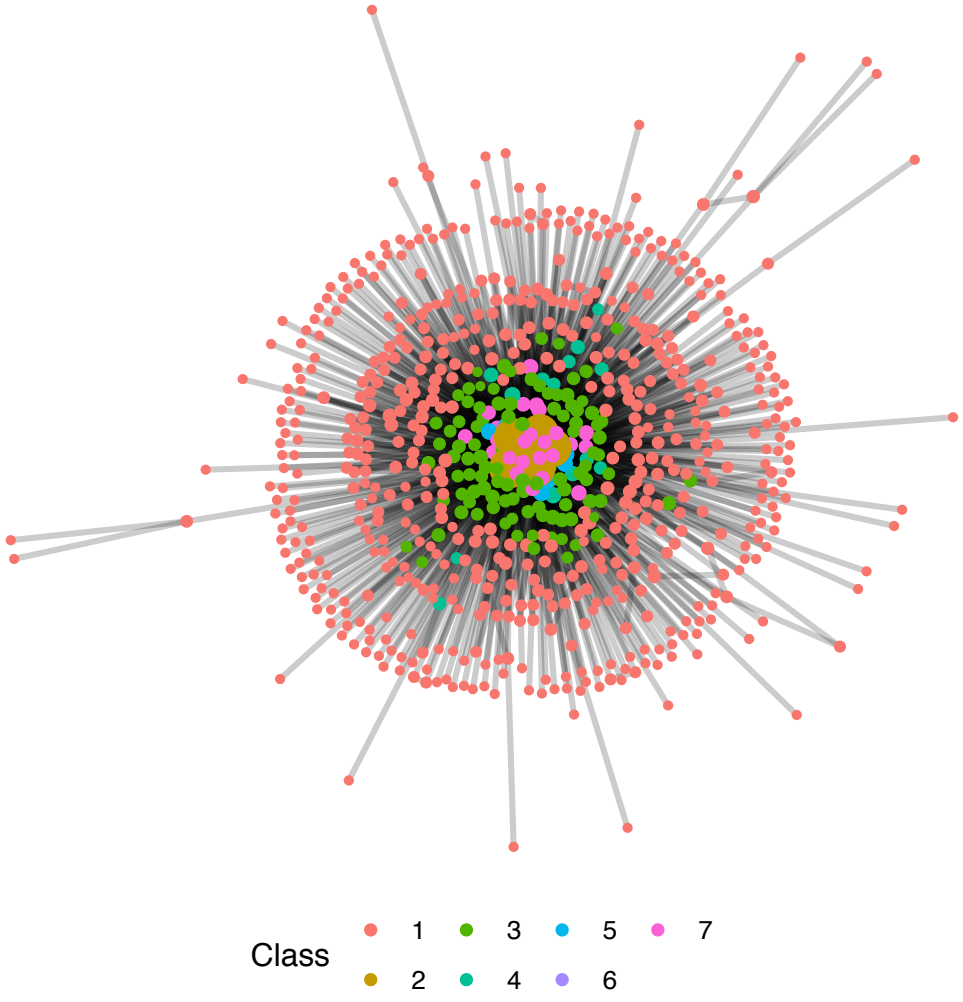



Figure 6.10: **Manual** series network with nodes coloured by classes determined by stochastic block model. There are 7 total classes.

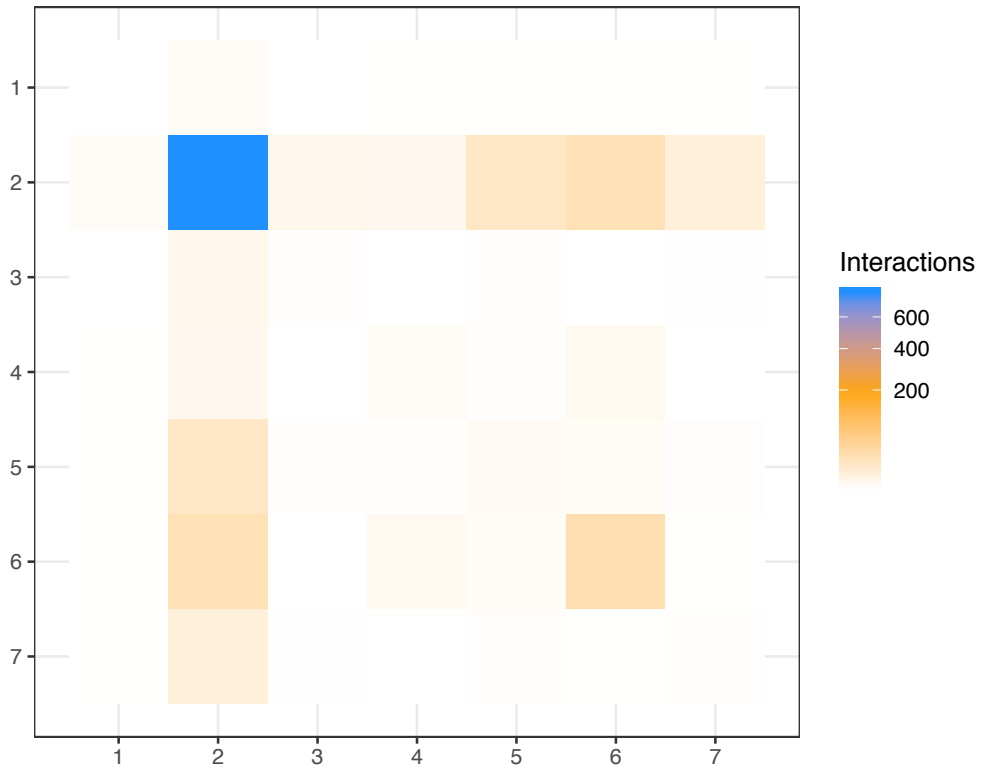



Figure 6.11: Heatmap of the **manual** series stochastic block model $\boldsymbol{\lambda}$ parameters. The legend shows that the darker shades correspond to larger parameters (*i.e.*, larger mean number of interactions).

Figure 6.10 shows the series **manual** network with the nodes coloured by classes. The **manual** series network has 7 classes with 585, 6, 102, 14, 11, 2 and 26 nodes. Figure 6.11 shows a heatmap of the $\boldsymbol{\lambda}$ parameters for the 7 classes. Note that the labelling of the classes is arbitrary, so we compare the classes in each of the datasets.

As in the **co-occurrence** network, there is a class containing exactly the six core characters, who interact frequently with each other and characters in other classes often interact with them. In the Figure 6.11 the core group is labelled by Class 2. There is also a class containing only Ross and Monica's parents Jack and Judy (Class 6), as in Class 5 in the **co-occurrence** network.

Class 5 in the **manual** network is most similar to Class 6 in the **co-occurrence** network, as it contains characters with close relationships to the core characters; Mike, Emma, Emily, Richard, Gunther, FrankJr, Ben,

Charlie, Janice, Susan and Carol. Class 7 in the **manual** network is similar to Class 3 in the **co-occurrence** network, as it contains recurring characters that had a smaller role than those in Classes 2, 5 and 6.

Class 4 contains characters that interact with each other almost as much as they interact with the core group. For example, Barry (Rachel's ex-fiancé) and Mindy (his new wife) interact with each other more than any core characters.

The remaining characters are in Classes 1 and 3. The difference between these classes is how much they interact with the core group. Characters in Class 3 interact with the core group more than characters in Class 1.

Table 6.1 shows a contingency table of the class membership of the characters with the same name in both the **manual** and **co-occurrence** datasets. This table shows how the character classifications overlap between the two datasets. For most classes in the **manual** series network, the majority of characters in the class are in the same class in the **co-occurrence** network. For example, Class 1 in the **manual** network (Man1) has 98 of the 132 possible commonly named characters in Class 2 in the **co-occurrence** network (Co2). The **manual** Class 6 is also exactly the same as the **co-occurrence** Class 5, although the Table 6.1 doesn't show this as Ross and Monica's parents have different names in the different datasets.

| | Man1 | Man2 | Man3 | Man4 | Man5 | Man6 | Man7 | total |
|---|---|---|---|---|---|---|---|---|
| Co1 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 6 |
| Co2 | 98 | 0 | 12 | 0 | 1 | 0 | 1 | 112 |
| Co3 | 1 | 0 | 6 | 0 | 1 | 0 | 14 | 22 |
| Co4 | 4 | 0 | 9 | 2 | 0 | 0 | 1 | 16 |
| Co5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Co6 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 8 |
| Co7 | 9 | 0 | 9 | 4 | 0 | 0 | 0 | 22 |
| Co8 | 1 | 0 | 2 | 0 | 0 | 0 | 2 | 5 |
| Co9 | 19 | 0 | 20 | 0 | 0 | 0 | 1 | 40 |
| total | 132 | 6 | 58 | 6 | 10 | 0 | 19 | 231 |

Table 6.1: Contingency table of the 231 characters that are in the **manual** and **co-occurrence** dataset, and which class they are in using the stochastic block model on each series network. The rows indicate the 9 classes of the **co-occurrence** series network, and the columns indicate the 7 classes of the **manual** series network.

However, there are some exceptions to the similar classes across the datasets. For instance, Class 8 in the **co-occurrence** network only has

five characters with names in both datasets, but these characters are split into Classes 1, 3 and 7 in the **manual** network.

### Season-level models

Figure 6.12 shows the **co-occurrence** Season 1 network as classified into classes by the stochastic block model, along with a heatmap that represents the interaction rate within and between classes. The **co-occurrence** networks for Seasons 2 to 10 are in Appendix A.11. Similar figures for the **manual** season networks are in Appendix A.12. In every season of both datasets there is a class containing the six core characters. In the **co-occurrence** dataset, there are 4 classes in every season, except for Seasons 5 and 6, which have 3 classes, and Season 7, which has 5 classes.

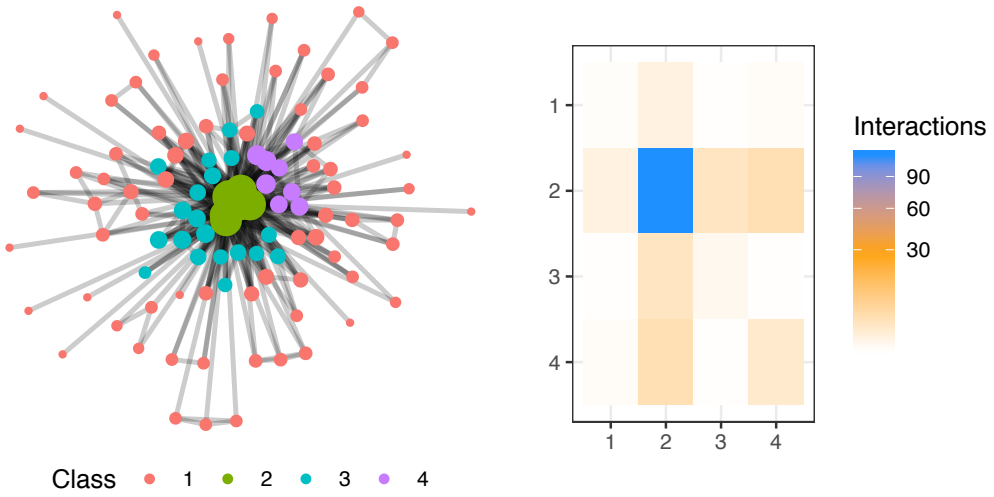



Figure 6.12: Season 1 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

In Season 1, Susan and Carol are alone in a Class 1, and Class 3 and Class 4 split the remaining characters into those who interact with the core group frequently (Class 4), which form an "inner non-core group", and those with less interaction with anyone (Class 3), forming an "outer non-core group". We see similar class structures in all the seasons, with the core group, an inner non-core group and an outer non-core group. Figure 6.13 shows the parameters representing the expected number of interactions with these three groups and the core group over time.

Characters in the core group interact with each other more than with characters from any other group, but we consistently see two non-core groups

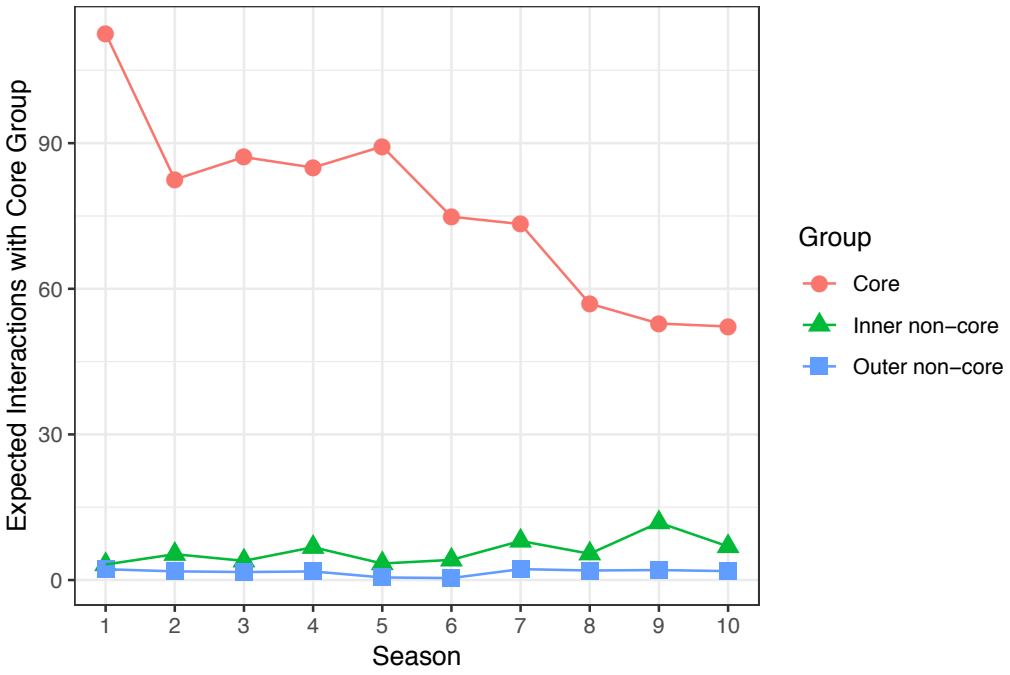



Figure 6.13: Parameters representing the expected number of interactions between characters in the core group, and other characters in the core group (red), and characters in the inner non-core (green) and outer non-core (blue) groups.

that also have some contact with the core group. The parameters of the interactions with these groups and the core group vary from season to season, but even more noticeably, the characters in these groups vary.

In Season 2, the class containing Susan and Carol expands to also include Richard, Mrs. Geller, Mr. Geller, Julie and Mrs. Green – characters that are very close to the core group. The remaining classes are split similarly to in Season 1, with an inner non-core group and an outer non-core group.

In Season 3, the class with a very close relationship to the core characters reflects the important side characters in this season. Interestingly, Class 4 contains Lauren, the Director, Kate and "woman". This is because these characters appear together in some scenes, so not only are there interactions between them and the core characters, but also interactions within the class. Class 3 contains the outer non-core group.

Season 4 is similar to Season 1 and 2, as it has a class very close to the core group (Class 2), and an inner non-core group (Class 4) and an outer non-core group (Class 3). Season 5 and 6, however, only have three classes;

the core group, an inner non-core group and an outer non-core group. The inner core groups are different in Seasons 5 and 6, however Mrs. Geller and Gunther are in this group in both seasons.

Season 7 has five classes. In Class 1, there are three characters who are very close to the core group in this season: Mrs. Geller, Mr. Geller and Tag. Class 5 is the second closest to the core characters, and contains characters such as Mrs. Bing, Mr. Bing and Gunther. Class 4 contains a group of characters from Joey's acting career, who interact with each other as much as with the core characters. The remaining characters are in Class 3, which is similar to the outer non-core group.

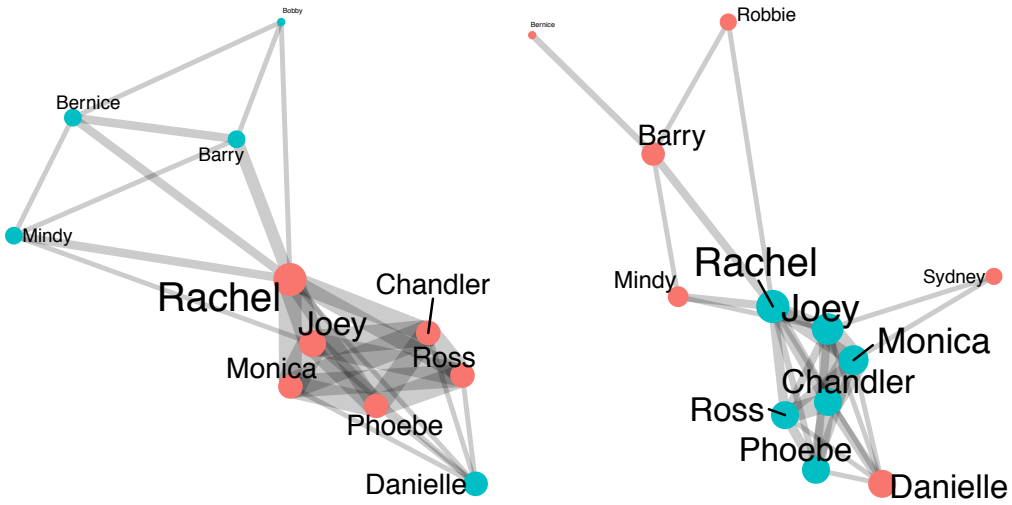
In Season 8, surprisingly, the class that is closest to the core group contains three characters: Mona, "man" and "woman". The inner non-core group is in Class 4, and Class 1 contains the outer non-core group. In Season 9, however, the closest class to the core group only contains two characters: Mike and Charlie. These characters are in long-term relationships with core characters, so it is not surprising they are in this class. Again, the remaining characters are split into an inner non-core group (Class 4) and an outer non-core group (Class 3).

Finally, in Season 10, the class close to the core group contains Mike, Charlie and Emma, so it doesn't change much from Season 9.

### Episode-level model

At the episode view, the core characters are not always in the same class, and there is more variability in the class classification across the two datasets. For some episodes, the stochastic block model classifies the characters into the same or similar classes in the **co-occurrence** and **manual** networks. For example, Figure 6.14 shows the **co-occurrence** and **manual** networks for Season 1, Episode 20: *The One With the Evil Orthodontist*, with nodes coloured by class as determined by the stochastic block model. For both datasets, the characters in this episode network are split into core and non-core classes, as we would split the network in the two-class Poisson model. Note that the class labelling and hence node colour is arbitrary.

In contrast, for some episodes the stochastic block model classifies the characters into very different classes in the **co-occurrence** and **manual** networks. For example, Figure 6.15 shows the **co-occurrence** and **manual** networks for Season 1, Episode 8: *The One Where Nana Dies Twice*, with nodes coloured by class as determined by the stochastic block model. In the **co-occurrence** network, there are only two classes: one containing the core characters and Mrs. Geller, and the other containing the non-core characters. In the **manual** network however, there are four classes. The core characters



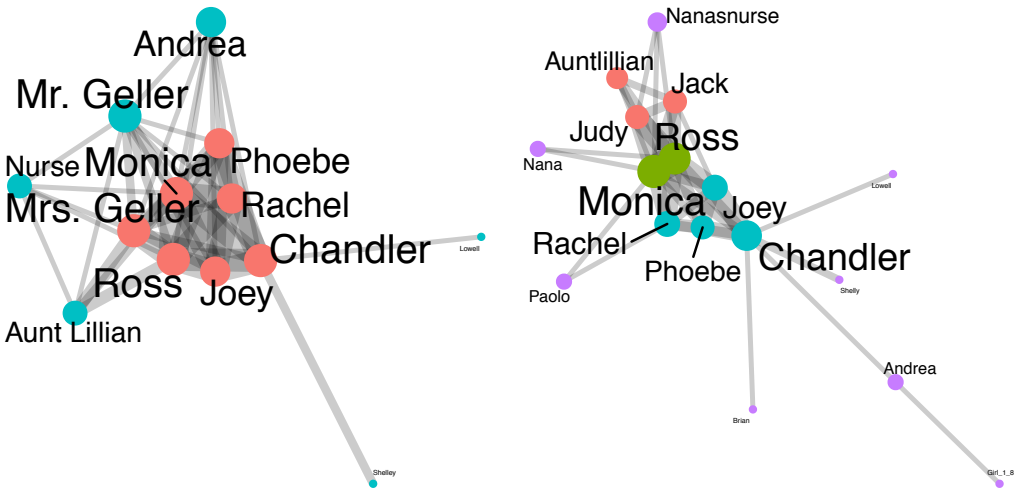

(a) **co-occurrence** (b) **manual**

Figure 6.14: **Co-occurrence** and **manual** networks for Season 1, Episode 20: *The One With the Evil Orthodontist*, with nodes coloured by class as determined by the stochastic block model.

are split into two groups: Monica and Ross, who interact with Jack, Judy and Aunt Lillian more frequently; and Rachel, Joey, Phoebe and Chandler. Jack, Judy and Aunt Lillian form a class, and the remaining non-core characters form the final class.

The majority of episode networks in both classes have two classes – sometimes the core class and non-core class, and sometimes the core class is split in two. In the **co-occurrence** dataset, 78.0% of episodes have two classes, compared to 78.9% of the **manual** networks. While these proportions are similar, Figure 6.15 demonstrates that the episodes with the same number of classes are not necessarily the same in each dataset. Some episodes had all characters in a single class (8.8% in the **co-occurrence** and 5.5% in the **manual** dataset). Some episodes have three classes (11.9% in the **co-occurrence** and 15.3% in the **manual** dataset), and the remaining episodes have four classes (1.3% in the **co-occurrence** and 0.4% in the **manual** dataset). No episodes have more than four classes in either dataset.

Figure 6.16 shows histograms of the number of classes in each season of the **co-occurrence** and **manual** dataset. There are only episodes with four classes in Season 1, as well as Season 7 and 10 for the **co-occurrence** dataset. In Seasons 1 to 7, there is a clear mode of two classes for both datasets, however for Seasons 8 and 9, there are more episodes with one or three classes. Season 10 has fewer episodes, hence fewer episodes with two




(a) **co-occurrence** (b) **manual**

Figure 6.15: **Co-occurrence** and **manual** networks for Season 1, Episode 8: *The One Where Nana Dies Twice*, with nodes coloured by class as determined by the stochastic block model.

classes, but also has very few episodes with one, three or four classes.

Notice that the number of classes the stochastic block model suggests decreases as we go from series to season to episode view. This could be because there are fewer characters when the time frame is smaller. Figure 6.17 shows scatterplots of the number of classes suggested by the stochastic block model against the size of the network for each episode of the **co-occurrence** and **manual** networks. Notice that in both datasets, in general there are more classes when there are more characters.

Any pattern in the number of classes in the episode networks as the season progresses is unclear, however, we can check for patterns statistically. It is also of interest whether episodes with more classes rate more highly than episodes with less classes. In the next sections, we search for attributes of the networks that change over time and model the success of the series.

## 6.4 Bivariate modelling with time

### 6.4.1 Episode view

Over the ten seasons of *Friends*, fans will notice the development of characters and changes in the social structure with the presences of new characters and absence of old characters. Here we use quantified attributes of the social

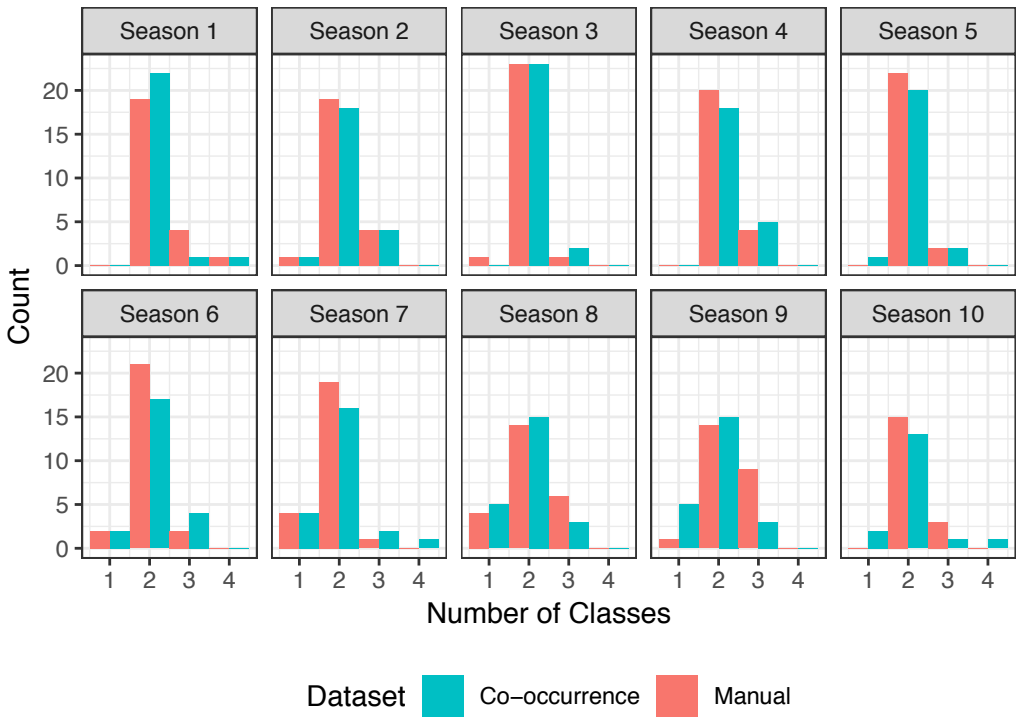



Figure 6.16: Histograms of the number of classes in each episode of the **co-occurrence** and **manual** networks as determined by the stochastic block model, split into the ten seasons.

network of *Friends* to model the network over time. We define and calculate the following variables for each episode network for the **manual** dataset:

- `words` – total number of words spoken,
- `lines` – number of speaking lines,
- `words_per_line` – average number of words spoken in each line,
- `size` – number of characters,
- `total_interactions` – total number of interactions,
- `density` – edge density,
- `clustering` – clustering coefficient,
- `classes` – number of classes suggested by the stochastic block model,
- `core_lambda` – average number of interactions between core characters,
- `core_interactions` – number of interactions between core characters,

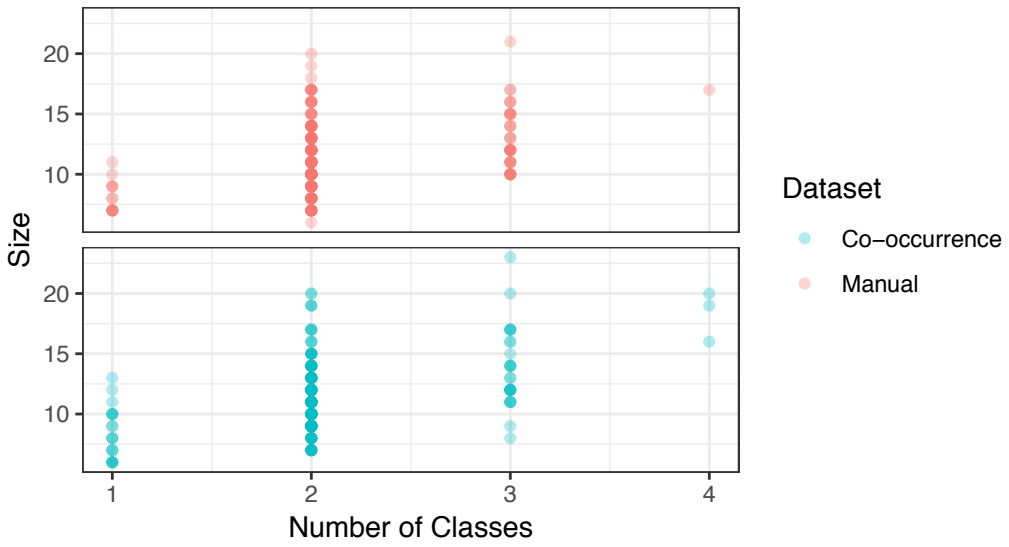



Figure 6.17: Scatterplots of the number of classes suggested by the stochastic block model against the size of the network for each episode of the **co-occurrence** (blue) and **manual** (red) networks.

- `core_proportion` – proportion of interactions that occur between core characters,
- `chan_degree` – Chandler's degree,
- `chan_norm_degree` – Chandler's normalised degree,
- `chan_betweenness` – Chandler's betweenness centrality,
- `chan_closeness` – Chandler's closeness centrality,
- `mon_degree` – Monica's degree,
- `mon_norm_degree` – Monica's normalised degree,
- `mon_betweenness` – Monica's betweenness centrality,
- `mon_closeness` – Monica's closeness centrality,
- `ross_degree` – Ross's degree,
- `ross_norm_degree` – Ross's normalised degree,
- `ross_betweenness` – Ross's betweenness centrality,
- `ross_closeness` – Ross's closeness centrality,
- `rach_degree` – Rachel's degree,

- `rach_norm_degree` – Rachel's normalised degree,
- `rach_betweenness` – Rachel's betweenness centrality,
- `rach_closeness` – Rachel's closeness centrality,
- `joey_degree` – Joey's degree,
- `joey_norm_degree` – Joey's normalised degree,
- `joey_betweenness` – Joey's betweenness centrality,
- `joey_closeness` – Joey's closeness centrality,
- `phoebe_degree` – Phoebe's degree,
- `phoebe_norm_degree` – Phoebe's normalised degree,
- `phoebe_betweenness` – Phoebe's betweenness centrality,
- `phoebe_closeness` – Phoebe's closeness centrality,
- `gunther_degree` – Gunther's degree,
- `mike_degree` – Mike's degree,
- `estelle_degree` – Estelle's degree,
- `janice_degree` – Janice's degree,
- `tag_degree` – Tag's degree,
- `jack_degree` – Jack's degree,
- `judy_degree` – Judy's degree,
- `ben_degree` – Ben's degree,
- `carol_degree` – Carol's degree,
- `emma_degree` – Emma's degree,
- `marcel_degree` – Marcel's degree,
- `rach_and_ross` – edge weight between Rachel and Ross,
- `mon_and_chan` – edge weight between Monica and Chandler,
- `joey_and_chan` – edge weight between Joey and Chandler, and
- `scenes` – number of scenes.

We check our model against a similar model for the **co-occurrence** networks in Section 6.4.3.

Initially, we create histograms of each variable, to get an idea of any outlying values we need to be aware of. We notice that the degrees of non-core characters are usually 0 as there are many episodes they are not in. We convert these variables into logical vectors for whether their degree is greater than 0 or equal to 0, *i.e.* whether they are in the episode or not. The histograms of the remaining numerical variables are in Appendix A.15.

We also plot the variables against the episode number and visually check for trends. The scatterplots of the variables against episode number are in Appendix A.15. We notice possible positive linear relationships between the number of lines and episode number, and the number of words and episode number.

We check for linear relationships by fitting a linear model to each variable independently. The response variable is the episode number. We normalise each predictor variable by dividing by the absolute maximum value so that all predictor variable values are between -1 and 1, which helps with the comparison of linear model coefficients. We calculate the p-values for the hypothesis that the coefficient of the predictor variable is 0 in each linear model, and adjust p-values using the False Discovery Rate correction, as outlined by Benjamini and Hochberg [23]. Figure 6.18 shows the p-values for each variable, with a line for the cut-off at the 5% significance level.

21 out of the 50 possible predictor variables appear to be significant predictors of the episode number, meaning there is a linear relationship between these variables and time. Figure 6.19 shows the coefficients of the normalised significant predictor variables.

The degree of Carol, Marcel, and the six core characters significantly decrease over time. Marcel only appears in the first two seasons, so it is not surprising that the coefficient of the indicator variable of whether Marcel is in the episode is negative. Similarly, Carol appears more in the first few seasons when Ross is "getting over" his relationships with her, and their son, Ben, is young. Later in the series, the producers focus on Rachel and Ross's daughter Emma, instead of Ben, so Carol does not appear (except for mentions) after Season 7.

In contrast, Emma and Mike are in the last few seasons, so their appearance increases over the series. Rachel gives birth to Emma in the final episode of Season 8, and appears in many episodes in Seasons 9 and 10. Mike is introduced in Season 9, Episode 3: *The One with the Pediatrician*, and appears frequently after that.

The six core characters interact with each other less as the series goes on. We see a similar trend in the total interactions and core interactions, which decrease over time too. Figure 6.20 shows a scatterplot of the number of interactions between core characters versus the episode number, with a

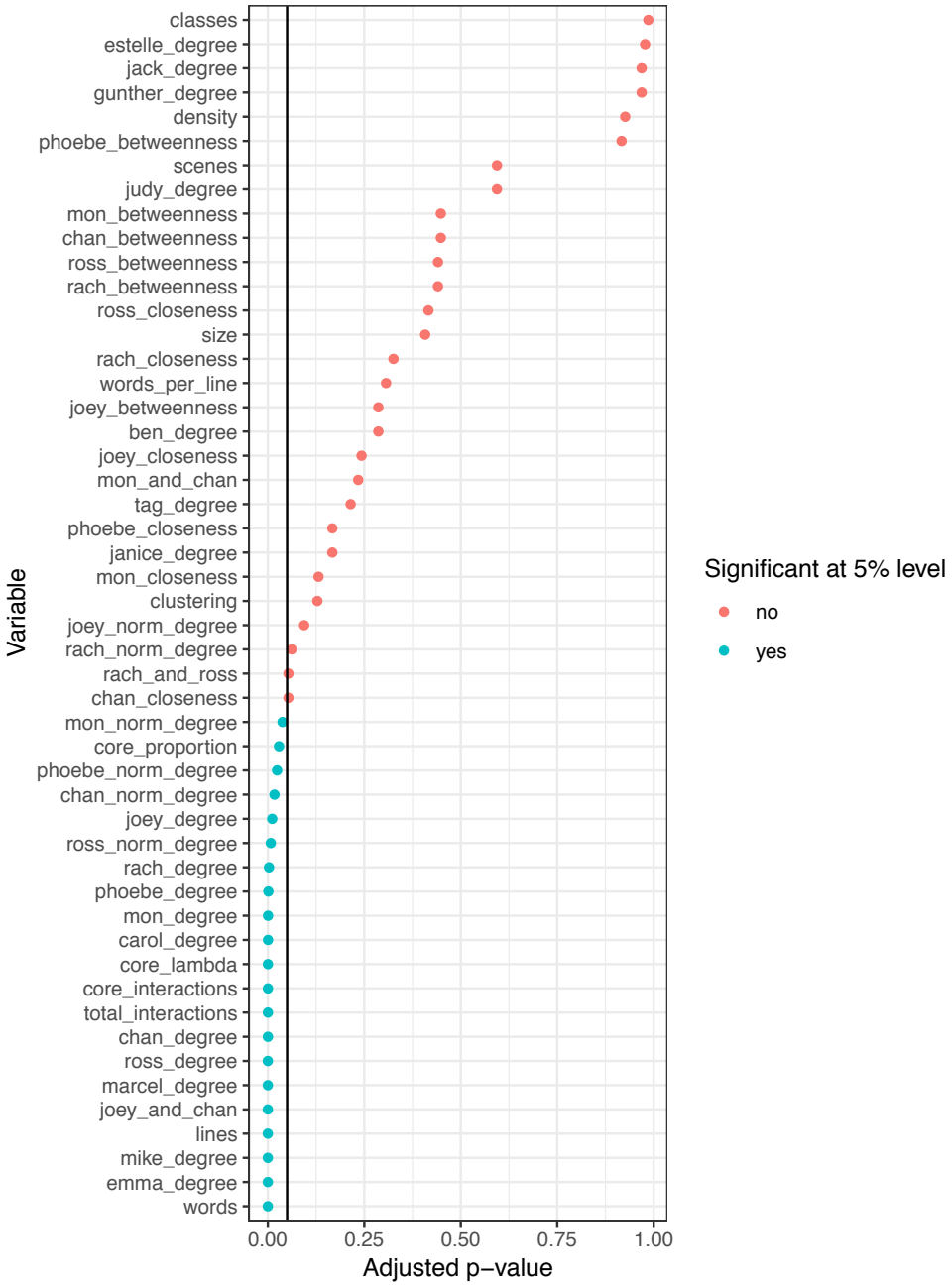



Figure 6.18: Scatterplot of adjusted p-values for the linear models of each variable with episode number for the **manual** episode networks. The black line is at 0.05, which is the cut-off for p-values at the 5% significance level. Significant variables are coloured in blue, and insignificant variables are red.

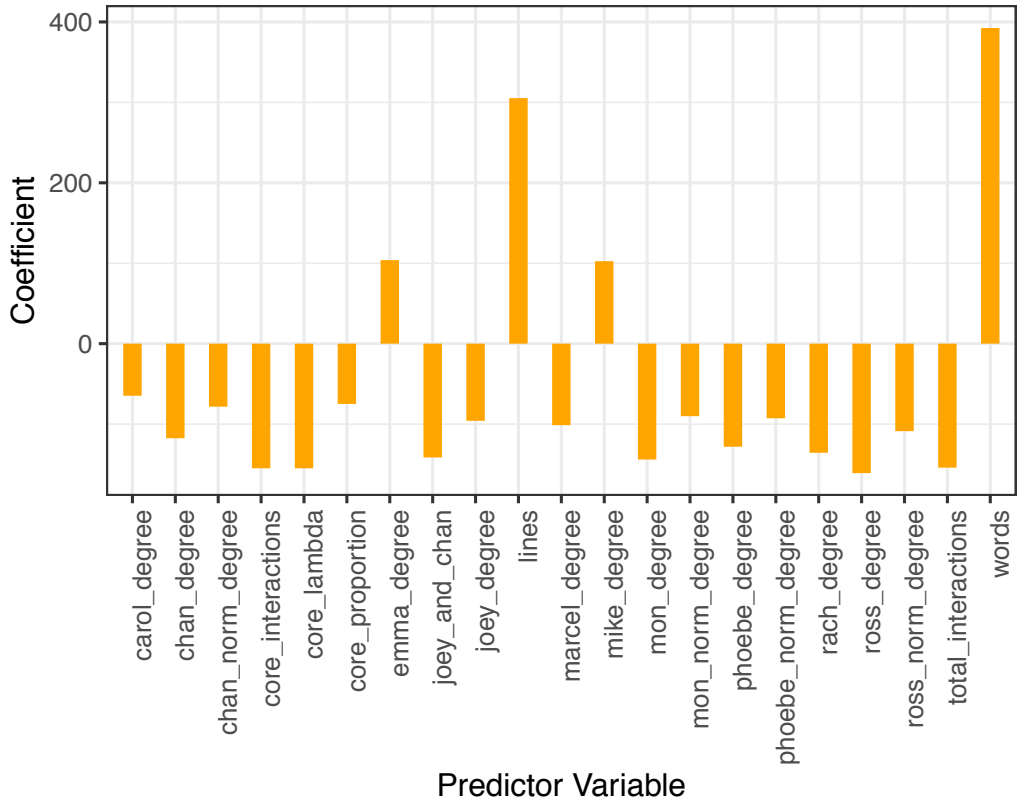



Figure 6.19: Line range plot of coefficients of normalised significant predictors for episode number for the **manual** episode networks.

line representing the predicted mean and confidence interval for the mean. The proportion of interactions that are between core characters significantly decreases over time as well, but the coefficient is smaller in magnitude than for total and core interactions, so the change over time is not as drastic.

The negative coefficient of `joey_and_chan` means that Joey and Chandler interact less over the series. As we discussed in Chapter 5, this happens because Chandler and Joey are very close in the first few seasons when they are roommates, but when Chandler and Monica start their relationship, Chandler's interactions occur more with Monica than with Joey. Figure 6.21 shows scatterplots of the edge weights between Joey and Chandler, and Chandler and Monica over the 236 episodes, with the linear prediction of the means. Interestingly, there is no significant linear trend in the number of interactions between Monica and Chandler over time. This is because overall, the characters interact less, and so even though most of Chandler's interactions are with Monica, Chandler makes fewer interactions in total.

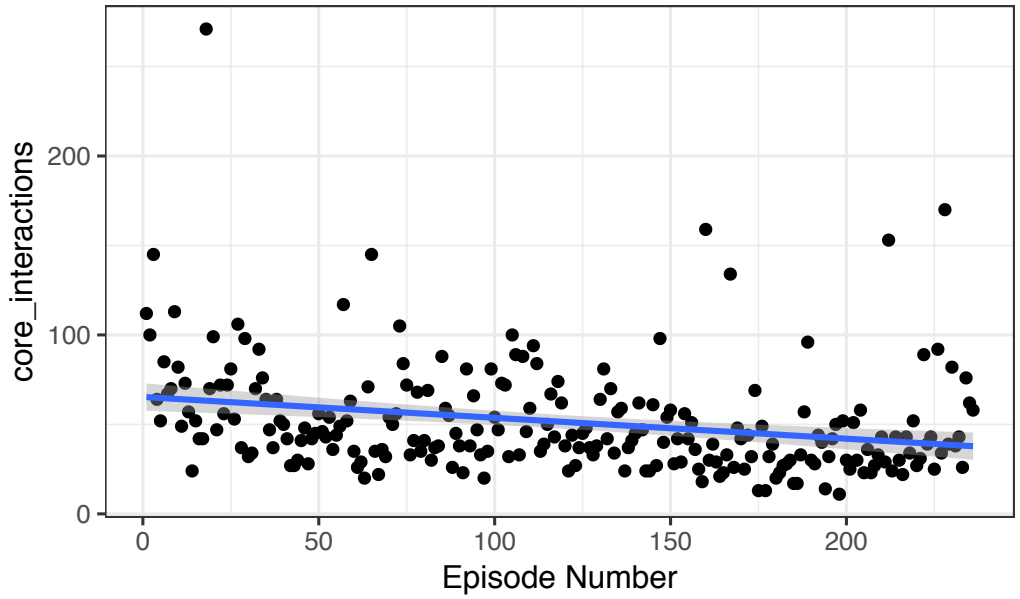



Figure 6.20: Scatterplot of the number of interactions between core characters (`core_interactions`) with episode number for the **manual** episode networks. The line represents the predicted mean and the shaded region represents the confidence interval for the mean.

Finally, the number of words and number of lines in each episode increase significantly throughout the series. Figure 6.22 shows a scatterplot of the number of words versus the episode number. The predicted average number of words increases by almost 500 from Episode 1 to Episode 236, which is an increase of 20.8%.

The increase in number of words over the series could be because of changes in cultural speaking patterns over the 10 years *Friends* was aired, or changes in development of characters and plot. Sherman analysed how the length of sentences changed from the 16th to 19th century and found that sentences have become 75% shorter over this time [110, 121]. However, changes in a span of 10 years are difficult to quantify. Also note that the number of lines increased significantly as well as the number of words, and the number of words per line was not a significant predictor for episode number at the 5% level, so there is no evidence for characters speaking for longer periods of time at once. To examine language changes between the years *Friends* was aired (1994 – 2004), one could investigate the number of words and lines per episode in other television shows around that time, and how they change over time.

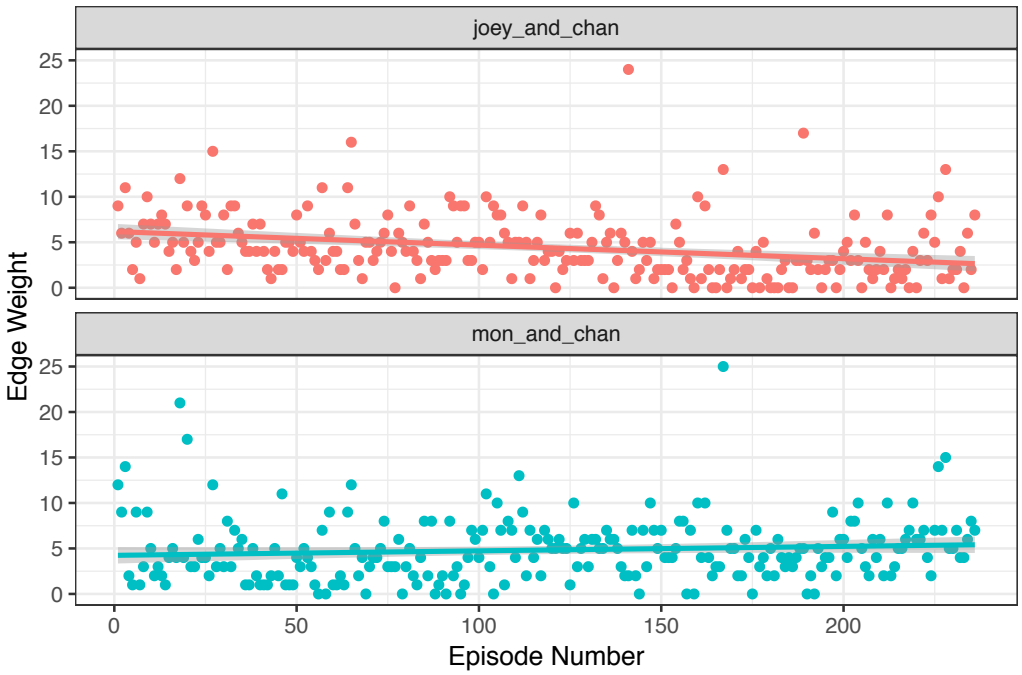



Figure 6.21: Scatterplots for the edge weights between Joey and Chandler (red) and between Monica and Chandler (blue) for the **manual** episode networks. The lines represent the predicted mean and the shaded region represents the confidence interval for the mean.

Looking at trends for the number of words and lines per episode could also provide evidence that producers increase the number of words and lines characters have as a television series develops. Fans of another American situation comedy, *Seinfeld* (1989 – 1998), calculated the percentage of words per character per season [9]. Using their extracted number of words spoken each season, we find that characters speak significantly more (see Appendix A.13 and Appendix B.3) in later seasons.

However, fans of other shows performed similar analyses and found that the number of words or lines in the episode decreases as the series goes on. For many characters in *The West Wing* (1999 – 2006), the number of lines per episode decreases [114]. Similarly in *The Walking Dead* (2010 – present), the average number of words spoken by Daryl Dixon per episode in each season decreases significantly [10] (see Appendix A.14 and Appendix B.4). We see a similar trend in the number of words spoken by four main characters from *The Office* (2005 – 2013) [8].

One explanation for this difference is that "90s sitcoms" such as *Friends*

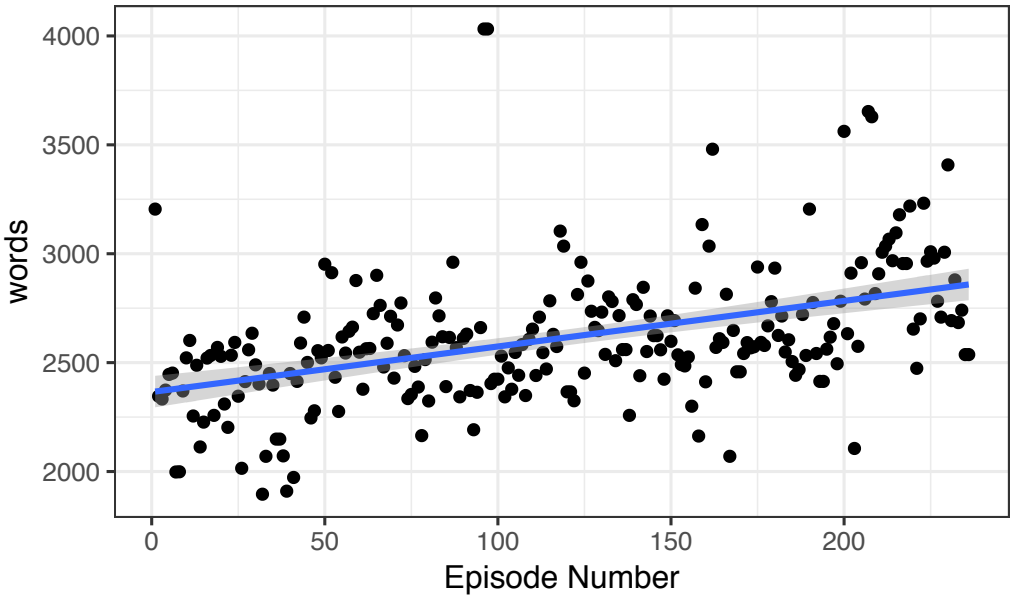



Figure 6.22: Scatterplot of number of words (`words`) with episode number for the **manual** episode networks. The line represents the predicted mean and the shaded region represents the confidence interval for the mean.

and *Seinfeld* have more words and lines as the series develops because viewers are familiar with the way characters speak, so they can fit more words and lines in. In drama series such as *The West Wing* and *The Walking Dead*, however, there is less need for speaking when the viewers are familiar with the characters.

### Predicting season number using episode metrics

So far we have considered relationships between variables and episode number, but this only allows for one datapoint for each episode. Alternatively, we could consider relationships between the variables and season number, where each season has 18 to 25 datapoints. We use the same method to fit the linear model with season number as the response variable instead of episode number. Figure 6.23 shows the adjusted p-values of the model fit in this way.

Comparing the model where the response variable is the episode number, to the one with season number as the response variable, `core_proportion` and `mon_norm_degree` are not significant predictors of season number at the episode view. Figure 6.24 shows the coefficients of the normalised significant predictors of season number. The variables have similar relationships as with

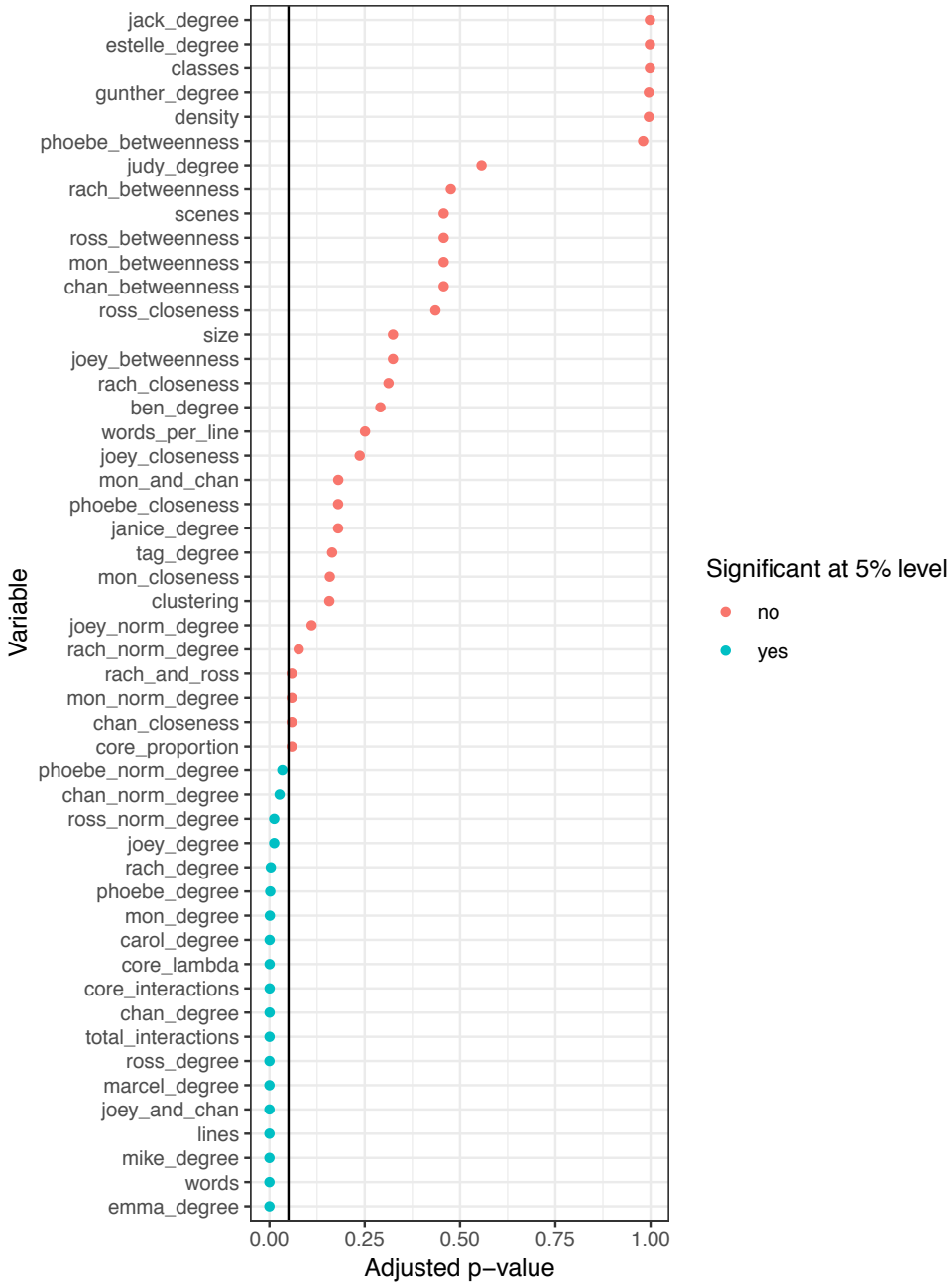



Figure 6.23: Scatterplot of adjusted p-values for the linear models of each variable with season number for the **manual** episode networks. The black line is at 0.05, which is the cut-off for p-values at the 5% significance level. Significant variables are coloured in blue, and insignificant variables are red.

the episode number, but the coefficients are smaller as the seasons only go from 1 to 10, as opposed to the episodes which go from 1 to 236. This supports our previous arguments that there are trends in the number of words, lines and interactions over time.

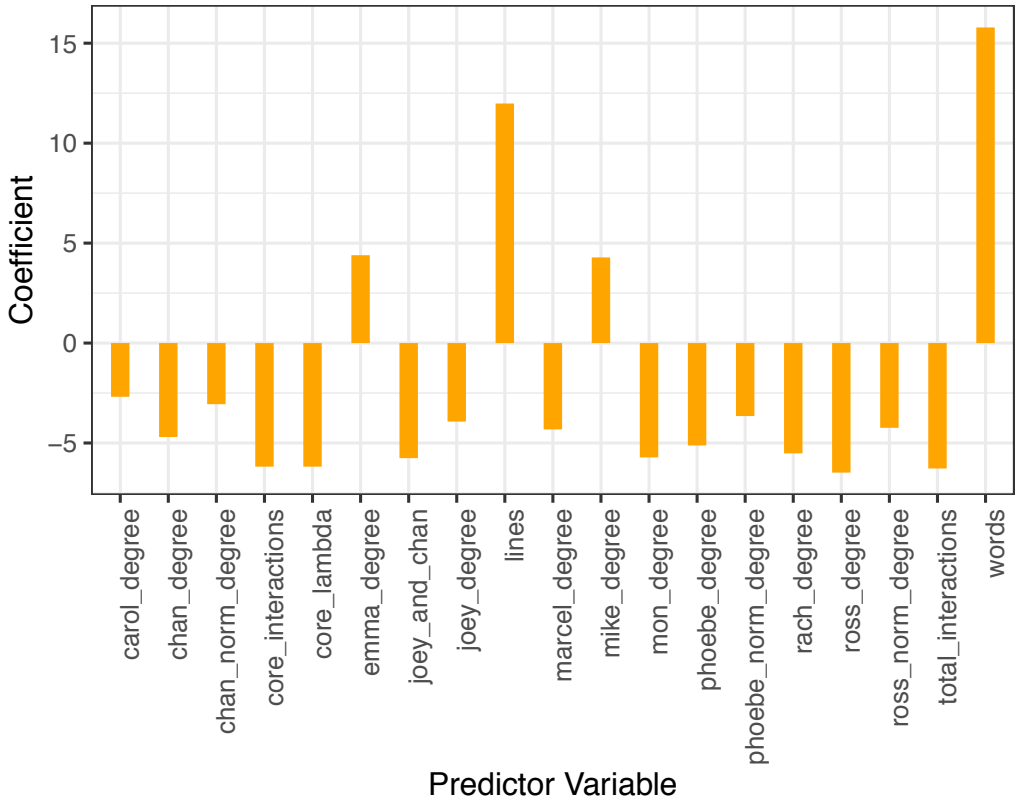



Figure 6.24: Line range plot of coefficients of normalised significant predictors for season number for the **manual** episode networks.

### 6.4.2 Season view

We also look for bivariate relationships with time and network features at the season view. We measure the same variables as above for the season view **manual** networks and check for linear relationships by fitting linear models, with the variables as predictors and season number as the response. Figure 6.25 shows the adjusted p-values of the coefficients of the variables, with a line at the 5% significance level cut-off.

Note that there are NA values for the p-value of the coefficients of `judy_degree`, `jack_degree` and `janice_degree`. These appear because Judy, Jack and

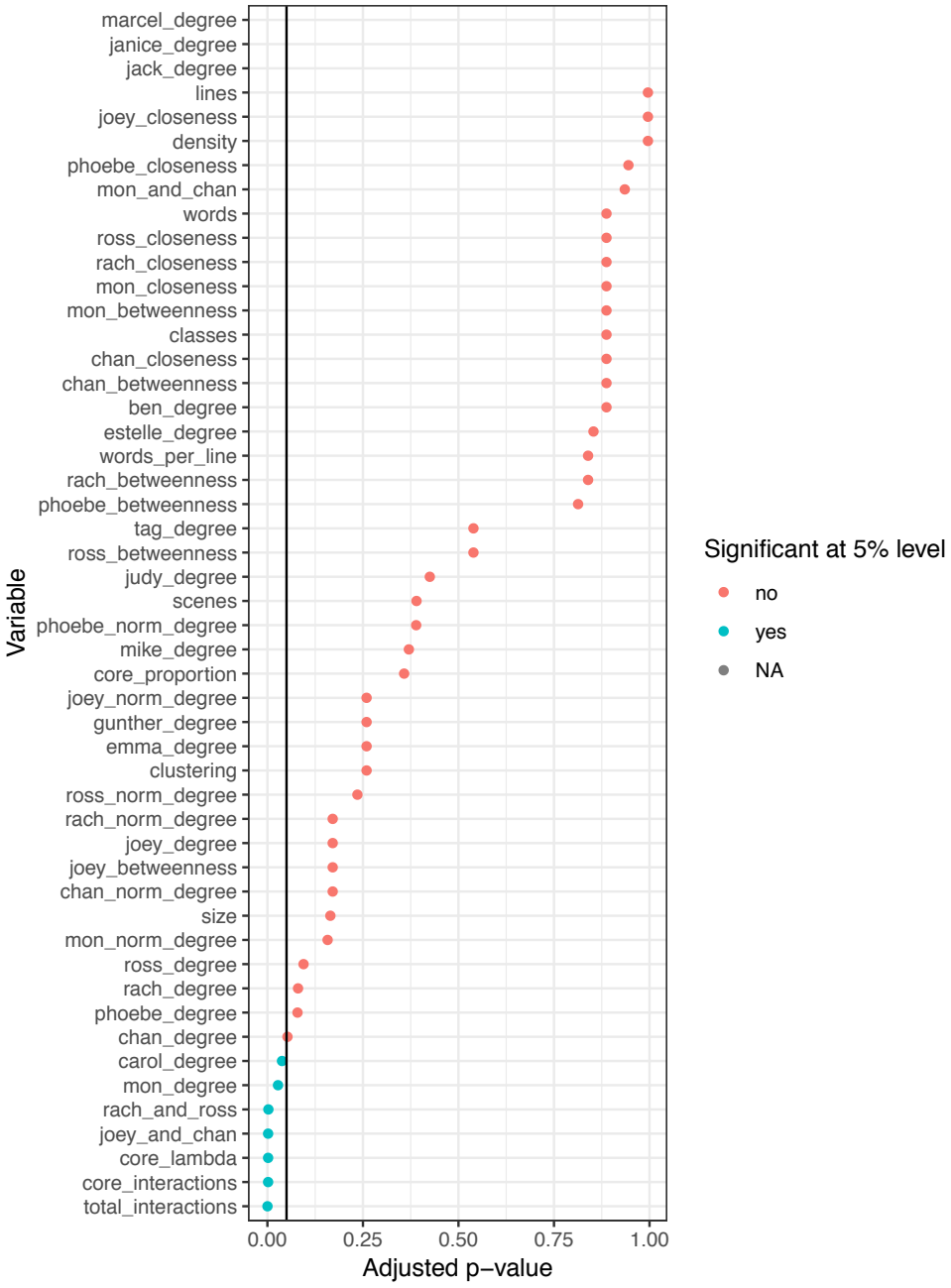



Figure 6.25: Scatterplot of adjusted p-values for the linear models of each variable with season number for the **manual** season networks. The black line is at 0.05, which is the cut-off for p-values at the 5% significance level. Significant variables are coloured in blue, and insignificant variables are red.

Janice appear at least once in every season of *Friends*.

The significant predictors for season number at the season view are `joey_and_chan`, `carol_degree`, `core_interactions`, `core_lambda`, `emma_degree` and `total_interactions`. Figure 6.26 shows the coefficients of the normalised significant predictor variables.

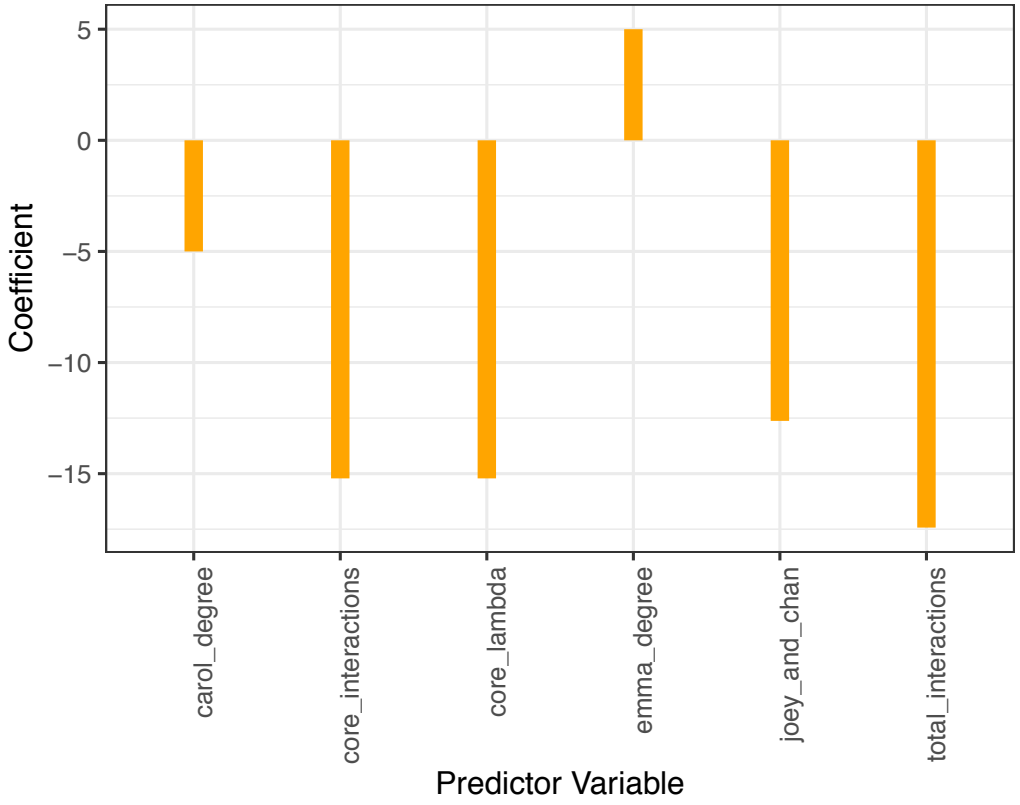



Figure 6.26: Line range plot of coefficients of normalised significant predictors for season number for the **manual** season networks.

As in the episode number model, Carol's presence decreases over time and Emma's presence increases over time. The number of interactions between Joey and Chandler also significantly decreases over time. Notice that the number of words and lines are not significant predictors at the season view because seasons with fewer episodes will have fewer words and lines. Figure 6.27 illustrates this point with a scatterplot of the number of words in each season. Season 10 has the least words as it only has 18 episodes. If we average the number of words by the number of episodes in each season, the model will be similar to that in Section 6.4.1.

The other significant predictors of season number at the season view

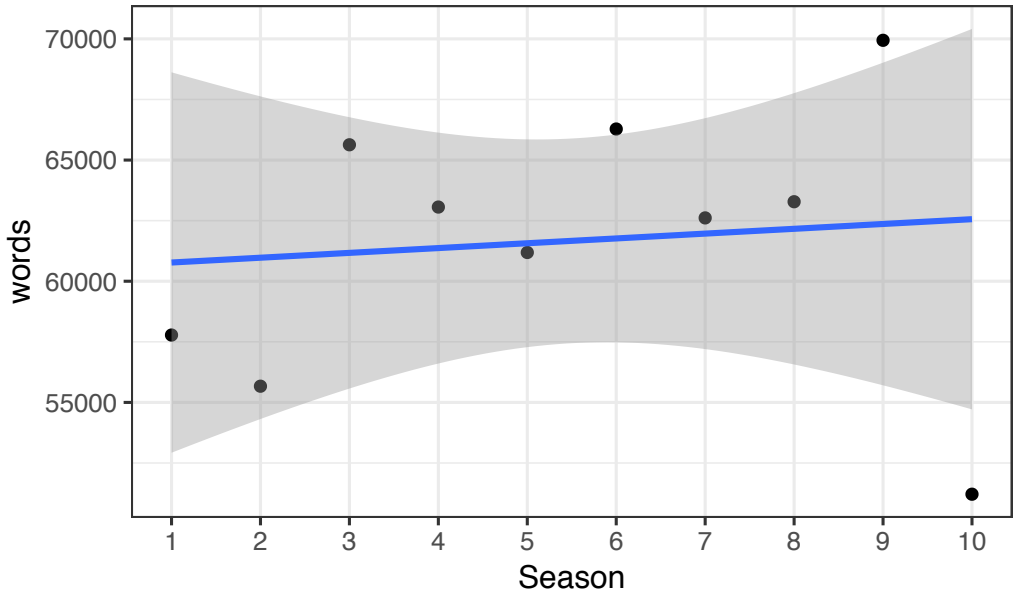



Figure 6.27: Scatterplot of number of words (`words`) with season number for the **manual** season networks. The line represents the predicted mean and the shaded region represents the confidence interval for the mean.

relate to the edge weights. The coefficients for these variables have the greatest magnitude, meaning they change the most with the seasons. The `core_lambda` is a scaled version of `core_interactions`, so when they are both scaled between -1 and 1, they are equal. Figure 6.28 shows a scatterplot of the decreasing number of interactions between core characters at the season view with a confidence interval for the predicted mean. The trend indicates that "the *Friends* get less friendly" over the series.

It is interesting, however, that the total number of interactions also decreases over the series. This is because most of the interactions occur between the core characters, so core interactions follow a similar trend to total interactions. Therefore the trend in number of interactions between core characters cannot be explained by the core characters becoming more friendly with non-core characters and spending their interactions with them. Also, the proportion of interactions that are between core characters was not a significant predictor of season number.

One theory to explain this trend is that as the series develops, characters can have deeper, longer interactions, as there are more complex storylines. The increasing number of words and lines per episode backs this theory up, as longer conversations mean characters can say more. Further evidence for

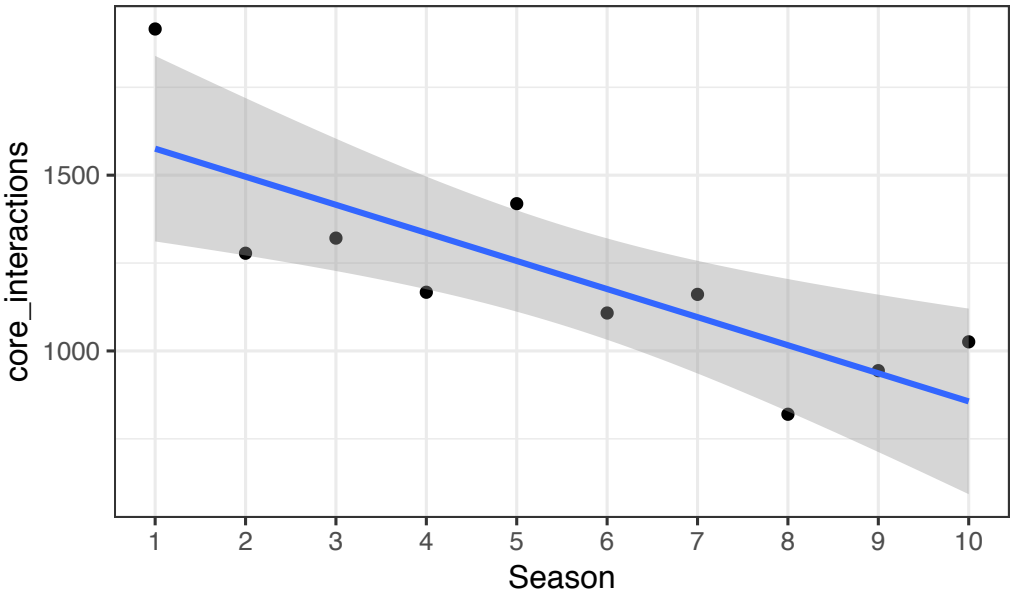



Figure 6.28: Scatterplot of the number of interactions between core characters (`core_interactions`) with season number for the **manual** season networks. The line represents the predicted mean and the shaded region represents the confidence interval for the mean.

this theory involves investigating other television shows, and narratives to see if similar trends are present, or if it is only the *Friends* that get less friendly.

### 6.4.3 Co-occurrence networks

We also model the same variables over time for the **co-occurrence** dataset. We find that the main results are consistent between the datasets. Figures for the **co-occurrence** bivariate analysis with time are in Appendix A.16. Here, we discuss the differences between the models for the two datasets.

#### Episode view

At the episode view, the variables `clustering` and `rach_and_ross` significantly decrease over time for the **co-occurrence** networks, even though they were not significant in the **manual** networks. The variables `emma_degree` and `marcel_degree` are not significant, which is unsurprising because as discussed in Chapter 5, these characters show up in the **co-occurrence** dataset a lot less as they do not talk.

Most of the predictor coefficients are similar across the datasets, but `joey_and_chan` has a larger coefficient in the **co-occurrence** networks. This means that while Joey and Chandler speak to each other much more earlier on in the series, the change in how many scenes they share is even more drastic.

When we average the **co-occurrence** variables over seasons, we get the same significant predictors, excluding `clustering`. Again, the coefficients are similar in scales to the coefficients for the episode times.

#### Season view

Looking at the variables from the season view, the variables that change significantly over time are the same in both datasets, except for `emma_degree`, which is only significant in the **manual** networks, and `mon_degree` and `rach_and_ross`, which are only significant in the **co-occurrence** networks.

The coefficients of the significant predictors for the **co-occurrence** season number are similar to those for the **manual** season number. In particular, the number of core interactions and total interactions decrease significantly over the 10 seasons, even though there is no significant linear trend in the number of scenes in each season.

## 6.5 Bivariate modelling with ratings

### 6.5.1 Predicting success

Our previous models look at the relationship of features of the networks over time. We can use similar features to predict the success of the *Friends* episodes. If certain features correlate with measures of success of episodes, we can infer the features that make an episode better. This information can potentially lead to creating better narratives by structuring the social network optimally.

To measure the success of each episode, we extract the ratings from IMDb, as discussed in Chapter 3. We also extract the number of IMDb users that rated each episode so we can adjust our model for possible bias from this. Figure 6.29 shows a scatterplot of the ratings for each episode, coloured by season. The five highest and five lowest rated episodes are labelled.

We use the variables listed in Section 6.4, as well as the number of IMDb users who rate they episode (`n_ratings`) to fit a multivariate linear model to predict the ratings of episodes.


Figure 6.29: Scatterplot of the ratings for each episode, coloured by season. The five highest and five lowest rated episodes are labelled.

### 6.5.2 Transformation of variables

First, we plot scatterplots of each variable with rating for every episode to check for non-linearity, outliers, *etc.* The scatterplots are in Appendix A.17. Some plots have a datapoint which could be an outlier, but we fit the model as in the standard manner to begin with. We also notice two variables that could have a non-linear relationship with `rating`: `n_ratings` and `density`. One of the assumptions of the linear model is that there is a linear relationship between the response and predictor variable, so we look at transformations for these variables.

#### Number of ratings

We compare three transformations on the number of ratings data: exponential transform, log transform and inverse transform. Figure 6.30 shows scatterplots of the transformed variables with rating. We divide the number of ratings by 1000 for the exponential transformation to adjust for computational storage limits.

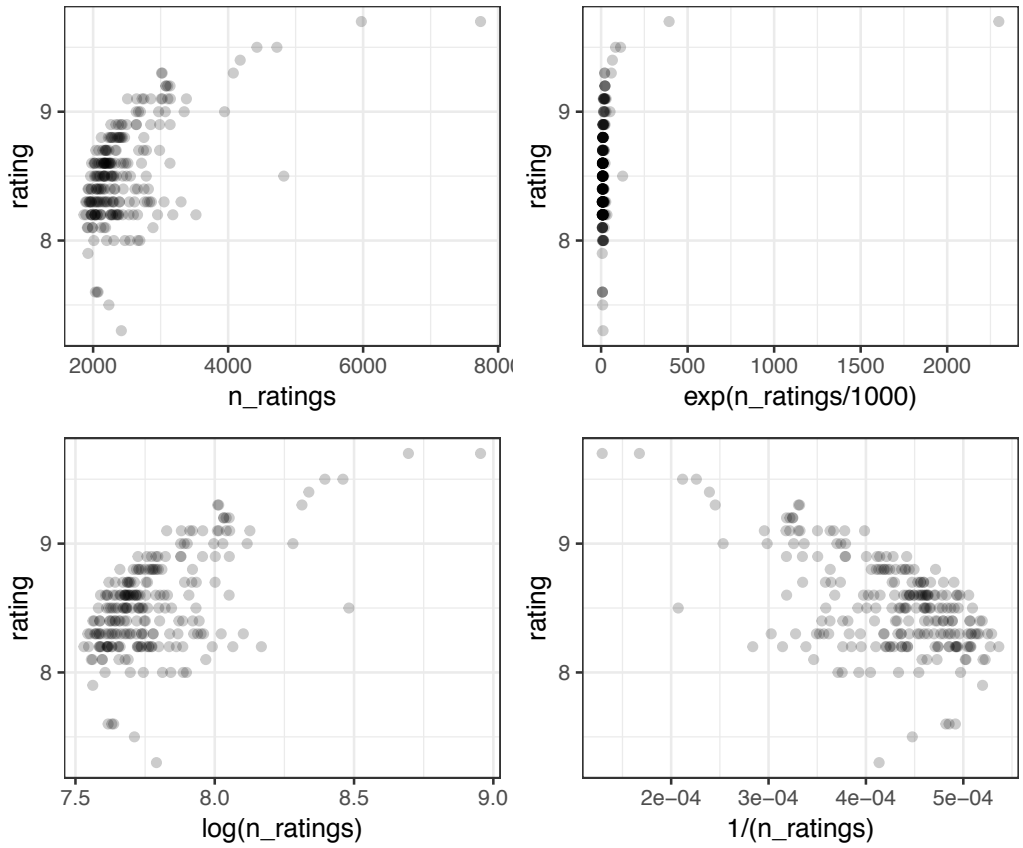



Figure 6.30: Scatterplots of the number of IMDb users rating each episode, `n_ratings`, with the average ratings, `rating`, for the raw data and three transformations: exponential (top right), log (bottom left) and inverse (bottom right).

We also fit linear models to each of these transformations, and find that the log transformation is the best, as it has the smallest residual standard error. We define a new variable

$$\texttt{n\_ratings\_transformed} = \log\left(\texttt{n\_ratings}\right).$$

### Density

Similarly, we compare the same three transformations on the edge density data. Figure 6.31 shows scatterplots of the transformed variables with rating.

Again, we fit linear models to each of these transformations, and find that the exponential transformation is the best, as it has the smallest residual

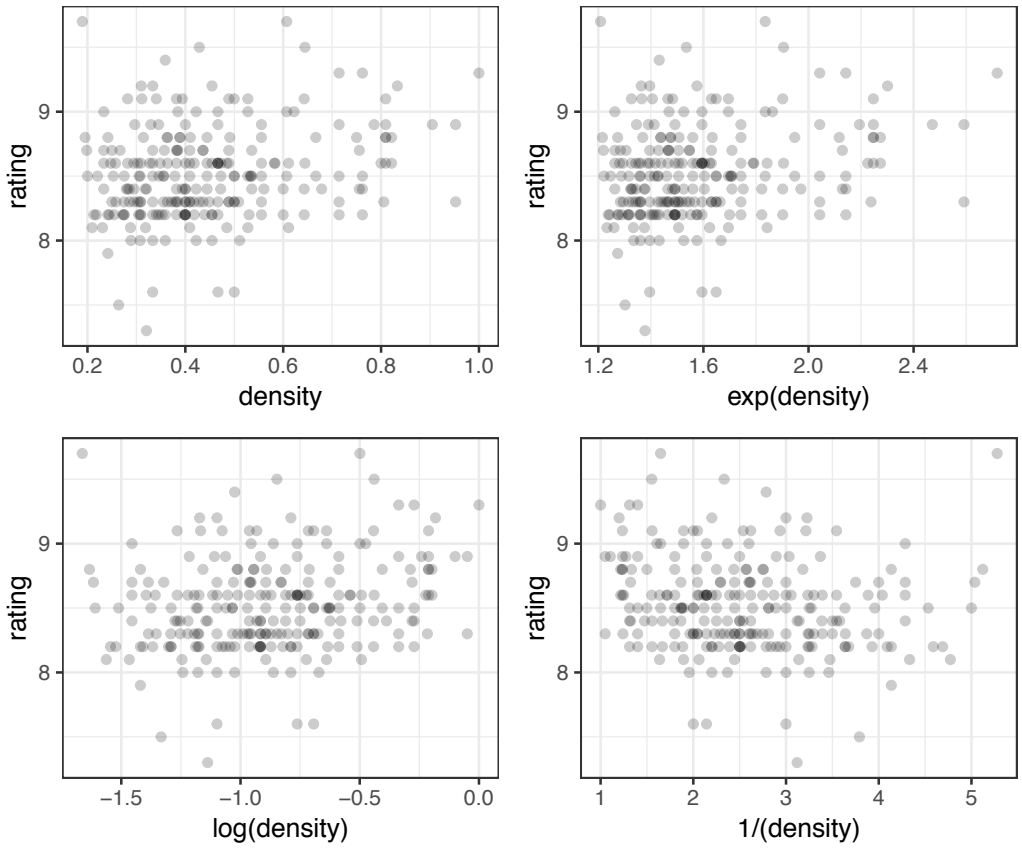



Figure 6.31: Scatterplots of the edge density of each episode, `density`, with the average ratings, `rating`, for the raw data and three transformations: exponential (top right), log (bottom left) and inverse (bottom right).

standard error. We define a new variable

$$\texttt{density\_transformed} = \exp(\texttt{density}).$$

### 6.5.3 Full model

We then fit individual linear models to each of our possible predictor variables, with `rating` as the response variable, similarly to the bivariate model with time in Section 6.4. We adjust the p-values to account for the false discovery rate of fitting many models. Figure 6.32 shows the p-values for the coefficients of each variable not equalling zero.

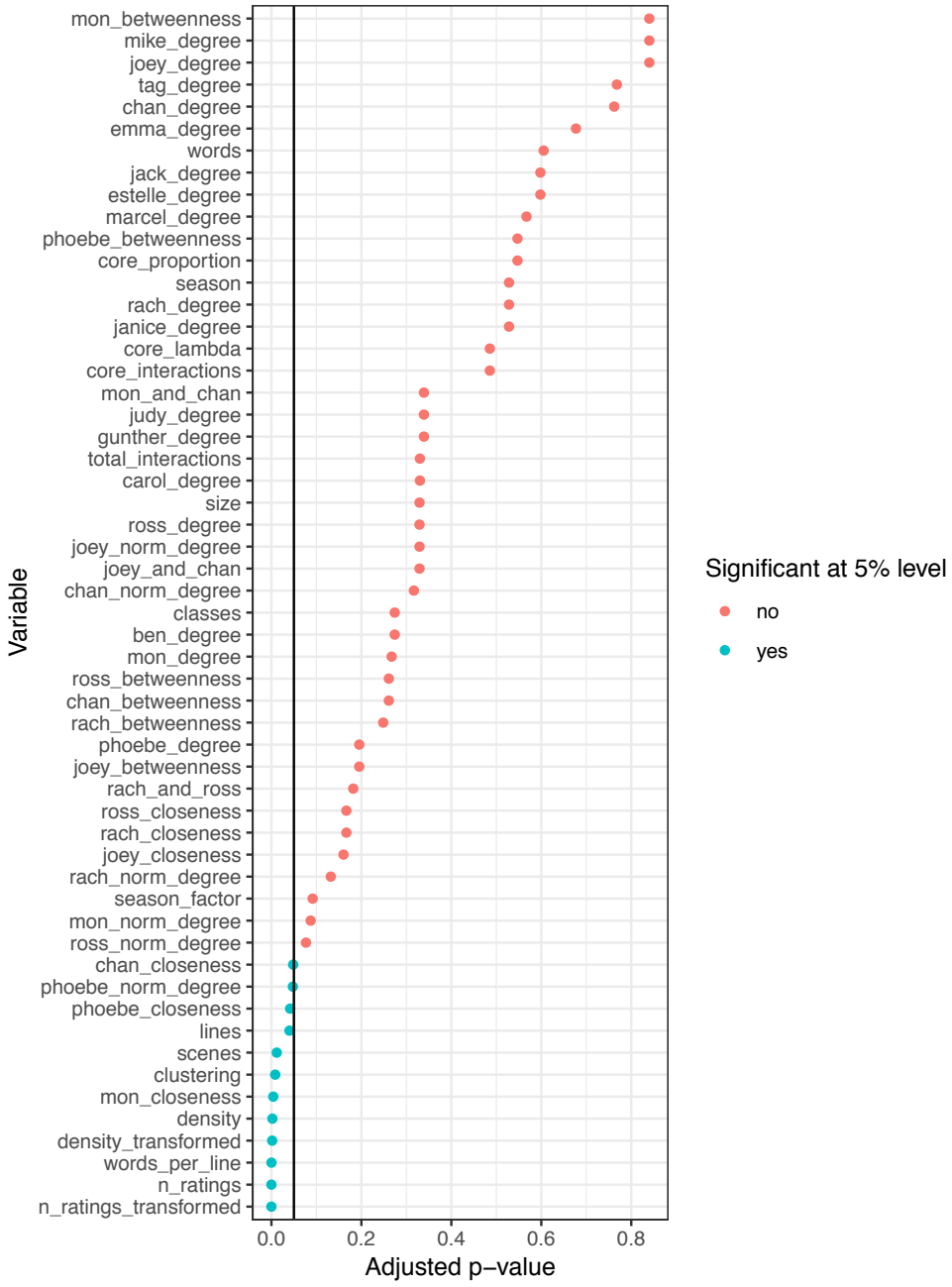



Figure 6.32: Scatterplot of adjusted p-values for the linear models of each variable with IMDb ratings. The black line is at 0.05, which is the cut-off for p-values at the 5% significance level. Significant variables are coloured in blue, and insignificant variables are red.

### 6.5.4 Model selection

We put every variable with a significant linear relationship with rating at the 5% level into a multivariate linear model:

$$\texttt{rating}_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i,$$

for datapoint $i = 1, \ldots, 236$ and

$$\boldsymbol{x}_i = \begin{bmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{im} \end{bmatrix}, \ \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \ \epsilon_i \sim N(0, \sigma^2).$$

Here, $\boldsymbol{x}_i$ are the predictors for the $i$th rating datapoint, and $\boldsymbol{\beta}$ are the coefficients. The resulting model, not all of the predictors are significant, so we use AIC backwards stepwise selection to find a suitable model.

### 6.5.5 Final model

The selected model from the AIC backwards stepwise selection method has six predictors: `chan_closeness`, `lines`, `scenes`, `clustering`, `density_transformed` and `n_ratings_transformed`. Details of the selected model are below:

```
Call:
lm(formula = rating ~ chan_closeness + lines + scenes + clustering +
    density_transformed + n_ratings_transformed, data = df_features)

Residuals:
     Min       1Q   Median       3Q      Max
-0.79852 -0.16874  0.02146  0.19785  0.53874

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.0027527  0.7468594  -1.343 0.180723
chan_closeness        -0.2496370  0.1523675  -1.638 0.102714
lines                  0.0021893  0.0005477   3.997 8.65e-05 ***
scenes                -0.0110943  0.0029644  -3.742 0.000230 ***
clustering            -0.5573694  0.2317373  -2.405 0.016959 *
density_transformed    0.5039654  0.1281961   3.931 0.000112 ***
n_ratings_transformed  1.1420232  0.0940671  12.141  < 2e-16 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2734 on 229 degrees of freedom
Multiple R-squared:  0.4603,Adjusted R-squared:  0.4461
F-statistic: 32.55 on 6 and 229 DF,  p-value: < 2.2e-16
```

Notice that `chan_closeness` is not a significant predictor in this model at the 5% level. We fit a multivariate model without `chan_closeness` and compare the AICs of the model. The model with `chan_closeness` has an AIC of 66.5, and without has an AIC of 67.3. The latter model is simpler, and the AIC is not more than 2 greater than the AIC of the former model, so we remove the predictor `chan_closeness` from the model. The resulting model is:

```
Call:
lm(formula = rating ~ lines + scenes + clustering + density_transformed +
    n_ratings_transformed, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.81993 -0.16947  0.02233  0.20127  0.52915

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -0.8964700  0.7467562  -1.200 0.231186
lines                  0.0022416  0.0005488   4.085  6.1e-05 ***
scenes                -0.0111366  0.0029751  -3.743 0.000230 ***
clustering            -0.5810124  0.2321329  -2.503 0.013012 *
density_transformed    0.4046954  0.1133867   3.569 0.000436 ***
n_ratings_transformed  1.1240510  0.0937668  11.988  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2744 on 230 degrees of freedom
Multiple R-squared:  0.4539,Adjusted R-squared:  0.4421
F-statistic: 38.24 on 5 and 230 DF,  p-value: < 2.2e-16
```

The model suggests that the number of scenes and the clustering coefficient of the episode network have a negative effect on the rating, whereas the number of lines and the edge density have a positive effect on the rating. Before we thoroughly analyse the model, we check the model assumptions.

### 6.5.6 Model assumptions

We check the assumptions of the linear model using diagnostic plots. Figure 6.33 shows the residuals versus leverage plot of this model, and we find the 94th datapoint is a potential outlier. This datapoint represents Season 4, Episode 21: *The One with the Invitation*, which is a flashback episode and has the lowest average IMDb rating of all the episodes, 7.3. A flashback episode contains clips of past episodes with little new content, so is not representative of most *Friends* episodes. Therefore we remove this episode from our dataset and refit the model.

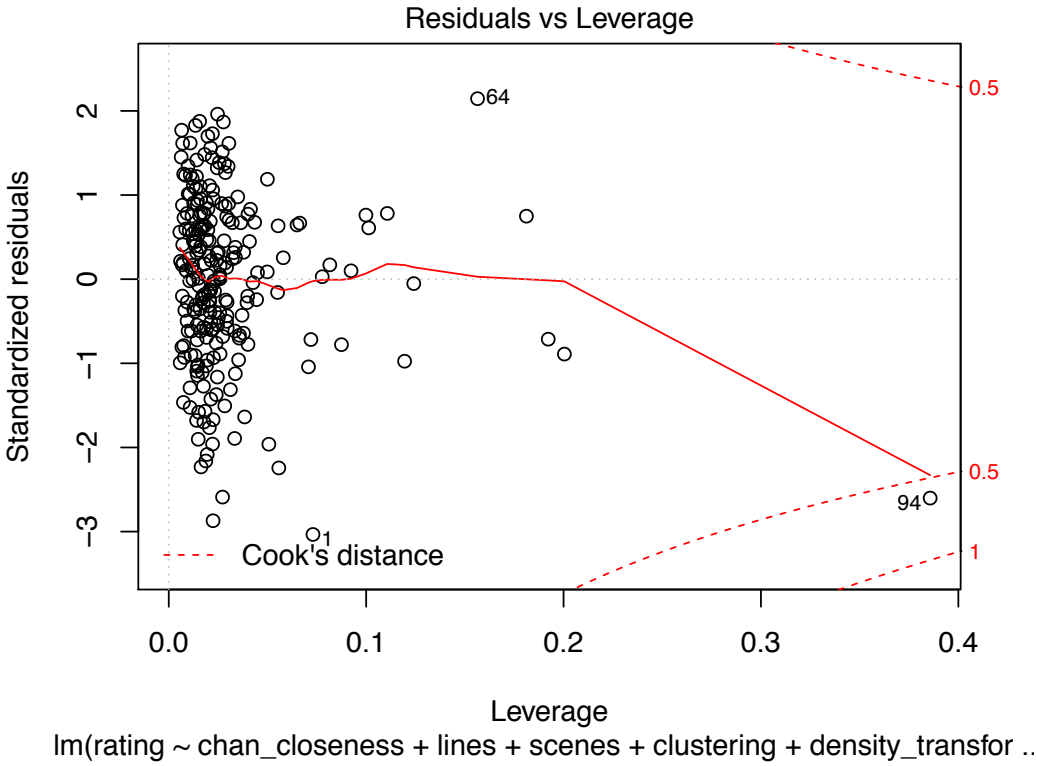



Figure 6.33: Residual versus leverage plot of the final model with all datapoints. Points outside the red dotted line representing Cook's distance are particularly influential and could be considered outliers.

We use the same method to transform variables, find significant linear predictors and select predictors for the final model. Details of the model selection process are in Appendix A.17. The final model with the 94th datapoint re-

moved has four predictors: `lines`, `clustering`, `density_transformed` and `n_ratings_transformed`. The model summary is:

```
Call:
lm(formula = rating ~ lines + clustering + density_transformed +
    n_ratings_transformed, data = df_features)

Residuals:
     Min       1Q   Median       3Q      Max
-0.86997 -0.15317  0.01449  0.19806  0.50250

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -1.0550336  0.7378755  -1.430 0.154124
lines                   0.0020924  0.0005425   3.857 0.000149 ***
clustering             -0.5698645  0.2294528  -2.484 0.013721 *
density_transformed     0.4330201  0.1120229   3.865 0.000144 ***
n_ratings_transformed   1.1212815  0.0927557  12.089  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2715 on 230 degrees of freedom
Multiple R-squared:  0.439,Adjusted R-squared:  0.4292
F-statistic: 44.99 on 4 and 230 DF,  p-value: < 2.2e-16
```

We check the assumptions of the new model using diagnostic plots in Figures 6.34, 6.35 and 6.36. Figure 6.34 shows scatterplots of residuals versus predictors for the final model without datapoint 94. The residuals appear random, as there are no clear patterns in these plots, so the assumption of linearity between residuals and predictors is reasonable.

The residuals versus fitted scatterplot in Figure 6.35 also has no obvious trend, which provides evidence that the residuals are randomly distributed. The scale-location plot also has no obvious trend, providing evidence for homoscedasticity of residuals. There are no datapoints outside of Cook's distance in the residuals versus leverage plot, so no more indications of outliers.

The normal quantile plot of the residuals has some curvature at the ends, but the middle datapoints lie on the line. This means there could be some skew in the residuals. Figure 6.36 shows the normal quantile plot of the residuals from the model next to randomly generated normal data with the same number of datapoints, same mean and same variance as the residuals. The real data has more curvature than the randomly generated data, suggesting that the residuals are not from a normal distribution. Figure 6.37

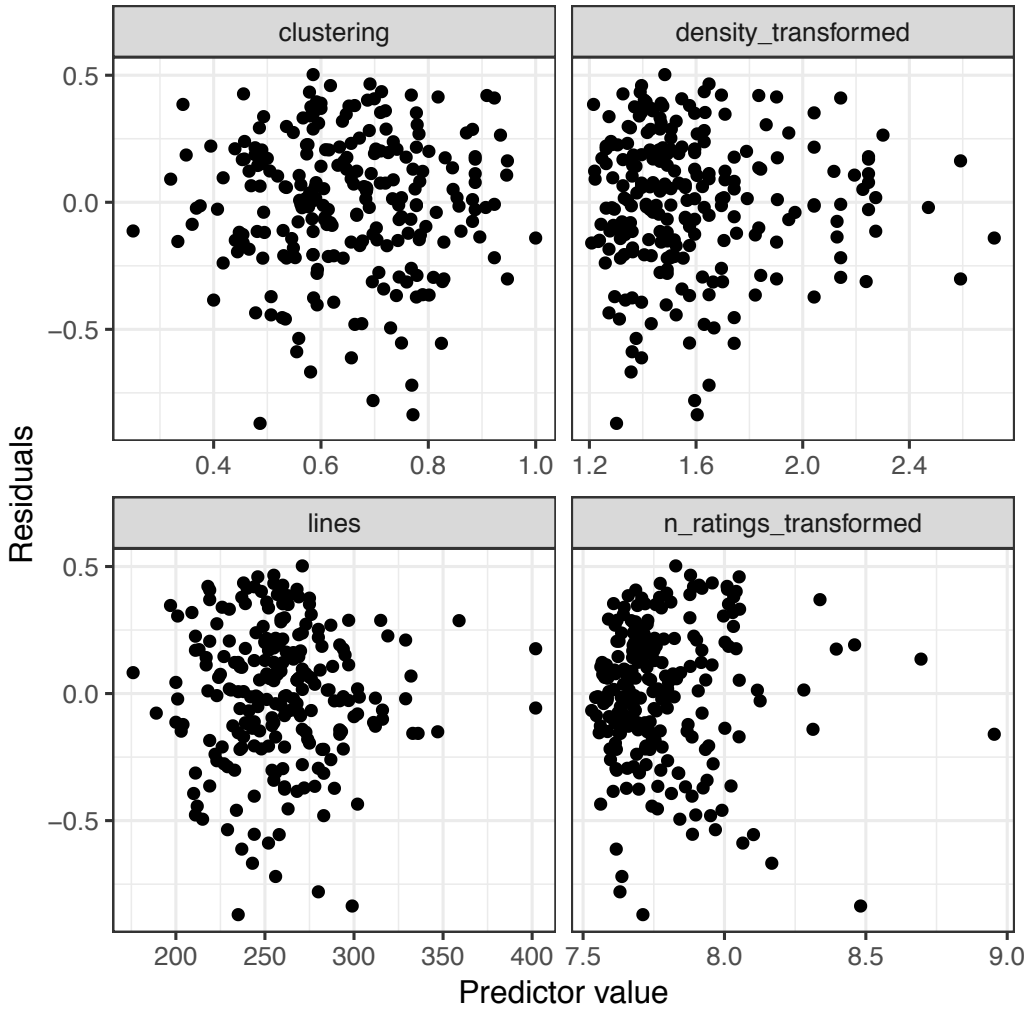



Figure 6.34: Scatterplots of residuals from the final model without datapoint 94 versus model predictors: `clustering`, `density_transformed`, `lines` and `n_ratings_transformed`.

shows a histogram of the residuals. As the normal quantile plot suggested, the residuals are left-skew. Therefore, a few datapoints have very low ratings.

This could be attributed to other flashback episodes, such as Season 6, Episode 20: *The One with Mac and C.H.E.E.S.E.*; Season 7, Episode 21: *The One with the Vows*; Season 8, Episode 19: *The One with Joey's Interview*; and Season 9, Episode 10: *The One with Christmas in Tulsa*. These episodes, along with Season 4, Episode 21: *The One with the Invitation*, which we removed, have the five lowest ratings (Figure 6.29). Hence, it would be interesting to consider removing these episodes, which do not accurately

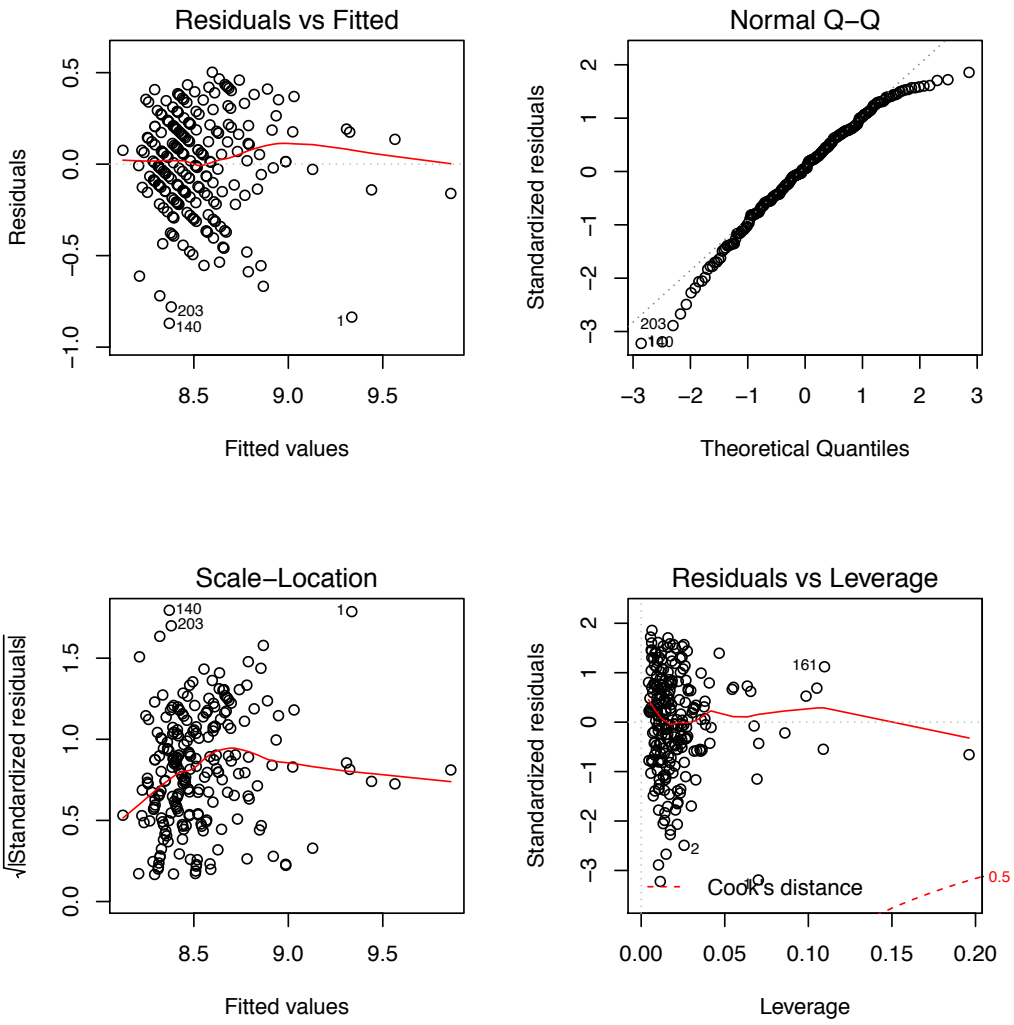



Figure 6.35: Diagnostic plots for the final model without datapoint 94: residual versus fitted scatterplot (top left), normal quantile plot of residuals (top right), scale-location plot (bottom left) and residuals versus leverage plot (bottom right).

represent many of the other episodes. Some flashback episodes, however, contain new material in flashback form, or the flashbacks only make up some of the episode. Hence, we would have to consider the degree to which each episode should be discounted for its flashbacks. For this analysis, however, we keep these episodes in the dataset and infer results using the final model

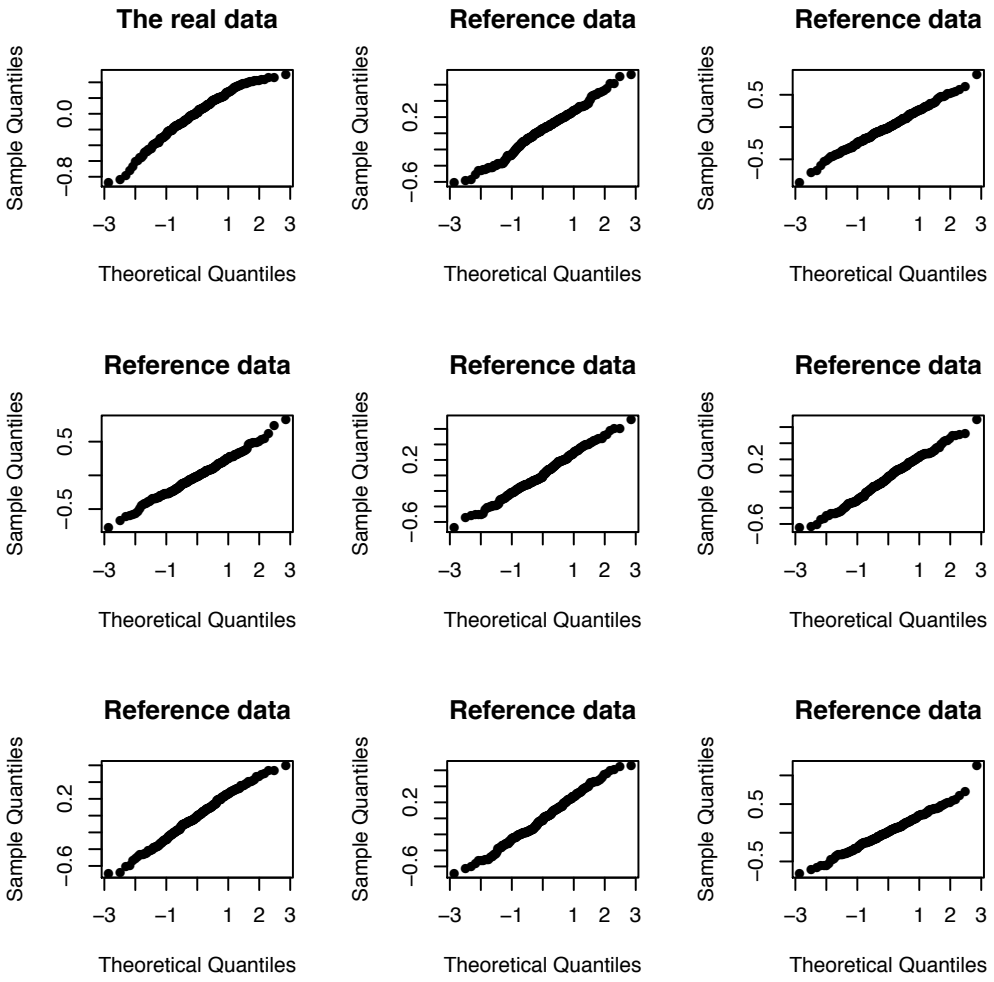



Figure 6.36: Normal quantile plot of residuals of the final model without datapoint 94 (The real data), and normal quantile plots of randomly generated normal data with the same mean, variance and number of datapoints as the residuals (Reference data).

without datapoint 94.

### 6.5.7 Discussion

The final model has four significant predictors for the rating of an episode of *Friends*. Figure 6.38 shows the effect of each of these predictors when all other predictors are held constant, with 95% confidence intervals for the predicted mean rating.

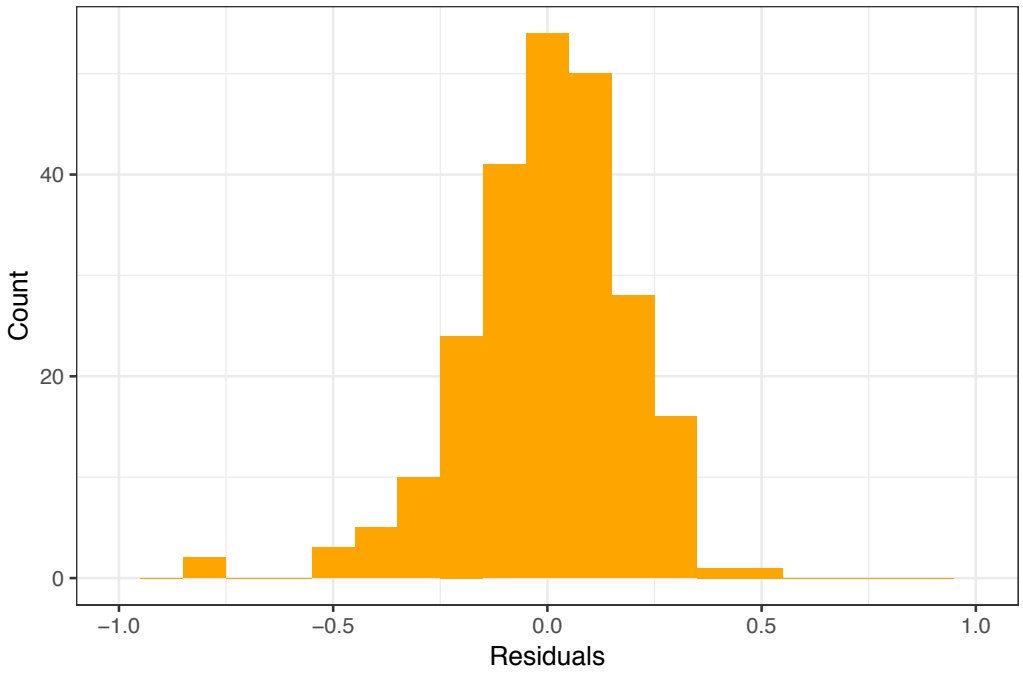



Figure 6.37: Histogram of residuals of the final model without datapoint 94.

### Number of raters

The predictor with the greatest effect on episode rating is the number of people who rated it. As the number of people rating the episode increases, the rating also increases, however, the opposite makes more sense. If an episode is particularly good, more viewers rate it. Recall that the transformed number of people rating the episode is on the log scale, so the number of raters increases exponentially compared to the rating. This is because the last episode, *The Last One: Part 2*, was rated by the most IMDb users, and this episode rated second highest.

### Number of lines

If we hold the clustering coefficient, density and number of people rating an episode constant, as the number of lines in an episode increases, the rating also increases. One explanation for this is that viewers prefer when characters have short, punctual lines, which means they can fit in more lines. In Brody's guide to writing good quality television scripts [32], he writes:

> "Good dialogue has a generally accepted definition. It's dialogue that is concise, witty, believable, and revealing of human

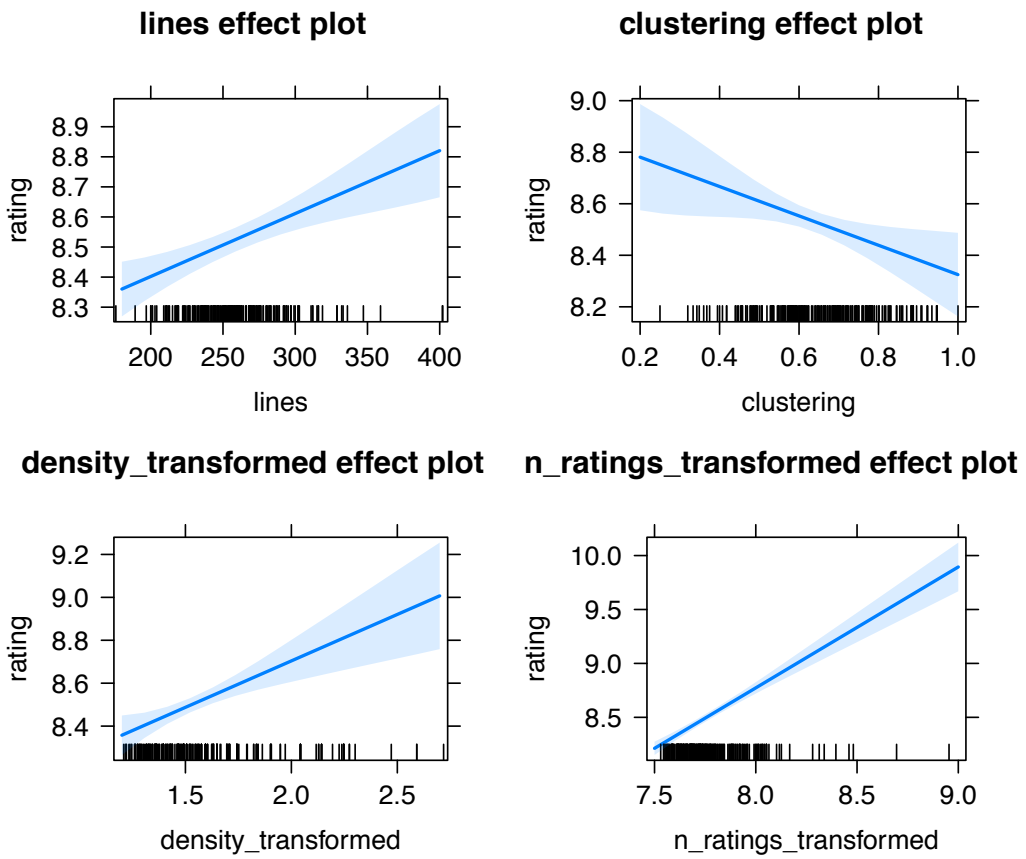



Figure 6.38: Line plots of the effect of the predictor variables `lines`, `clustering`, `density_transformed` and `n_ratings_transformed`, if all other predictors are held constant, on rating in the final multivariate model without datapoint 94. The shaded region represents the 95% confidence interval for the predicted mean rating.

character and emotion."

Audiences prefer television shows with concise, witty dialogue [99], and it appears as though *Friends* viewers prefer this in episodes with the series too. This theory is supported by a negative linear relationship between the average number of words per line and the episode's rating. The words per line was not included in the final model, however, because it is correlated with the number of lines. To further investigate the effect of the number of lines and length of lines on the success of television series, one could extract the number of words and lines from other television series and look at the relationship of these with episode ratings.

### Clustering and density

The clustering coefficient of an episode network has a negative effect on the IMDb rating of the episode, whereas the density has a positive effect. However, density and clustering are positively correlated. Figure 6.39 shows a scatterplot of the transformed density and clustering of the **manual** networks, coloured by the rating. As the exponential of the density increases, the clustering coefficient of the network also increases on average. However highly rated episodes require maximising the density and minimising the clustering.

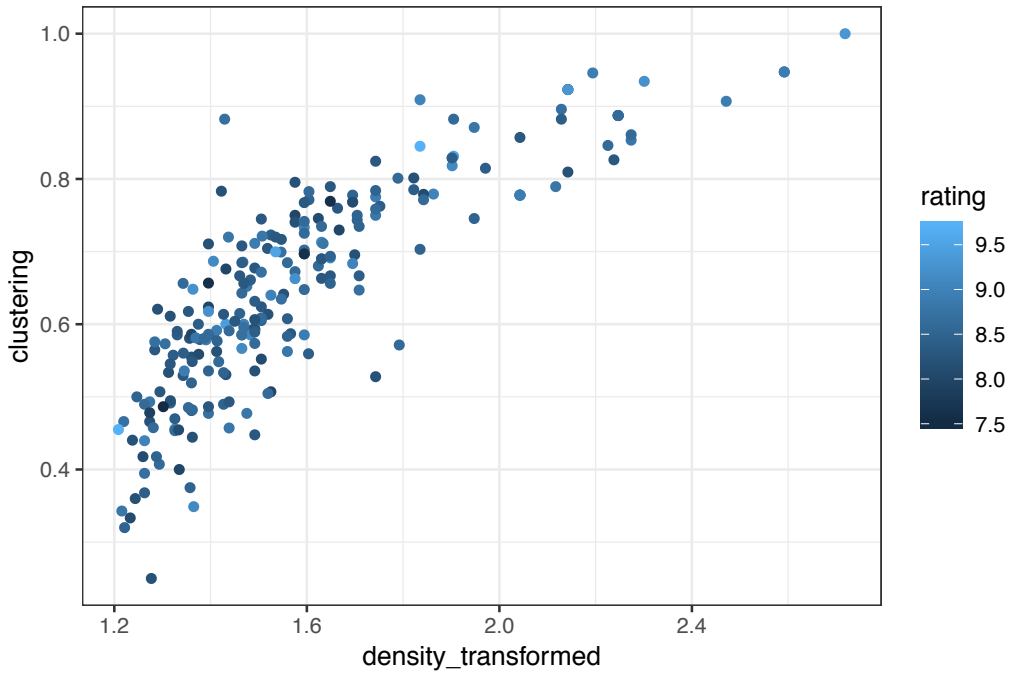
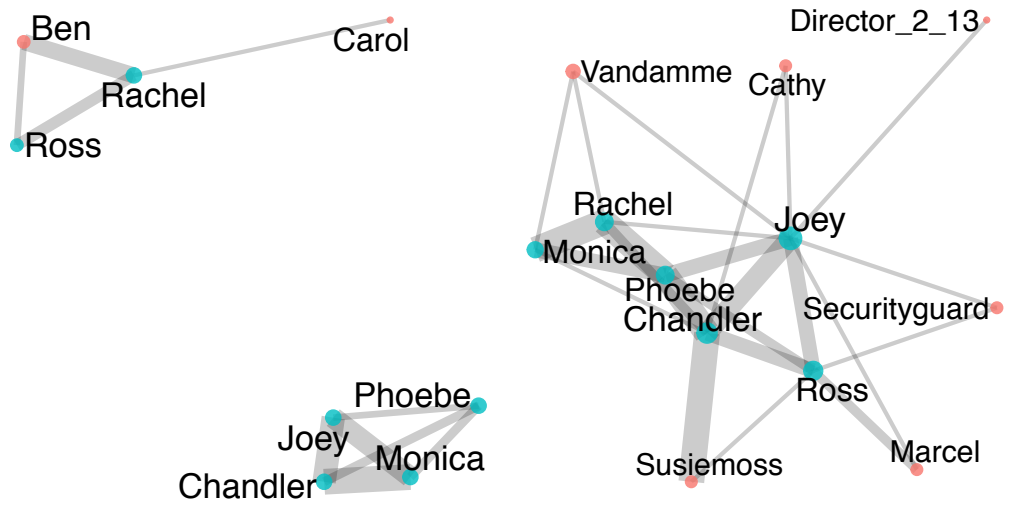



Figure 6.39: Scatterplot of exponentially transformed density (`density_transformed`) and clustering coefficient (`clustering`) of each episode of the **manual** networks, coloured by IMDb (`rating`).

To construct an episode with high density and low clustering, one could either fix the density and minimise clustering, or fix clustering and maximise density. Figure 6.40 shows the networks for two episodes with similar densities; Season 7, Episode 16: *The One with the Truth About London* (density of 0.357) and Season 2, Episode 13: *The One After the Superbowl: Part 2* (density of 0.364). The former episode has a high clustering coefficient of 0.882, and the latter has a low clustering coefficient of 0.457.




(a) High clustering network – Season 7, Episode 16: *The One with the Truth About London.*

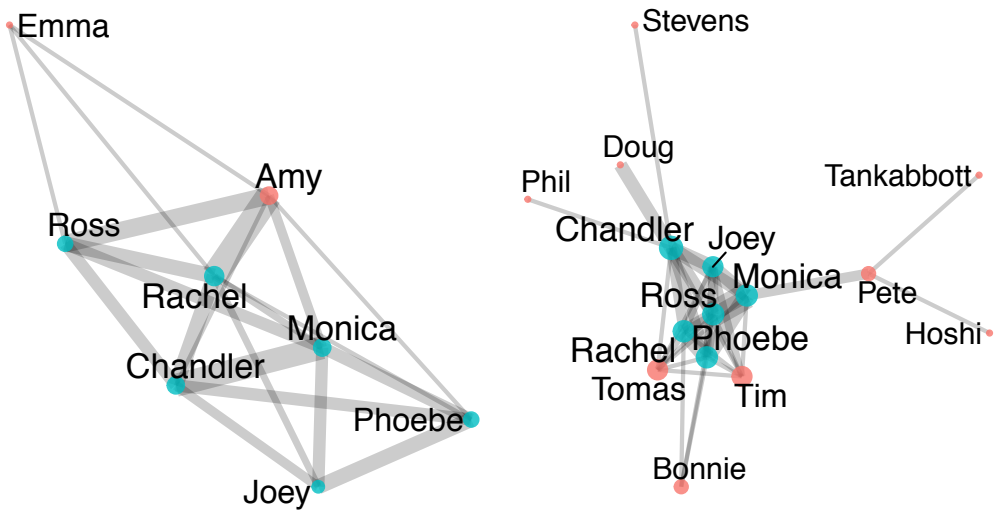
(b) Low clustering network – Season 2, Episode 13: *The One After the Superbowl: Part 2.*

Figure 6.40: **Manual** networks of *Friends* episodes with similar densities but different clustering coefficients. Panel (b) has low clustering and high density, so would be predicted to have a higher rating by our model.

Figure 6.41 shows the networks for two episodes with similar clustering coefficients; Season 9, Episode 8: *The One with Rachel's Other Sister* (clustering coefficient of 0.789) and Season 3, Episode 24: *The One with the Ultimate Fighting Champion* (clustering coefficient of 0.783). The former episode has a high density of 0.750, and the latter has a low density of 0.352.

Based on these networks, it appears as though viewers prefer episodes where all characters are connected in some way, even if it is through other characters, and we also enjoy episodes with minimal side characters, *i.e.*, there are few non-core characters, and they are closely connected with many of the core characters. Kevin S. Bright, one of the executive producers and director of *Friends*, remarked that viewers really enjoyed the episodes with just the core characters [41], which they call "bottle shows". In fact, the first episode with only core characters was made to save money on additional sets and guest actors, but it was so well-received that they decided to make more.

An interesting extension of this observation would be to find whether the social networks other television shows exhibit similar relationships with clustering and density and success, or whether this is unique to *Friends*.




(a) High density network – Season 9, Episode 8: *The One with Rachel's Other Sister*.

(b) Low density network – Season 3, Episode 24: *The One with the Ultimate Fighting Champion*.

Figure 6.41: **Manual** networks of *Friends* episodes with similar clustering coefficients but different densities. Panel (a) has high density and low clustering, so would be predicted to have a higher rating by our model.

## 6.6 Summary

In the first section of this chapter, we modelled the series, season and episode networks for the **co-occurrence** and **manual** dataset. We found that a two-class Poisson model fit many episode networks well, however the model underestimated the clustering coefficient. We discussed supervised nature of the grouping of the characters into two classes, which lead us into using stochastic block models to classify characters into more groups without depending on prior knowledge of the characters.

The stochastic block models allowed us to analyse the relationships between groups of characters, and which characters belong to each group. We found that for each season, the core characters were alone in a class, and there was a non-core inner group and a non-core outer group.

Next, we used linear models to find trends in the network metrics over time. We found that the number of words and lines spoken in each episode increases, while the total number of interactions decrease. We also used multivariate linear models to predict the rating of *Friends* episodes using the network metrics. The final ratings model suggests that the best episodes of *Friends* are the ones where the characters should have short, concise and

frequent lines, and the social network of character interactions are dense, but not too clustered.

With further analysis of other television shows, these factors could help television script writers to design more successful televisions shows. Furthermore, with social networks from different types of narratives, such as films and novels, we could investigate the differences in successful films, movies and television shows, and generalise our results to all types of narratives.

# Chapter 7

# Conclusion

"I'm not great at the advice... Can I interest you in a sarcastic comment?"

*Matthew Perry* as *Chandler Bing*
*Season 8, Episode 17*

## 7.1 Summary

Social networks are a powerful tool for narrative analysis. In this thesis, we compared techniques to extract social networks from narratives, and used time-evolving social networks to analyse the television series *Friends*. Our main contributions to the field of narrative social network analysis are our findings, that:

- co-occurrence network extraction, and other automated extraction methods provide efficient and reliable social networks for narrative analysis (Chapter 4);
- the *Friends* get less friendly over the course of the series (Chapter 5); and
- the ratings of *Friends* episodes are influenced by the number of lines, the clustering coefficient and the edge density of the social network (Chapter 6).

In more detail, the extraction of a social network from a narrative is a vital part of the analysis. One method is to manually record character interactions

while watching or reading the narrative. This "manual" method is accurate, but time consuming, so automated methods have been developed. We looked at two automated methods in detail; co-occurrence network extraction and extraction using natural language processing for identifying mentioned or recipient characters of dialogue. By modelling these techniques, and simulating social networks similar to those in episodes of *Friends*, we compared metrics of the simulated episode networks. We found that for many analyses, the extraction technique would not affect the results of the analysis. If one investigates the clustering of narrative social networks, however, the automated networks are unreliable, as they can distort the local and global clustering coefficients compared to the manually extracted network. On the other hand, analyses of the importance of relationships and characters in the narrative remain the same, irrespective of the technique used to extract the social network. Therefore, for many analyses, the use of automated social network extraction for narrative analysis is justified.

For further evidence, we analysed and compared two datasets of social networks from the television series *Friends*. For the **manual** dataset, Bazzan [21] watched all episodes of *Friends*, and recorded every interaction, *i.e.*, when characters talked, touched or looked at each other. In the **co-occurrence** dataset, we defined interactions as characters speaking in the same scene, and automatically extracted social networks using the scripts for *Friends*.

Through analysis of these networks, we made several interesting findings, which are consistent for both datasets. For instance, we found that, unsurprisingly, the six core characters; Chandler, Joey, Monica, Phoebe, Rachel and Ross are the most important characters in the series by far. Through character analysis in the social networks, we found that Chandler is the central character by many measures of centrality, and Phoebe is the least central of the core group. This is surprising, as both Chandler and Phoebe were originally meant to be side characters to the core group, but Chandler clearly did not end up that way. We also found that the famous intermittent relationship between Ross and Rachel, which was meant to be the central romance of the series, was not as prominent as the surprising relationship between Monica and Chandler.

We made further findings about *Friends* using network models and network metrics in linear models. We modelled the *Friends* networks using a two-class Poisson model for each episode and season of *Friends*, where the core characters made a class, but found that the model underestimated clustering at the episode level, and several metrics at the season level. This suggested that there is clustering in the social network that we did not account for by splitting the characters into two classes. We suggested a stochastic block model to split the characters into more classes, allowing for more

groups of families (*e.g.* Monica, Ross, Jack and Judy), partners (*e.g.* Carol and Susan) and work groups (*e.g.* Joey's costars). In particular, we found that in each season of both datasets, the characters tended to group into their roles, such as the core characters, who always formed a class, and long-term partners of the core characters, who formed another class.

Finally, we modelled network metrics over time and with ratings to find informative trends. The main findings of trends over time are that the total number of interactions between any pair of characters in a season or episode significantly decrease over time. This trend is noticeable in both datasets, at the episode and season level, so we infer that the *Friends* characters get less friendly. We also notice that the number of words and lines spoken per episode increase over the series, which is similar to some other television series (*e.g.*, *Seinfeld*), but dissimilar to others (*e.g.*, *The Walking Dead*, *The Office* and *The West Wing*).

We fit a multivariate linear model to the Internet Movie Database ratings for each episode using network metrics and features as predictor variables. We then used step-wise selection processes to find a final model to predict episode rating using network features. From the final model, we inferred that to create a highly rated episode social network, it would need to have high density, but low clustering coefficient. "Bottle episodes", which consist mainly of the core characters in a single place, fit this description. Viewers of *Friends* also prefer episodes with fewer lines.

## 7.2 Contribution to literature

This thesis contributes to the original literature in three main ways;

1. providing evidence that automated social network extraction methods are reliable for narrative analyses,

2. gaining deeper insights into a specific narrative – *Friends* – using social networks, and

3. developing the techniques for temporal narrative social network analysis and modelling.

First, we discuss and examine several narrative social network extraction techniques, and quantitatively compare three of the most used methods. Our results show that while manual extraction is the most accurate method, automated methods such as co-occurrence networks or using natural language processing usually achieve the same analysis results. As manual extraction is time consuming, these results mean that one can extract social networks

from narratives quickly and automatically without damaging the analysis. Then we can extract and analyse a large corpus of narratives using their social networks.

We analysed two types of social networks representing the characters and relationships from *Friends* in great detail, and so our results add to the literature on the series. We also made discoveries about the trends over time and episode ratings of *Friends*. With more research into other narratives, our results about *Friends* could lead to results about television shows or all narratives in general.

The last of the main contributions to the literature on narrative social networks was development of the narrative analysis methods. Narrative analysis in the literature using social networks usually involve calculating global or character metrics, and describing or comparing the networks and characters. We performed this analysis in detail on the *Friends* social network, but also modelled the temporal network for a deeper understanding of the characters and their social structure. We introduced the use of stochastic block models for narrative analysis, and fit linear models to the network metrics over time and with ratings. These steps provided more information about the *Friends* series, and prove to be effective methods for narrative analysis.

## 7.3 Future research

While our research contributes to the literature in several ways, it also provides a platform for future research. In particular, we suggest ways to improve on our models and analysis throughout the thesis. We also discuss unanswered questions that were touched on in the thesis.

In the comparison of extraction techniques in Chapter 4, we use a two-class Poisson model, but we later find that this model underestimates clustering. It is not clear that the extraction technique comparison relies heavily on this model, but a more complicated, but better fitting model could be of use. We also assumed a constant number of interactions in each scene. One could investigate the effect of changing parameters of the model, such as the number of scenes, number of interactions and error rate of the natural language processing. It is also of interest to consider other network extraction techniques, and differences in directed or undirected, and weighted or unweighted networks.

When modelling the social networks of *Friends* characters, we noticed the two-class Poisson model underestimated the clustering coefficient. We suggested some alternate models, such as the preferential attachment model, or the small-world model, however these model unweighted, one-class networks,

so the models would have to be adapted to fit the *Friends* networks. The stochastic block models provided insights into the classes and character roles of *Friends* characters, and an interesting extension is to fit the same model to other narrative networks. This leads to questions of the form, "Do the same models fit all narratives, or all narratives of a certain type?"

Our comparison of narrative social network extraction techniques showed that narrative networks extracted automatically are as reliable networks extracted manually for most analyses. One could use this information to extract a large corpus of narrative social networks (*e.g.*, from Project Gutenberg [3], which provides thousands of novels for free online), and apply our techniques for narrative analysis to achieve general results about narratives. For example, what do narratives of particular genres or medias have in common? Do the same network attributes cause higher success in all narratives? And do all narratives exhibit similar patterns to those we see in *Friends* (*e.g.*, the characters getting less "friendly")? And if they do, why? In Chapter 6 we suggested that the number of interactions in the series decreases because the storylines become more complex once we are familiar with the characters. One could measure the complexity of a storyline using information theory entropy measures on the words in the narrative (*i.e.*, the script if the narrative is a television series). Much previous work has gone into methods for measuring complexity of text [70], including some analyses of narratives [87, 102, 122].

In particular, we found in Chapter 6 that the number of words and lines spoken by characters in *Friends* decreases over time. We hypothesised that this is because of language changes between the years *Friends* aired: 1994 – 2004. To test this theory, one could perform similar analysis on other television series that aired at similar times. For example, *Frasier* (1993 – 2004), *Dawson's Creek* (1998 – 2003), *Rugrats* (1991 – 2004), *Sex and the City* (1998 – 2004), *Buffy the Vampire Slayer* (1997 – 2003), and *Sabrina the Teenage Witch* (1996 – 2003) aired over similar years to *Friends*.

Further narrative analysis will allow us to understand different cultures through the stories they tell, and different periods of time through the narratives that were created and were popular at different times. Our research provides a building block for improving and understanding the narratives that surround us.

# Appendix A

# Figures and Tables

## A.1 Betweenness centrality core character ranks

Figure A.1 shows the betweenness centrality ranks for all core characters, analogous to Figure 4.7 in the main text. Although there are minor differences in many of the ranks from the co-occurrence networks compared to the manual networks, large differences are rare. Therefore analysis of characters using betweenness centrality is not greatly affected by the network extraction method. For example, the spike in Chandler's betweenness centrality rank in Season 9 is present in both datasets, as well as Joey's constantly high betweenness rank.

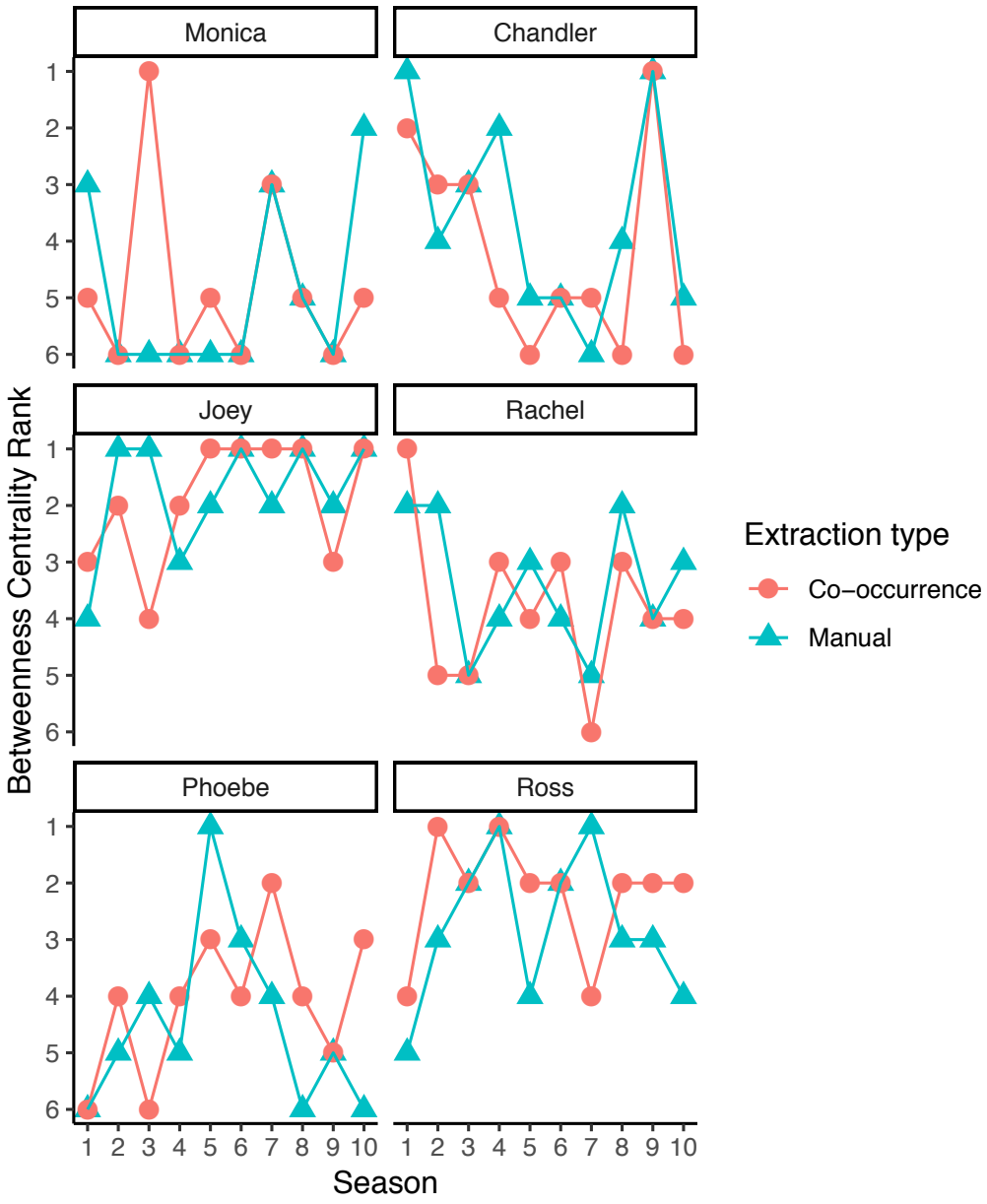



Figure A.1: Betweenness centrality ranks of each core character over every season for the co-occurrence and manual datasets.

## A.2 Season view *Friends* networks

Figures A.2 – A.19 show the social networks for Seasons 2 to 10 of *Friends*, for the **co-occurrence** and **manual** datasets. The networks for Season 1 are Figure 5.3 and Figure 5.4 in the main text.

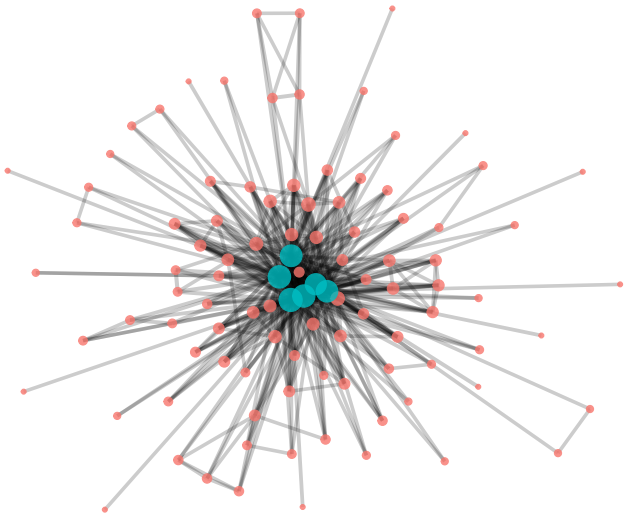


Figure A.2: Season 2 network for the **co-occurrence** dataset.

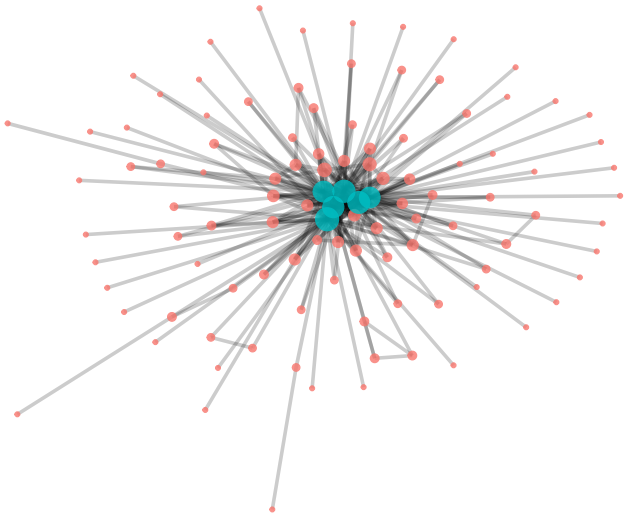


Figure A.3: Season 2 network for the **manual** dataset.

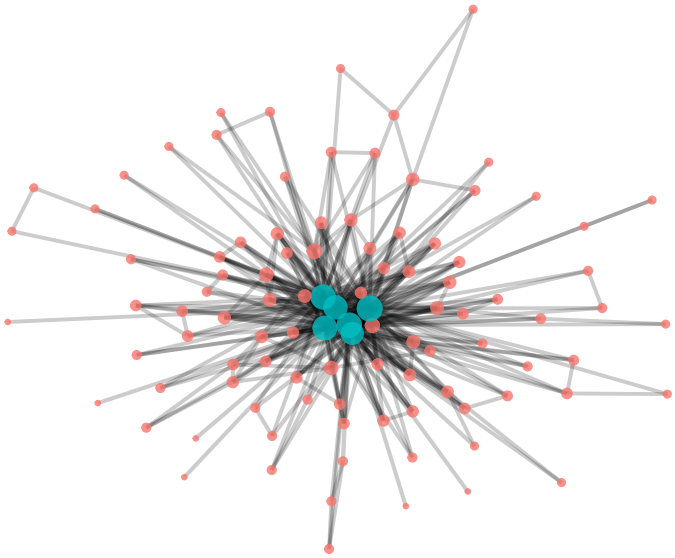


Figure A.4: Season 3 network for the **co-occurrence** dataset.

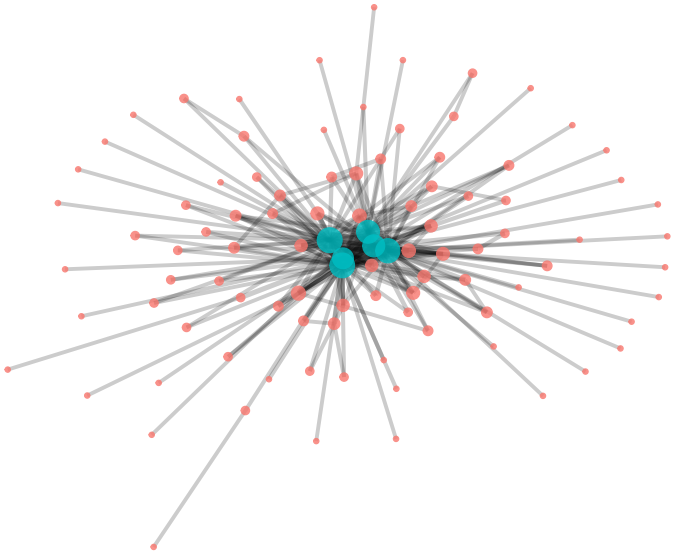


Figure A.5: Season 3 network for the **manual** dataset.

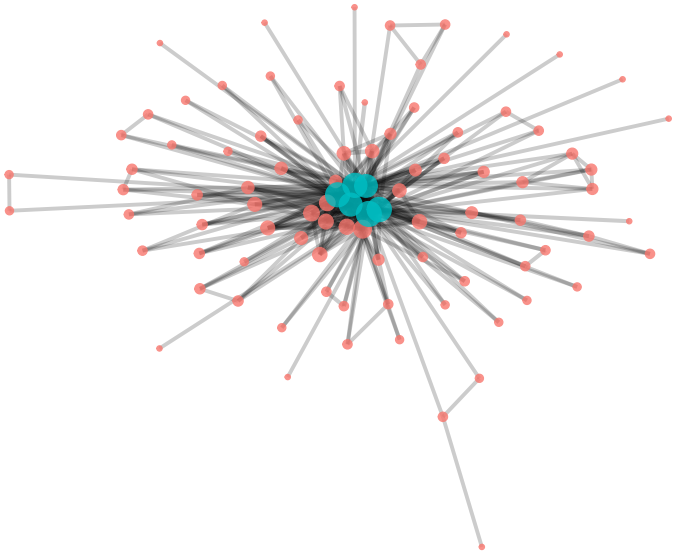


Figure A.6: Season 4 network for the **co-occurrence** dataset.

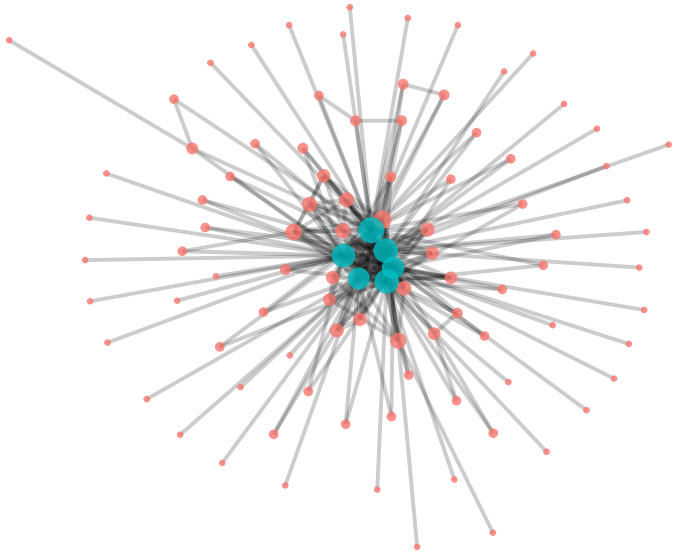


Figure A.7: Season 4 network for the **manual** dataset.

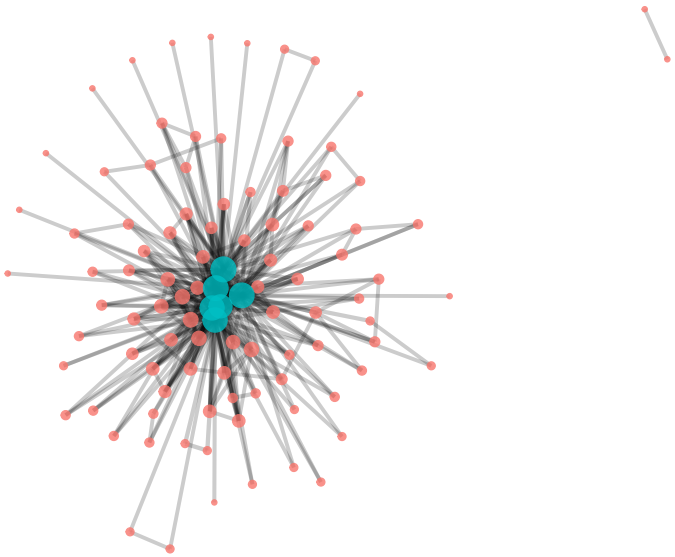


Figure A.8: Season 5 network for the **co-occurrence** dataset.

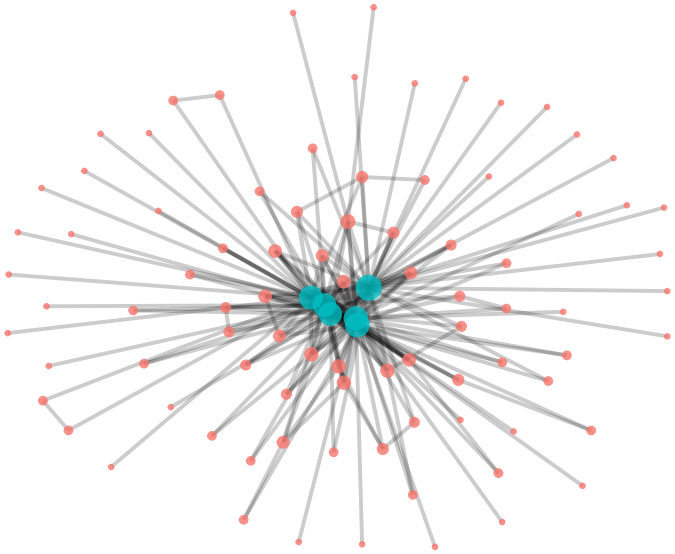


Figure A.9: Season 5 network for the **manual** dataset.

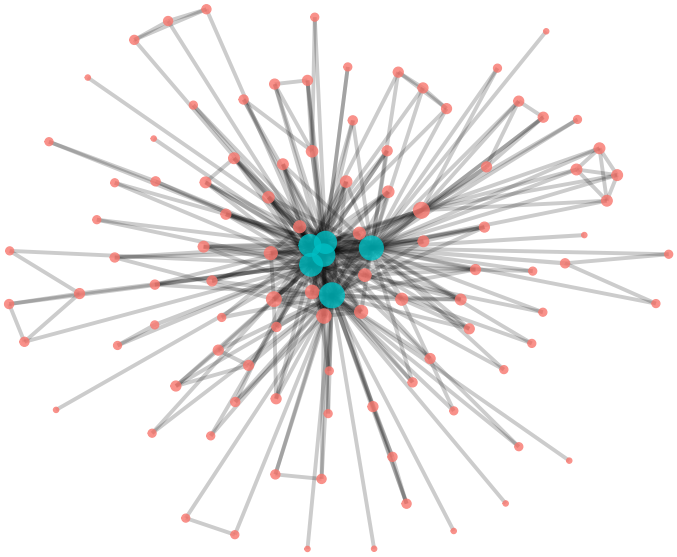


Figure A.10: Season 6 network for the **co-occurrence** dataset.

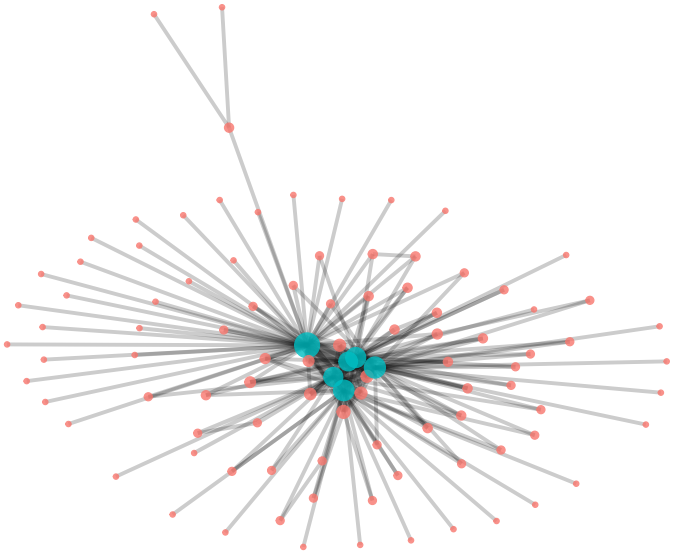


Figure A.11: Season 6 network for the **manual** dataset.

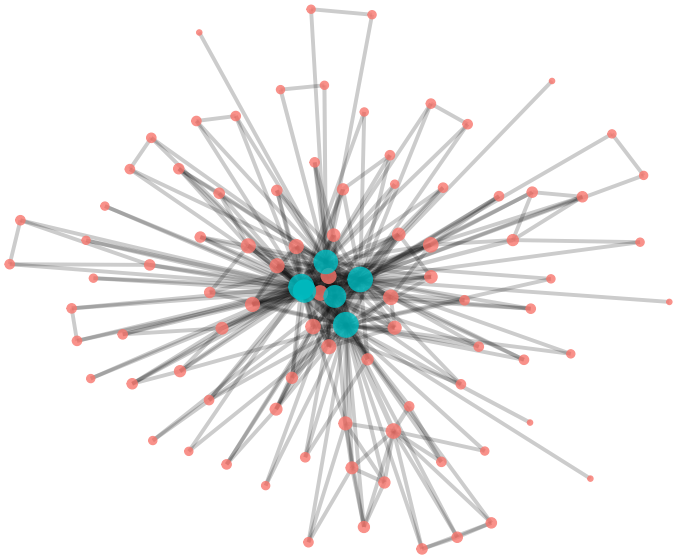


Figure A.12: Season 7 network for the **co-occurrence** dataset.

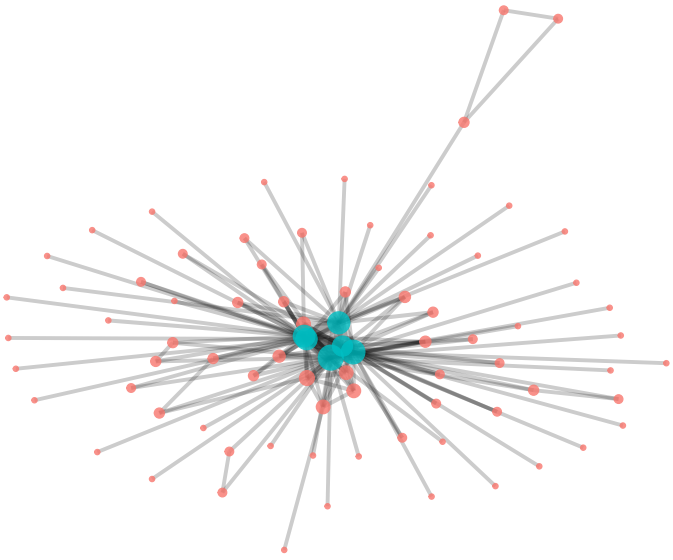


Figure A.13: Season 7 network for the **manual** dataset.

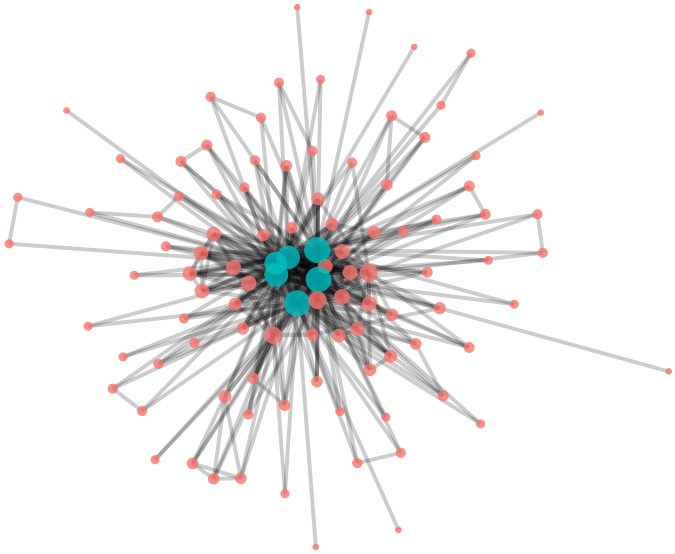


Figure A.14: Season 8 network for the **co-occurrence** dataset.

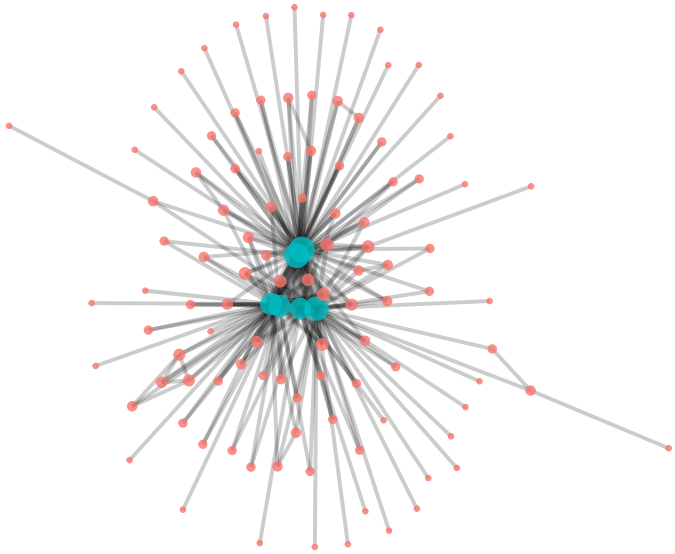


Figure A.15: Season 8 network for the **manual** dataset.

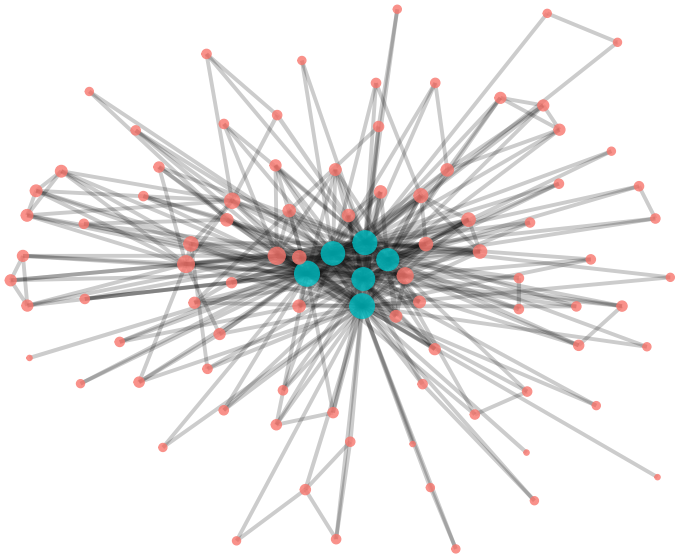


Figure A.16: Season 9 network for the **co-occurrence** dataset.

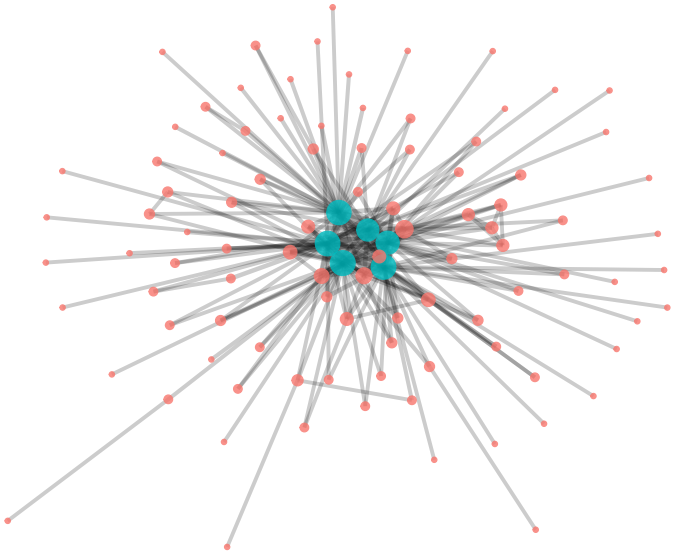


Figure A.17: Season 9 network for the **manual** dataset.

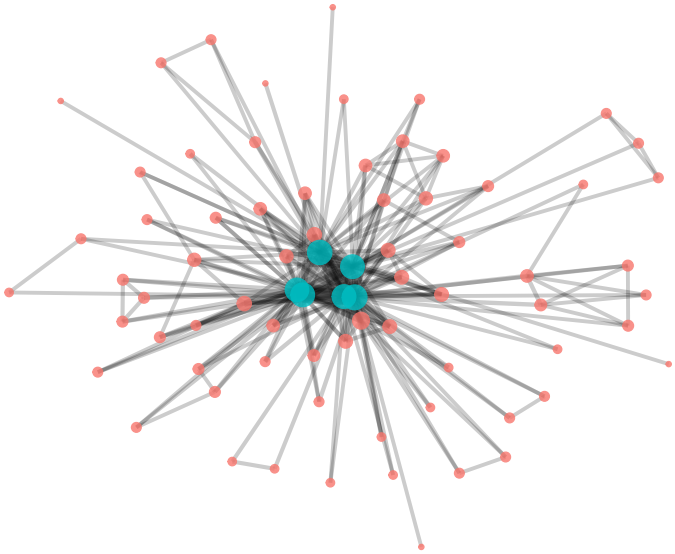


Figure A.18: Season 10 network for the **co-occurrence** dataset.

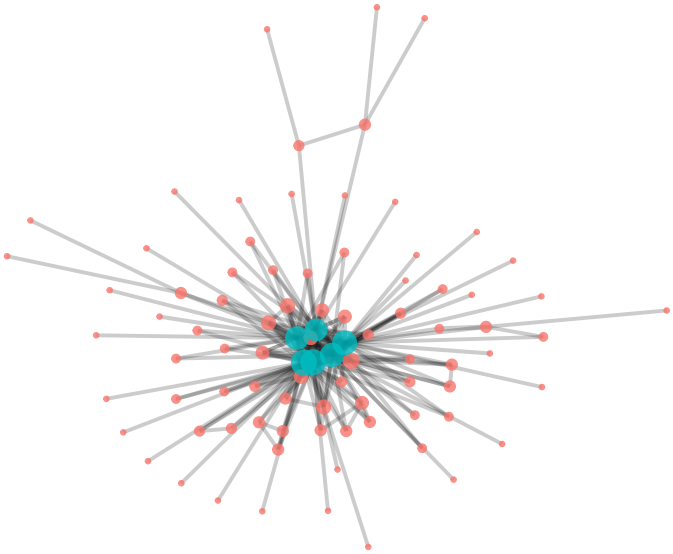


Figure A.19: Season 10 network for the **manual** dataset.

## A.3 Core character metric boxplots for manual networks

Figure A.20 shows boxplots of the character metrics for the six core characters in the **manual** season networks, and Figure A.21 shows boxplots of the character metrics for the core characters in the **manual** episode networks. These are similar to Figure 5.12 and Figure 5.13 in the main text, but are calculated for the **manual** networks instead of the **co-occurrence** networks.

While there are some differences in the exact values of the metrics over the two datasets, the analysis from the boxplots is almost identical. This supports the idea that the extraction method has little effect on the analysis of the narrative through its social network.

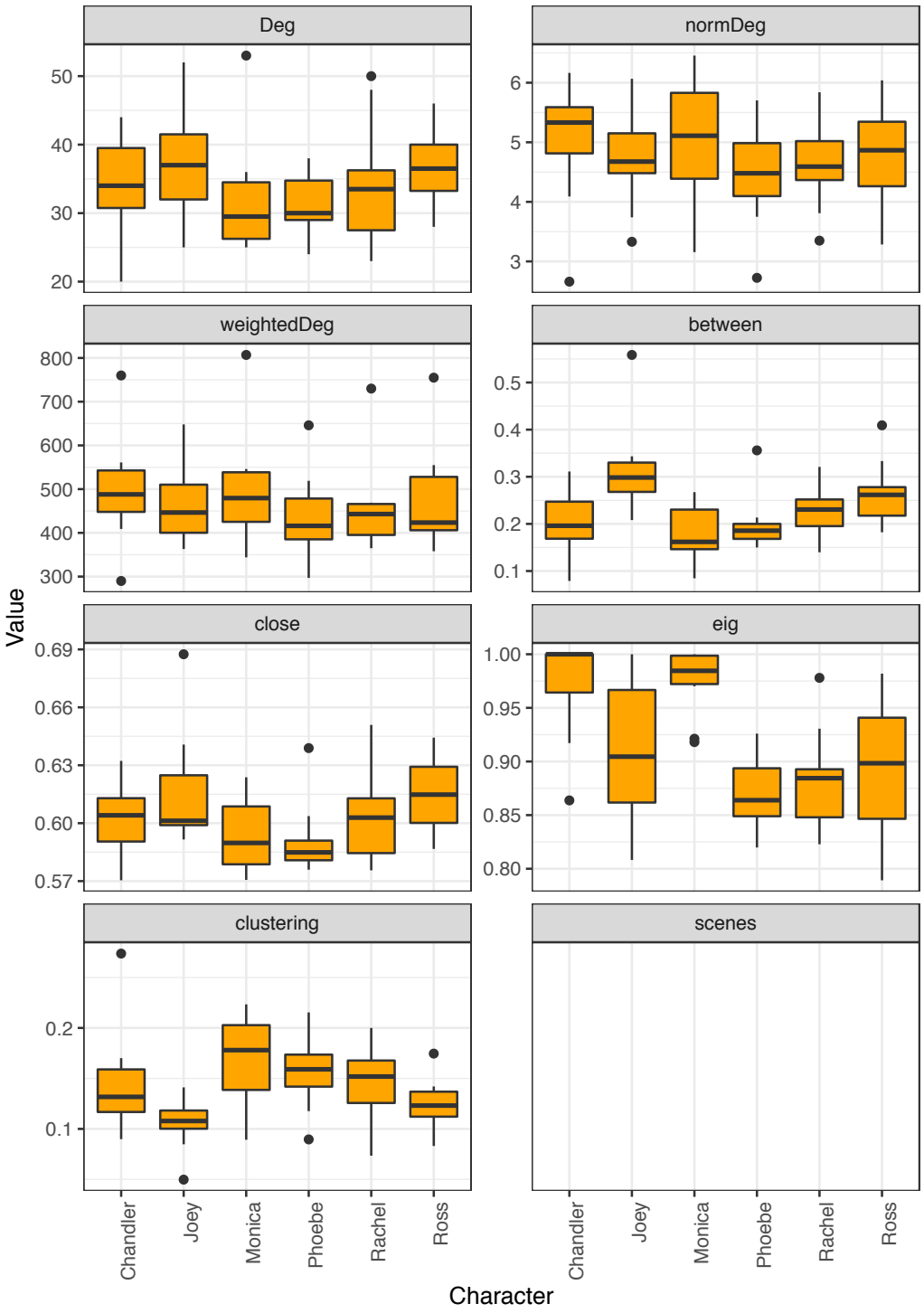



Figure A.20: Boxplots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen) and local clustering coefficient (clustering) for the six core characters in the **manual** season networks. Note that we have no data for the proportion of scenes (scenes) in the **manual** networks.

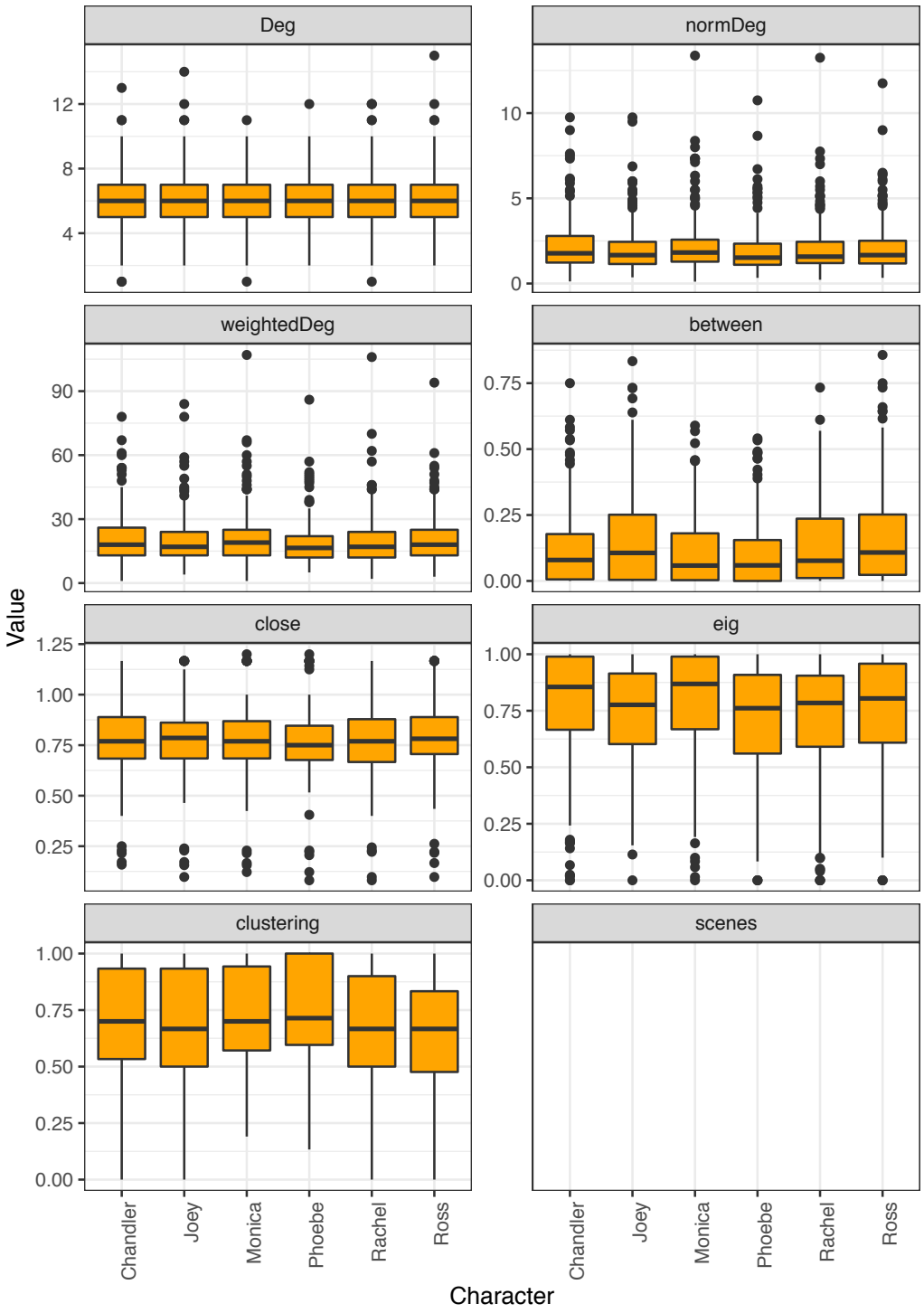



Figure A.21: Boxplots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen) and local clustering coefficient (clustering) for the six core characters in the **manual** episode networks. Note that we have no data for the proportion of scenes (scenes) in the **manual** networks.

## A.4 Closeness centralities in largest connected component

Figure A.22 shows the closeness centralities of the six core characters in both the **co-occurrence** and **manual** series networks, similarly to Figure 5.10 in the main text. However, in the main text, the closeness centrality calculates the path length from each node to every other node. If there is no path, the size of the network is used. In the **co-occurrence** series network, this results in an underestimation of the closeness centralities of characters, as there is more than one component. Here, we recalculate the closeness centralities, but only use the largest component. We find the closeness centralities for each character is now larger in the **co-occurrence** network than in the **manual** network, which is not surprising, as there are more connections, so many path lengths are shorter.

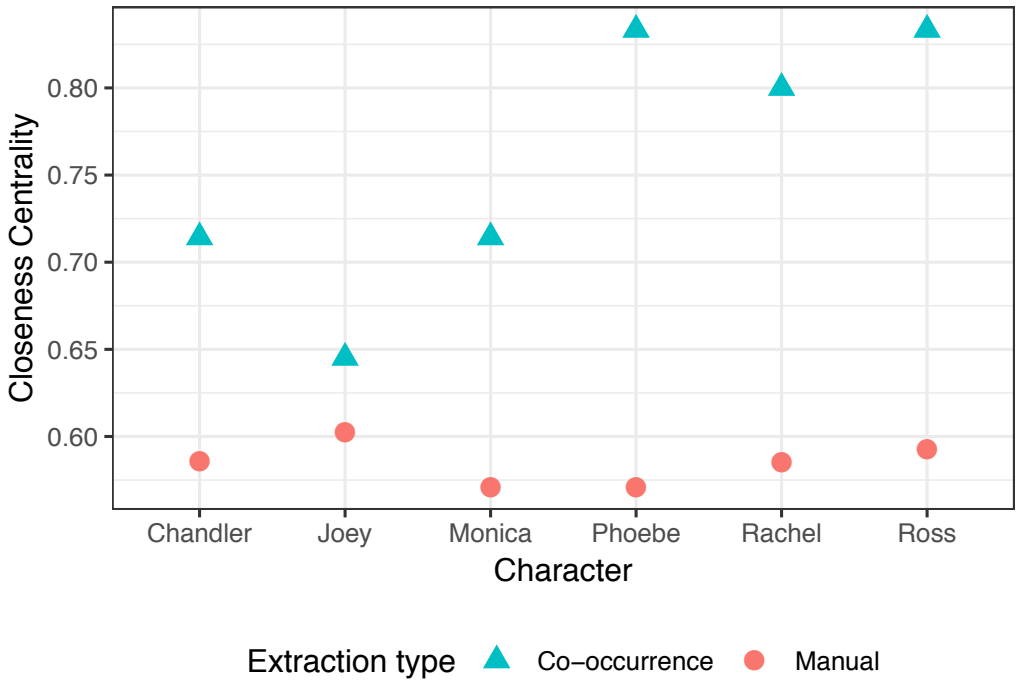



Figure A.22: Closeness centralities for the six core characters in the largest connected components of the **co-occurrence** (blue) and **manual** (red) series networks.

Figure A.23 shows the closeness centralities of the six core characters in both the **co-occurrence** and **manual** series networks, analogous to Fig-

ure 5.12 in the main text. For some seasons, only calculating paths to characters in the largest component increases the closeness centrality of all the characters. We can now see that Ross has the highest closeness centrality median for the **co-occurrence** season networks. Note that the boxplots for core character closeness centralities in the **co-occurrence** season networks now look very similar to the closeness centralities in the **manual** season networks (Figure A.20).

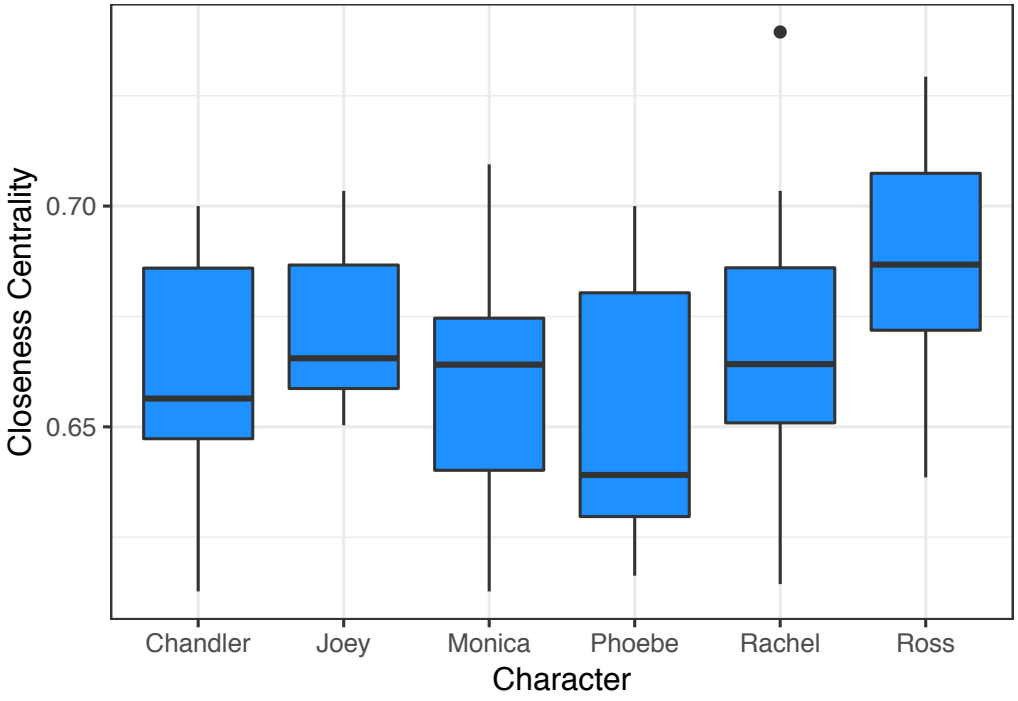



Figure A.23: Boxplots of closeness centralities for the six core characters in the largest connected components of the **co-occurrence** season networks.

Figure A.24 is analogous to Figure 5.23 in the main text, showing the closeness centralities of the six core characters in the **co-occurrence** season networks over time. For Seasons 2, 3 and 5, which have more than one component, the closeness centralities increase to similar values to the other seasons. We can now see trends in the closeness centralities of characters, and find that the patterns are similar to the degree of the characters.

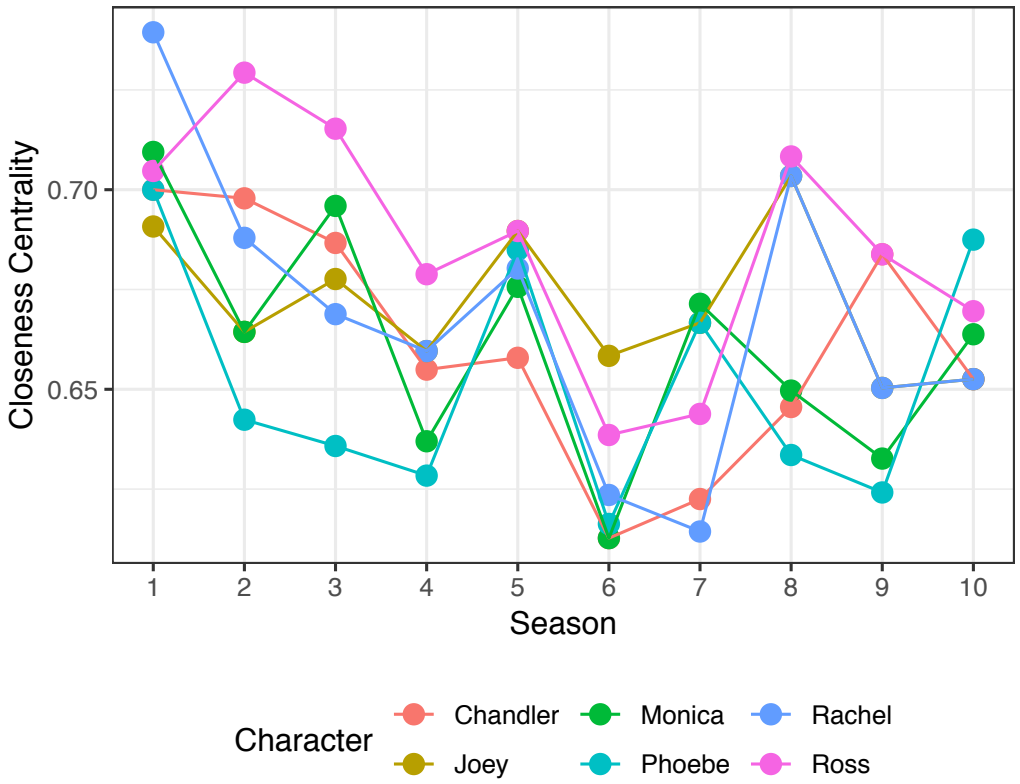



Figure A.24: Closeness centralities for the six core characters in the largest connected components of the **co-occurrence** season networks over time.

## A.5 Network metrics with "woman" removed

Table A.1 is analogous to Table 5.2 in the main text, with an extra column showing the metrics for the **co-occurrence** series network with the node "woman" removed. The analysis would be identical without the node "woman".

| | Co-occurrence | No Woman | Manual |
|---|---|---|---|
| size | 660 | 659 | 746 |
| totalEW | 18528 | 18327 | 16569 |
| totalE | 2632 | 2572 | 1609 |
| avEW | 7.04 | 7.13 | 10.30 |
| density | 0.00605 | 0.00593 | 0.00290 |
| avDeg | 7.98 | 7.81 | 4.31 |
| avPath | 2.27 | 2.28 | 2.59 |
| diameter | 4 | 4 | 5 |
| clustering | 0.0551 | 0.0528 | 0.0335 |
| clique | 13 | 13 | 10 |

Table A.1: Table of global metrics: size, total edge weight (totalEW), number of edges (totalE), average edge weight (avEW), density, average degree (avDeg), average path length (avPath), diameter, clustering coefficient (clustering) and size of the largest clique (clique) for the co-occurrence and manually extracted static networks.

Figure A.25 is analogous to Figure 5.10 in the main text. It shows scatterplots of the character metrics for the core characters in the series networks for the **co-occurrence** dataset, the **manual** dataset, and the **co-occurrence** dataset with the node "woman" removed. The analysis would be identical without the node "woman".

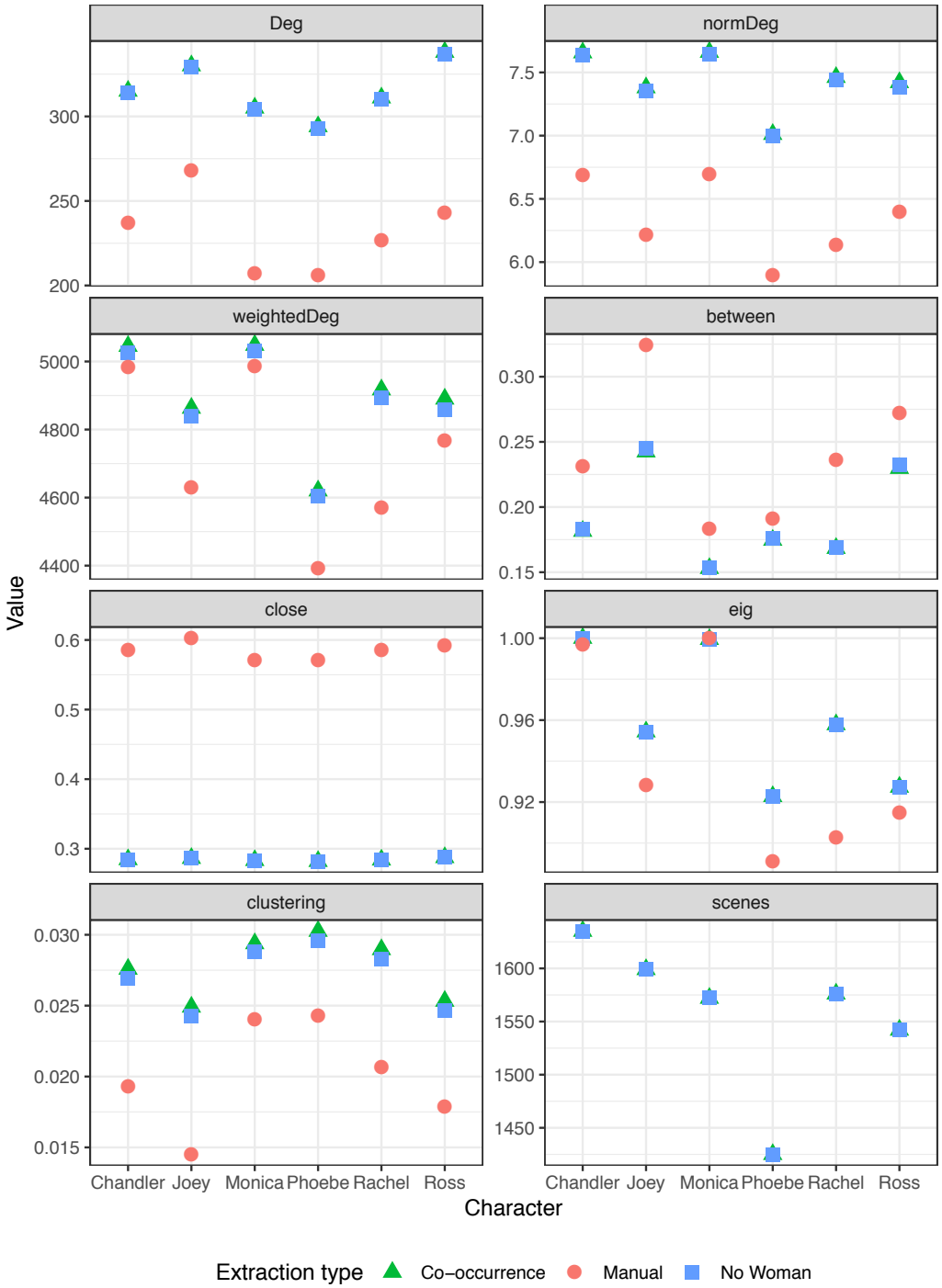



Figure A.25: Scatterplots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen), local clustering coefficient (clustering) and proportion of scenes (scenes) for the six core characters in the **co-occurrence** (green), **manual** (red), and **co-occurrence** with "woman" removed (blue) series network.

## A.6 Manual dataset season edge weights

Figure A.26 shows the boxplots of edge weights between every core character pair for the **manual** season networks. This is analogous to Figure 5.19 in the main text, which shows the edge weights for the core characters in the **co-occurrence** season networks. There are some differences between the two datasets, but the key features are present in both.

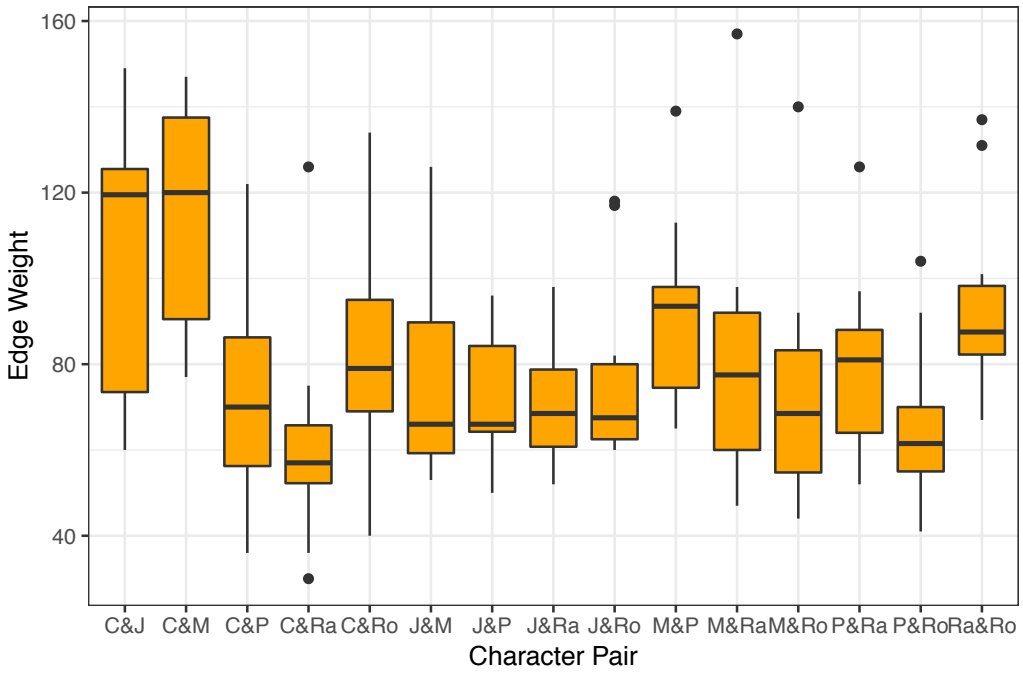



Figure A.26: Boxplots of edge weights between core characters (C = Chandler, J = Joey, M = Monica, P = Phoebe, Ra = Rachel and Ro = Ross) for the **manual** season networks. The edge weight is equivalent to the number of interactions between the characters in the season.

## A.7 Manual dataset episode edge weights

Figure A.27 shows the boxplots of edge weights between every core character pair for the **manual** episode networks. This is analogous to Figure 5.20 in the main text, which shows the edge weights for the core characters in the **co-occurrence** episode networks. There are some differences between the two datasets, but the key features are present in both.

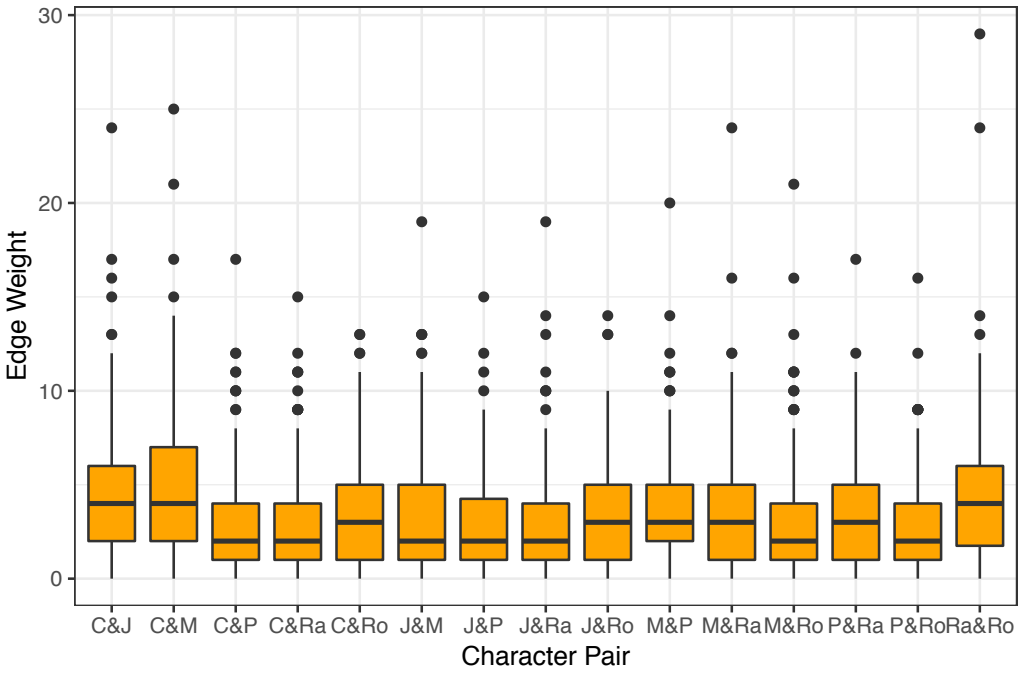



Figure A.27: Boxplots of edge weights between core characters (C = Chandler, J = Joey, M = Monica, P = Phoebe, Ra = Rachel and Ro = Ross) for the **manual** episode networks. The edge weight is equivalent to the number of interactions between the characters in the episode.

## A.8 Manual dataset episode bivariate character metrics

Figure A.28 shows the character metrics for the six core characters in the **manual** season networks over time. This is analogous to Figure 5.23 in the main text, which shows the character metrics for the six core characters in the **co-occurrence** season networks over time. Here, we list some key features that are in the **manual** dataset, but not the **co-occurrence** dataset:

- peak in Joey's degree in Season 6,
- peak in Joey's closeness and betweenness centrality in Season 6
- peak in Chandler's clustering coefficient in Season 7.

Bazzan discussed these features in her analysis of the **manual** *Friends* network [21]. Apart from these, the character metrics are quite similar in the two datasets.

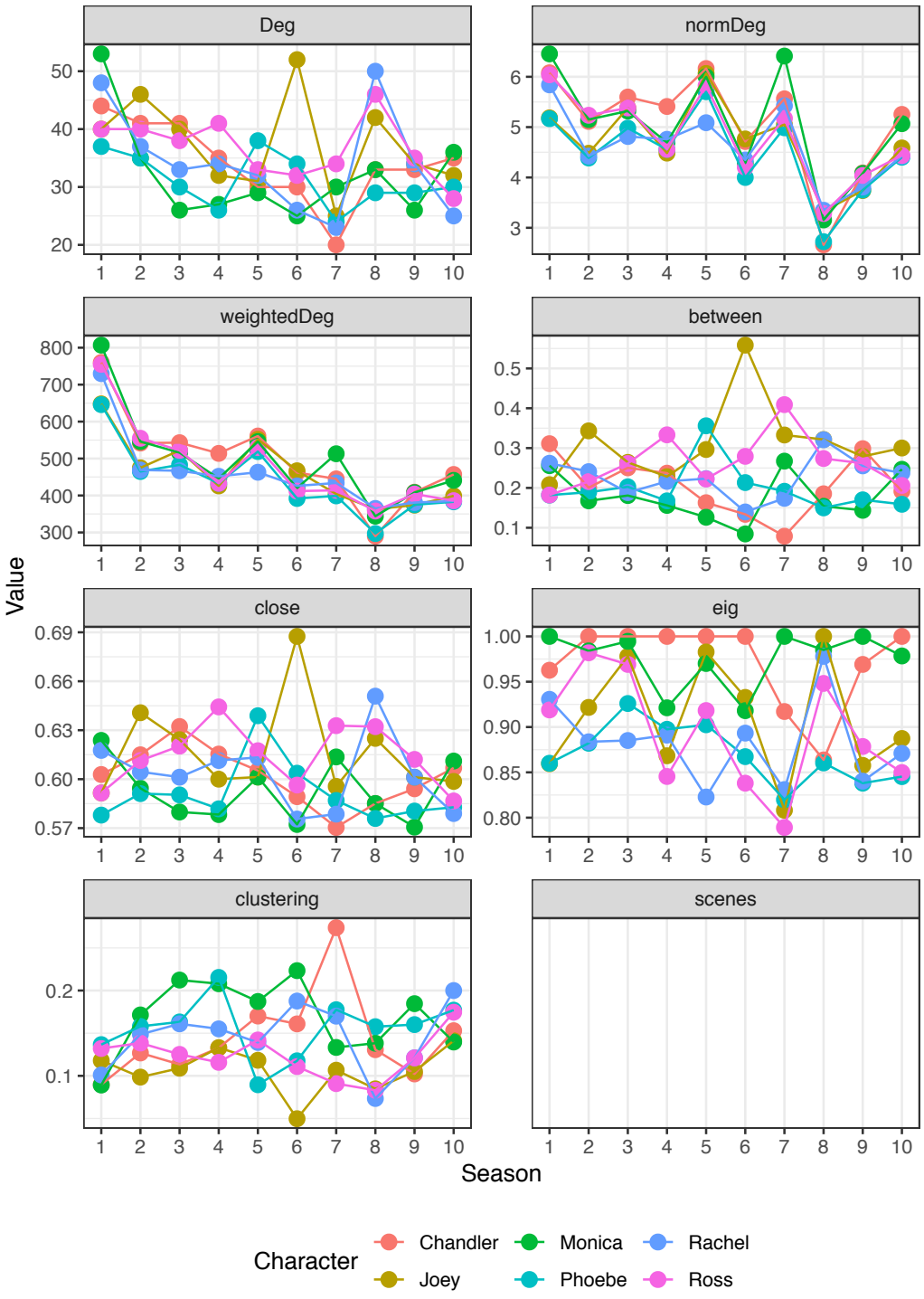



Figure A.28: Scatterplots of character metrics: degree (Deg), normalised weighted degree (normDeg), weighted degree (weightedDeg), betweenness centrality (between), closeness centrality (close), eigenvector centrality (eigen) and local clustering coefficient (clustering) for the six core characters in the **manual** season networks over time. Note that we have no data for the proportion of scenes (scenes) in the *manual* networks.

## A.9 Core relationship edge weights

Figures A.29 to A.33 show scatterplots of the edge weights between Joey, Monica, Phoebe, Rachel and Ross, respectively, and the other core characters over time in the **co-occurrence** dataset. Figure 5.24 in the main text shows shows a scatterplot of the edge weights between Chandler and the other core characters over time in the **co-occurrence** dataset.

Recall that Figure 5.24 shows that Chandler was closest to Joey in the first four seasons, but Monica's relationship with Chandler takes over in the remaining six seasons.

In Figure A.29, we notice that Joey is closest to Chandler until Season 6, when he becomes closer to Rachel. Once he stops interacting with Chandler so much, he interacts less with all characters.

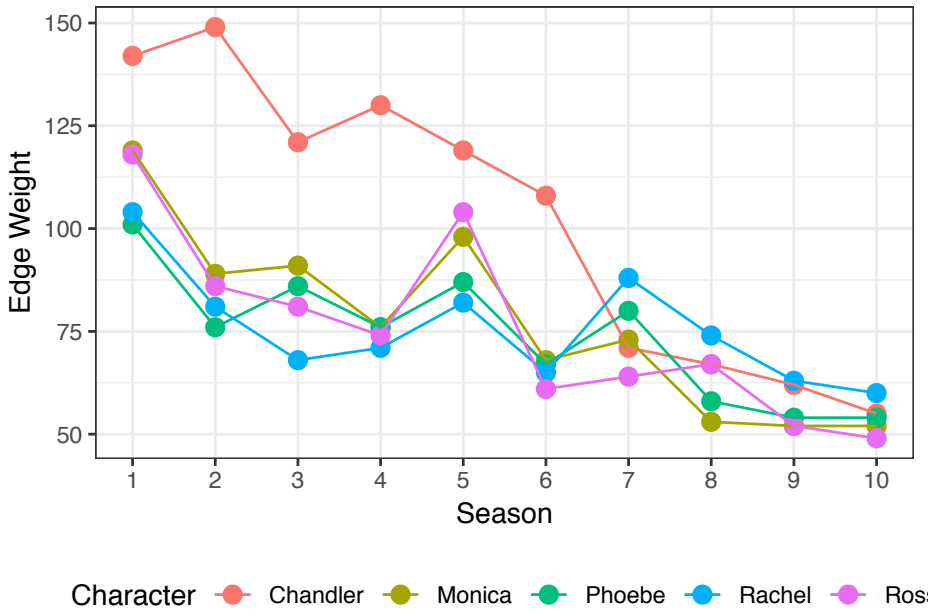



Figure A.29: Scatterplot of the edge weight of Joey with the five other core characters in the **co-occurrence** season networks over time.

In Figure A.30 show that Monica interacted similarly with all the core characters, until her and Chandler started dating in Season 5. From Season 5 onwards, her interactions with Chandler remain strong, but her interactions with the other core characters decrease.

In Figure A.31, we see that Phoebe is usually closest to Monica, even though Monica was closer to other characters than she is to Phoebe. This

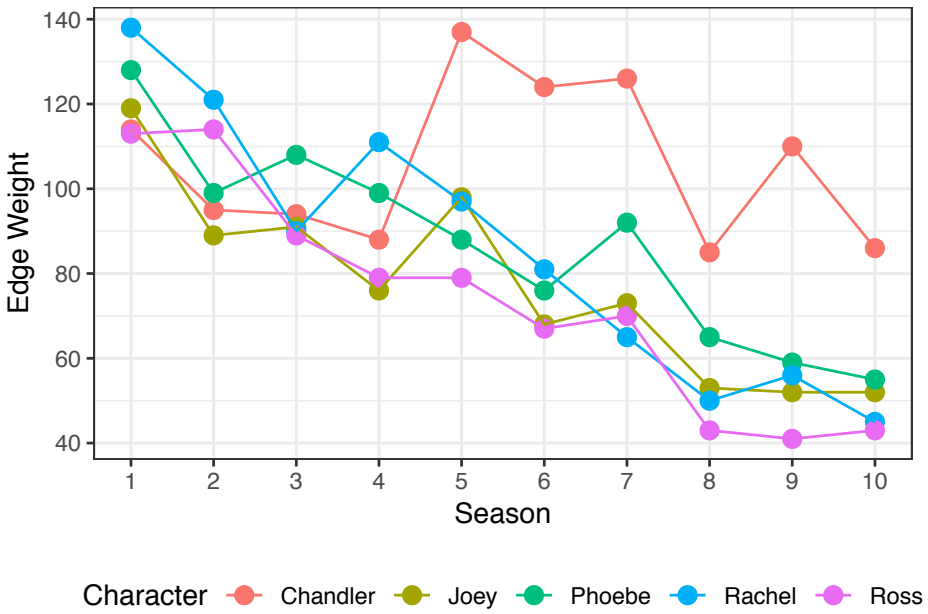



Figure A.30: Scatterplot of the edge weight of Monica with the five other core characters in the **co-occurrence** season networks over time.

supports the theory that Phoebe is the least important character out of the core characters.

Figure A.32 show that Rachel was close to many different characters, depending on the season. In Seasons 1, 4 and 5, while she is roommates with Monica (and not dating Ross), she is closest to Monica. In Seasons 6 and 7 when she lives with Phoebe, she is closest to Phoebe. In every other season, she is closest to Ross, which is in line with Rachel and Ross's intermittent relationship.

Ross, however, is solely focussed on Rachel for most of the series. Figure A.33 shows that he is closest to Rachel by far in Seasons 2, 3, 4, 6, 7, 8 and 10. In the other seasons, Rachel ranks close to the top in interactions with Ross.

In all figures we see a general trend of the edge weight/number of interactions in the season decreasing as the series develops. We see this pattern several times throughout the thesis.

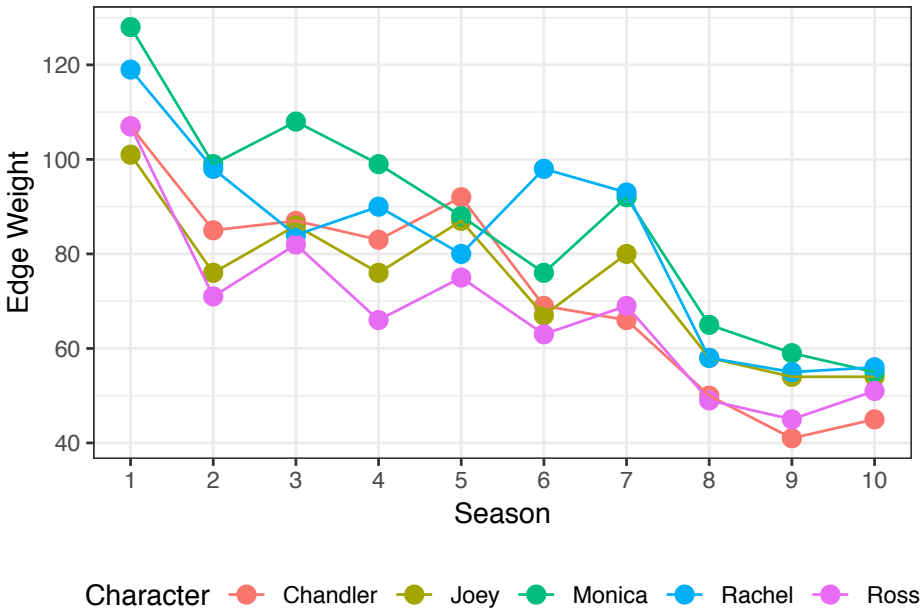



Figure A.31: Scatterplot of the edge weight of Phoebe with the five other core characters in the **co-occurrence** season networks over time.

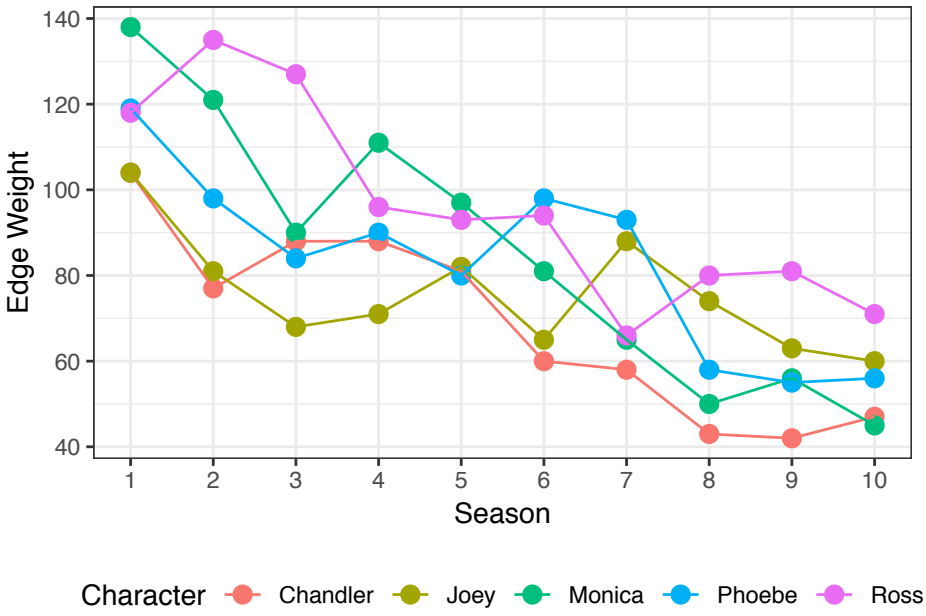



Figure A.32: Scatterplot of the edge weight of Rachel with the five other core characters in the **co-occurrence** season networks over time.

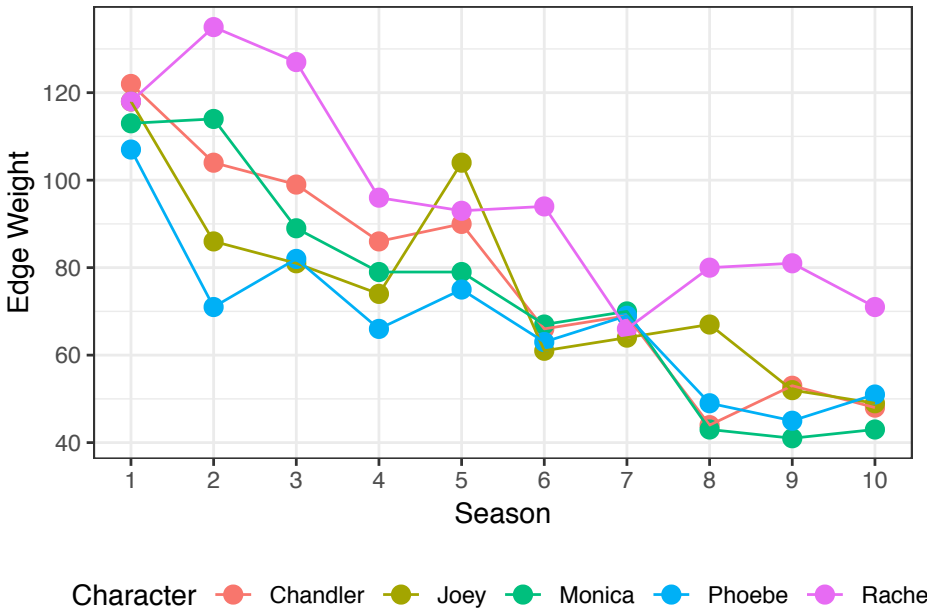



Figure A.33: Scatterplot of the edge weight of Ross with the five other core characters in the **co-occurrence** season networks over time.

## A.10 Two-class Poisson model simulations

Figure A.34 shows histograms of metrics for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 1, Episode 16: *The One with Two Parts: Part 1*, and Figure A.35 shows histograms of metrics for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 6, Episode 9: *The One Where Ross Got High*.

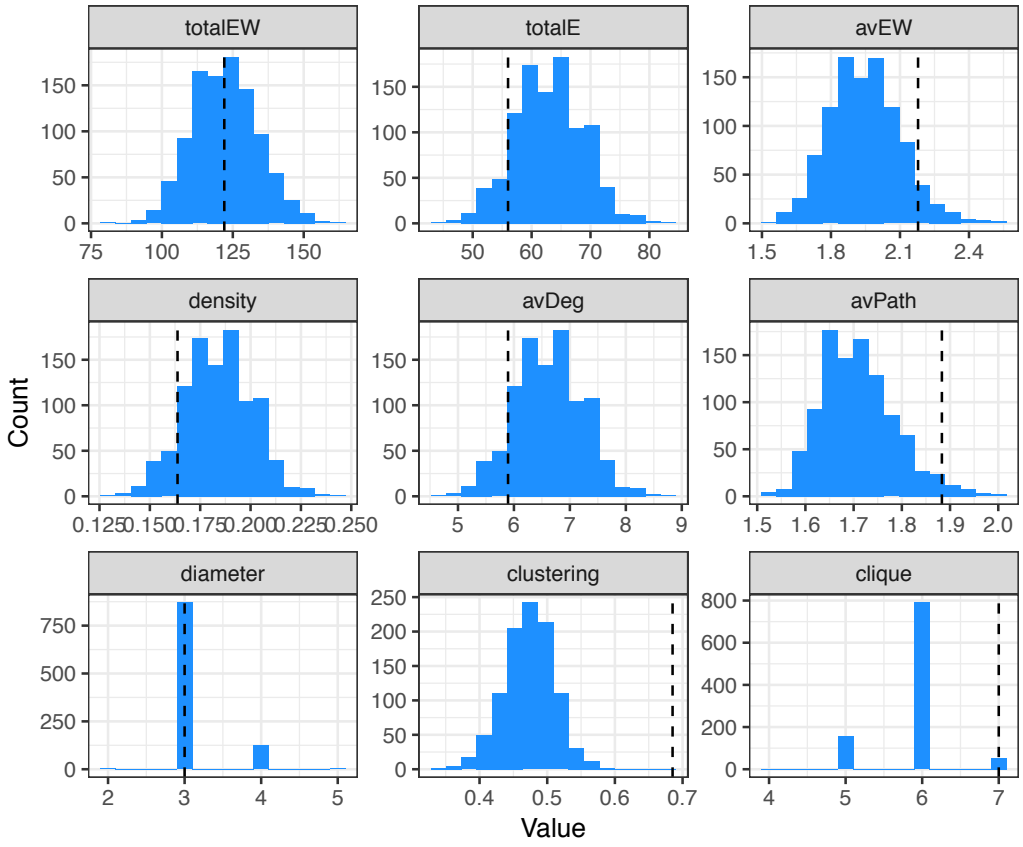



Figure A.34: Histograms of metrics (total edge weight: totalEW, total number of edges: totalE, average edge weight: avEW, density, average degree: avDeg, average path length: avPath, diameter, clustering coefficient: clustering, and size of the largest clique: clique) for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 1, Episode 16: *The One with Two Parts: Part 1*. The dotted black line represents the metric of the actual network for episode.

For both sets of parameters, and the set of parameters used in the main

text (see Figure 6.6), the two-class Poisson model achieves reasonable networks, except for the clustering. In both figures, the simulations underestimate the clustering in the episode networks.

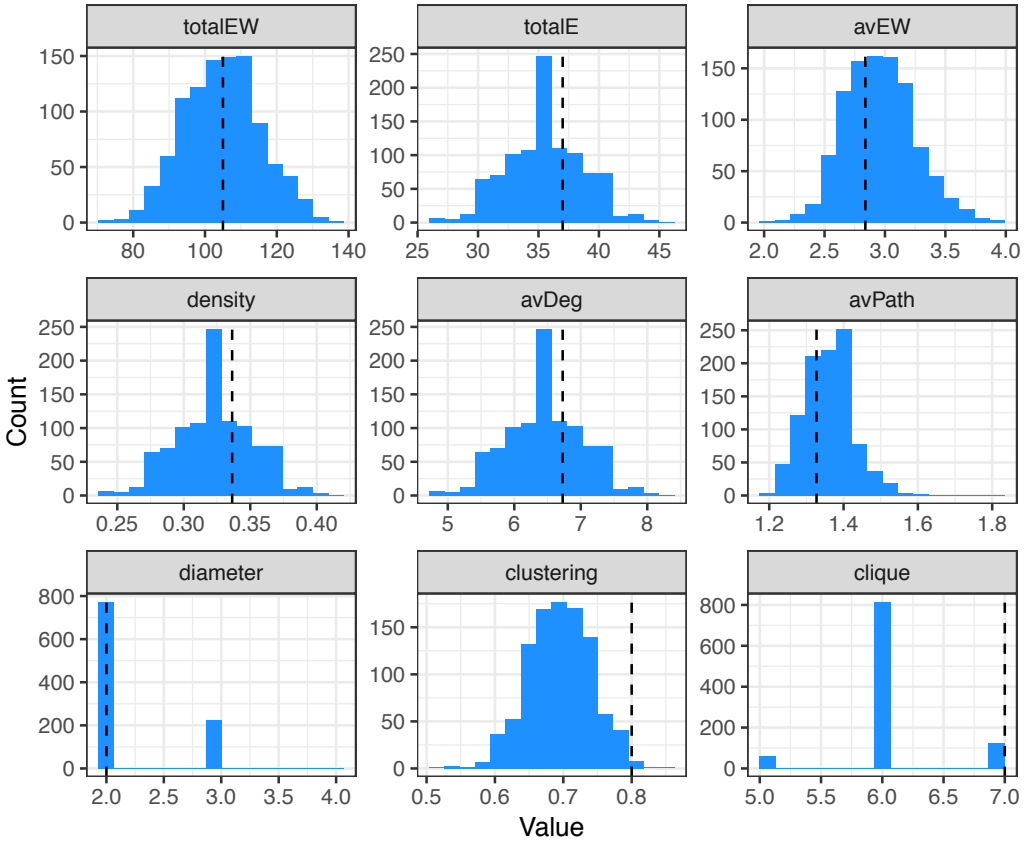



Figure A.35: Histograms of metrics (total edge weight: totalEW, total number of edges: totalE, average edge weight: avEW, density, average degree: avDeg, average path length: avPath, diameter, clustering coefficient: clustering, and size of the largest clique: clique) for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 6, Episode 9: *The One Where Ross Got High*. The dotted black line represents the metric of the actual network for episode.

Figure A.36 shows histograms of metrics for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 2. This is analogous to Figure 6.7 in the main text, which shows histograms of metrics for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 1.

For both sets of parameters, the two-class Poission model under or over-

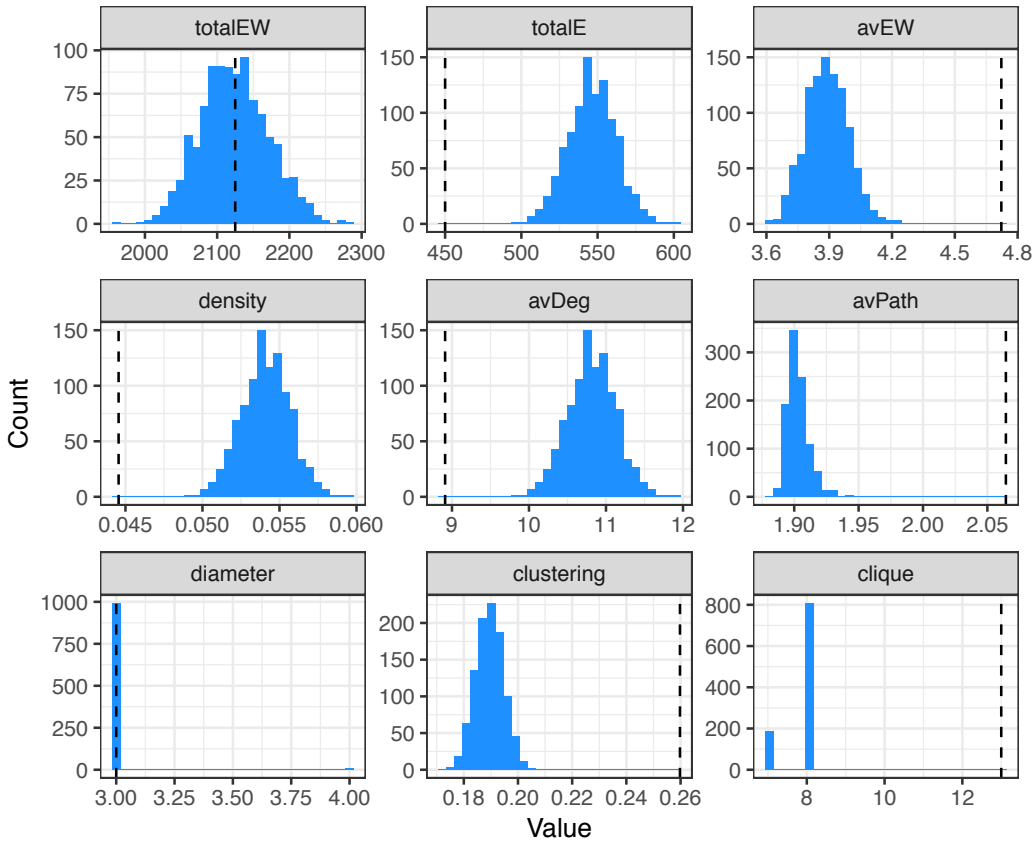



Figure A.36: Histograms of metrics (total edge weight: totalEW, total number of edges: totalE, average edge weight: avEW, density, average degree: avDeg, average path length: avPath, diameter, clustering coefficient: clustering, and size of the largest clique: clique) for 1000 simulations of episodes with the estimated parameters as the **co-occurrence** network for Season 2. The dotted black line represents the metric of the actual network for episode.

estimates many of the metrics. Therefore the model does not fit the *Friends* season networks well.

## A.11 Stochastic block model co-occurrence season networks

Figures A.37 to A.45 show the **co-occurrence** season networks as classified into classes by the stochastic block model, along with heatmaps that represent the interaction rate within and between classes. The similar figure for Season 1 in Figure 6.12 in the main text.

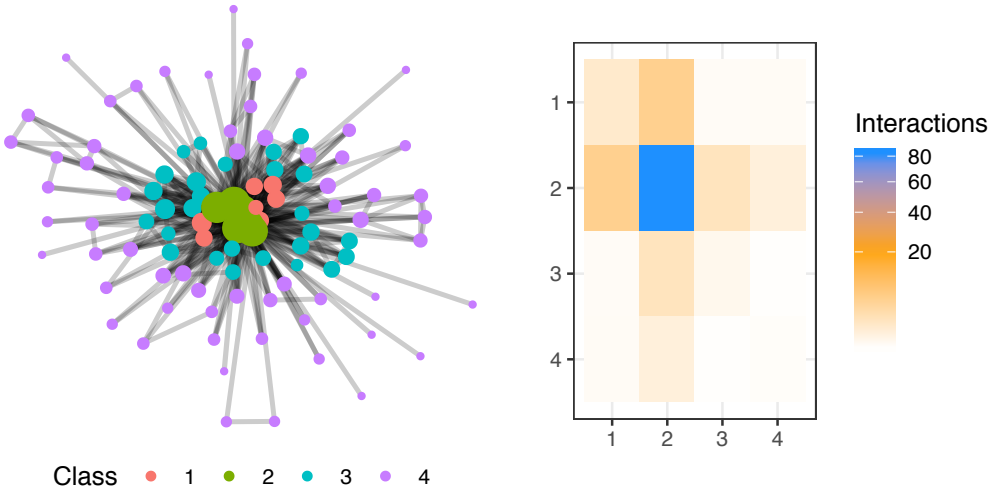



Figure A.37: Season 2 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

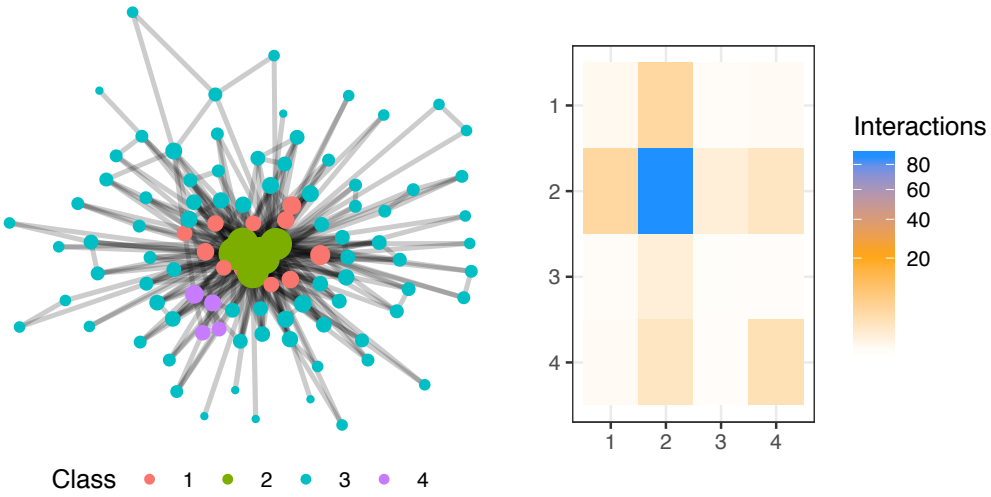



Figure A.38: Season 3 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

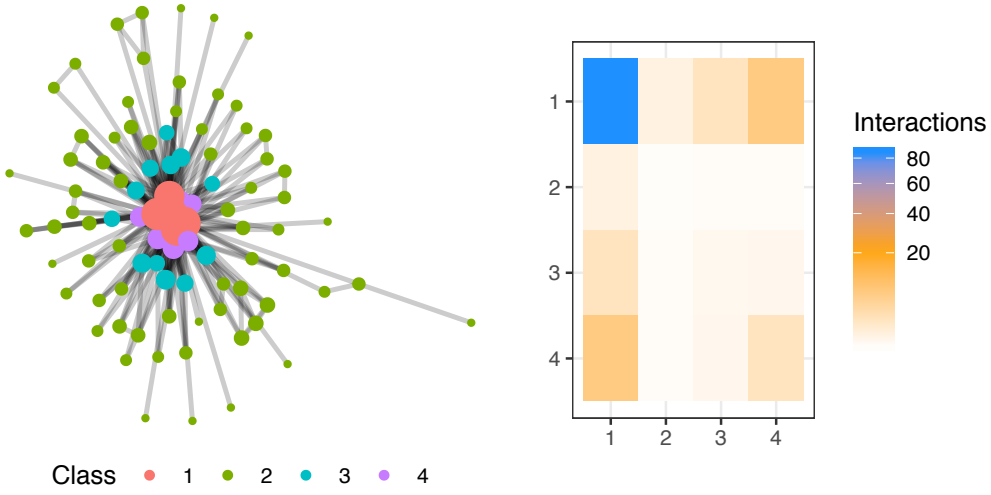



Figure A.39: Season 4 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

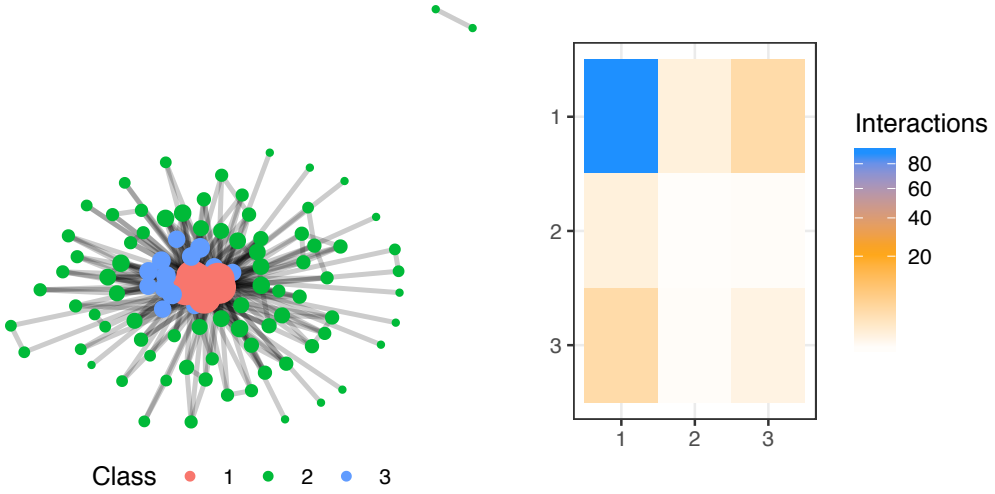



Figure A.40: Season 5 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

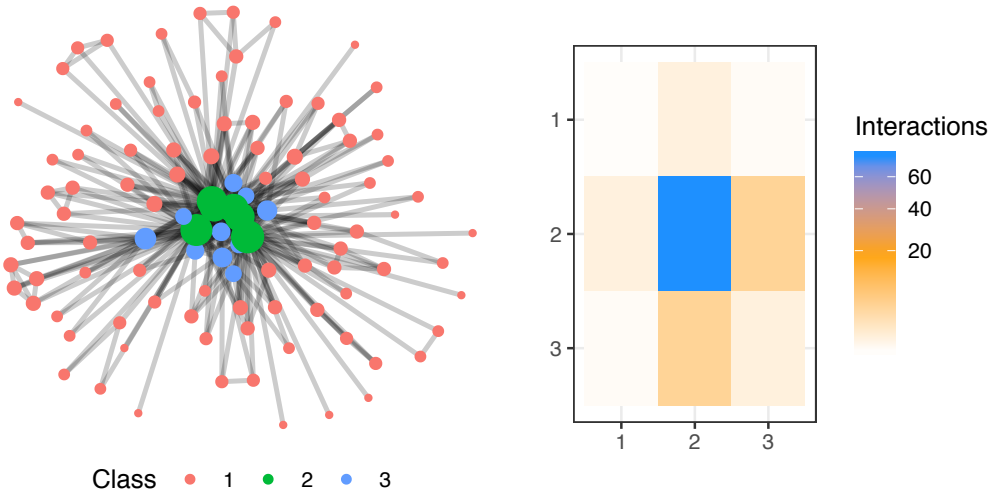



Figure A.41: Season 6 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

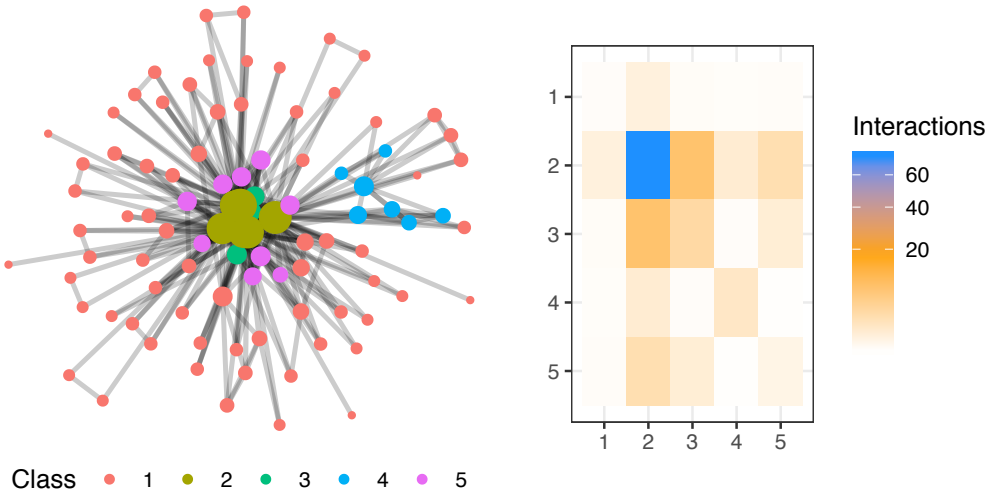



Figure A.42: Season 7 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

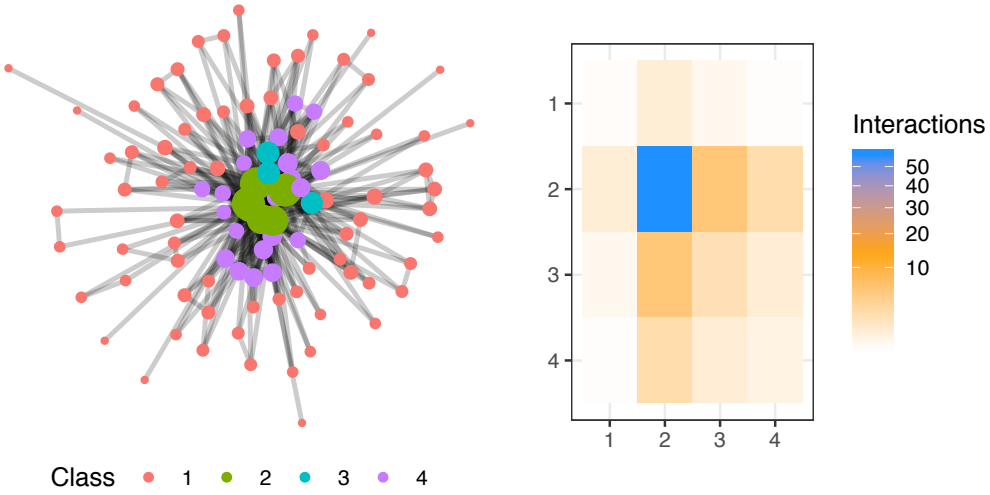



Figure A.43: Season 8 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

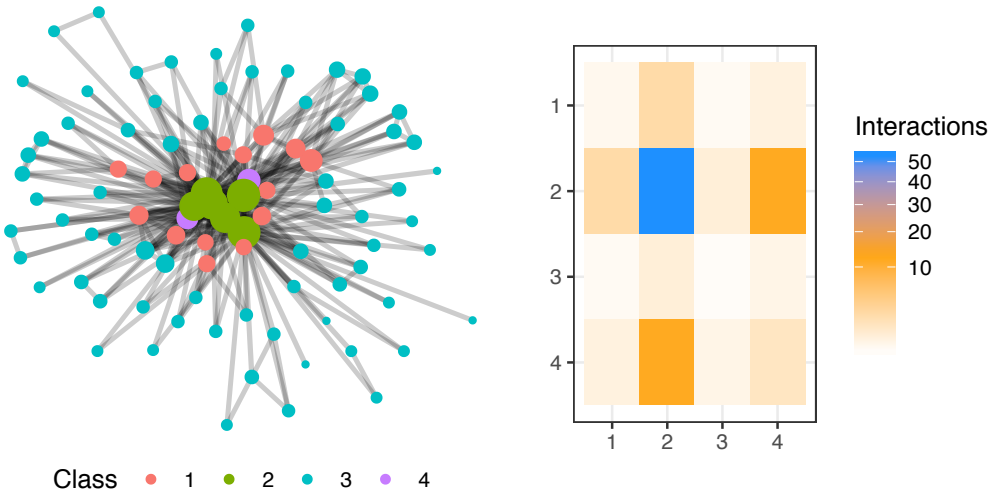



Figure A.44: Season 9 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

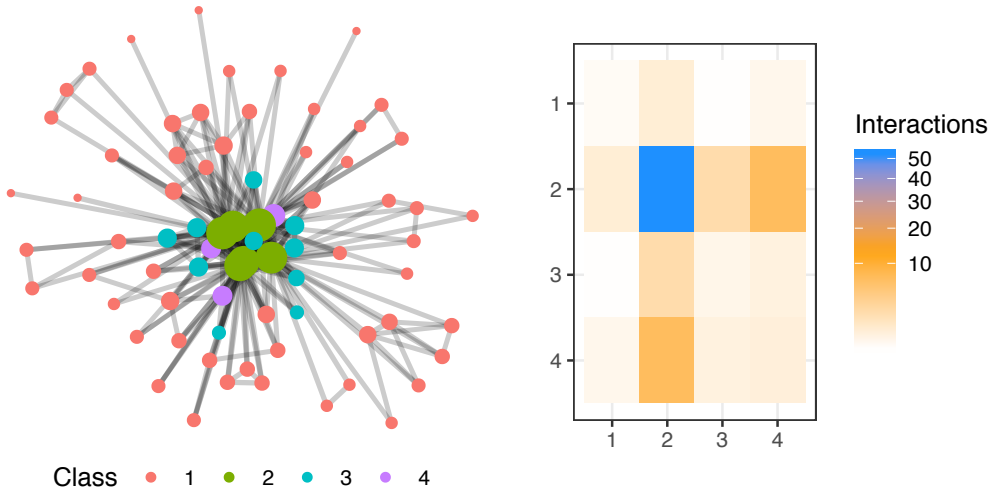



Figure A.45: Season 10 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **co-occurrence** networks.

## A.12 Stochastic block model manual season networks

Figures A.46 to A.55 show the **manual** season networks as classified into classes by the stochastic block model, along with heatmaps that represent the interaction rate within and between classes. For many of the seasons, the classes are very similar to the classes in the **co-occurrence** season networks (see Figure 6.12 and Figures A.37 to A.45).

In Season 4, the **manual** network has 5 classes compared to 4 in the **co-occurrence** network. In Season 8, the **manual** network has 3 classes compared to 4 in the **co-occurrence** network. The greatest difference, however, is in Season 7. In the **co-occurrence** network, there are 5 classes in Season 7, but there are only 3 in the **manual** network.

Despite some differences in the number of classes, in general the classes have similar characters in both datasets. In particular, the six core characters make up one class in every season of both datasets.

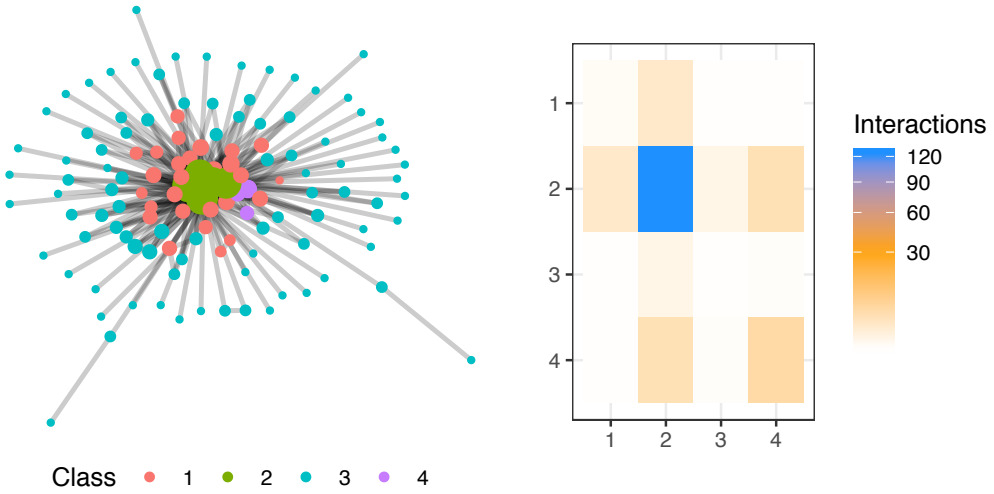



Figure A.46: Season 1 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

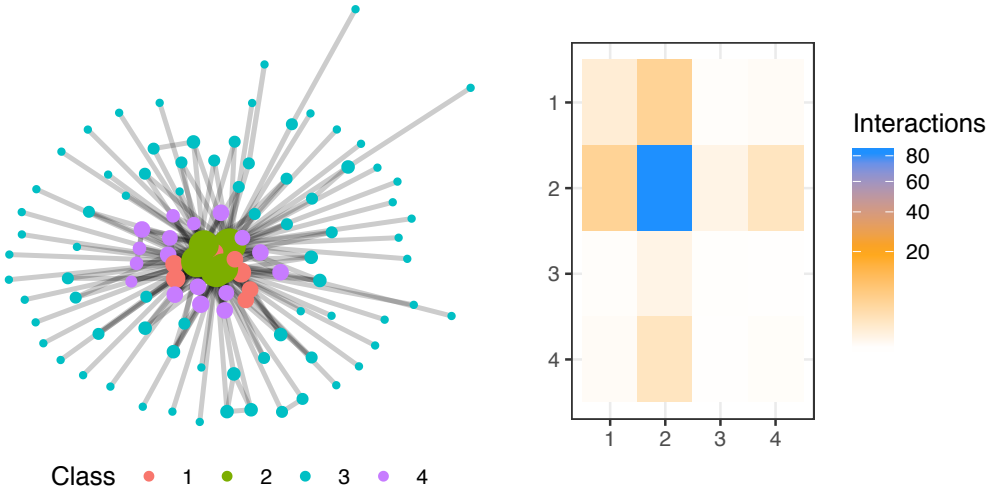



Figure A.47: Season 2 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

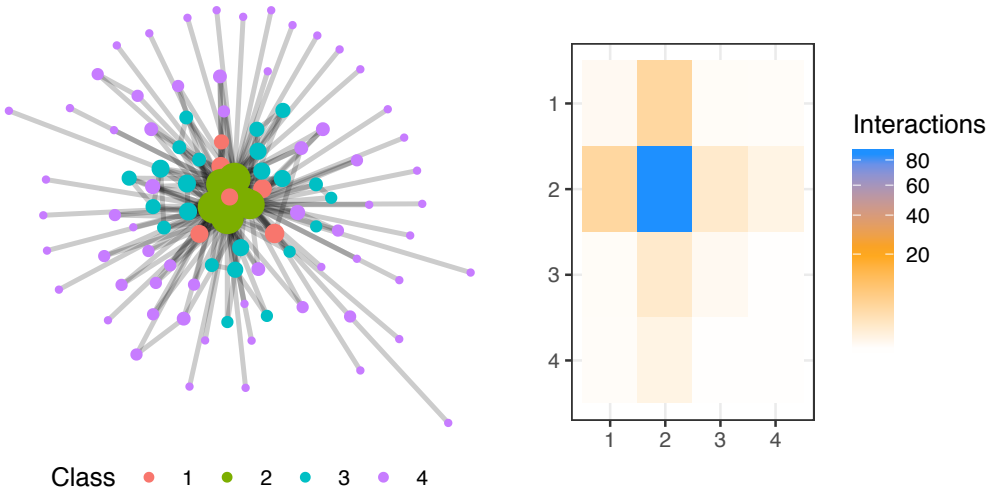



Figure A.48: Season 3 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

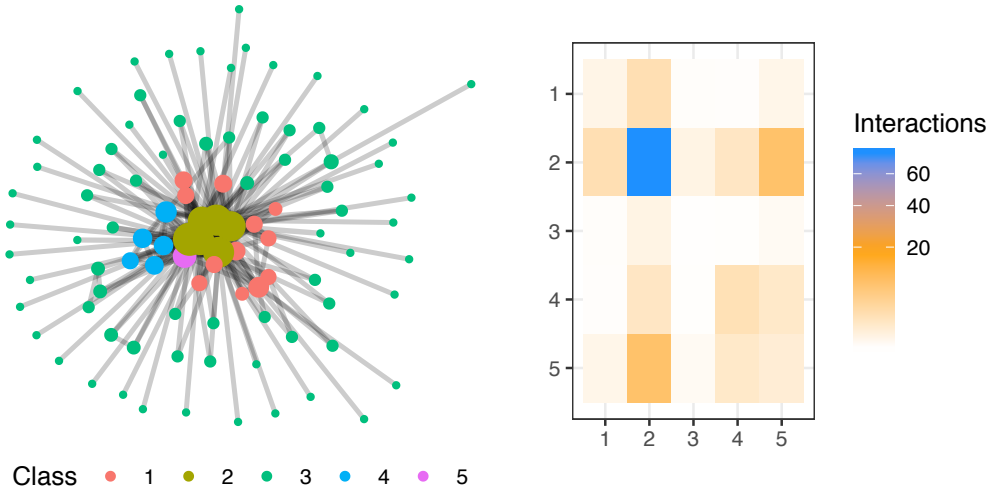



Figure A.49: Season 4 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

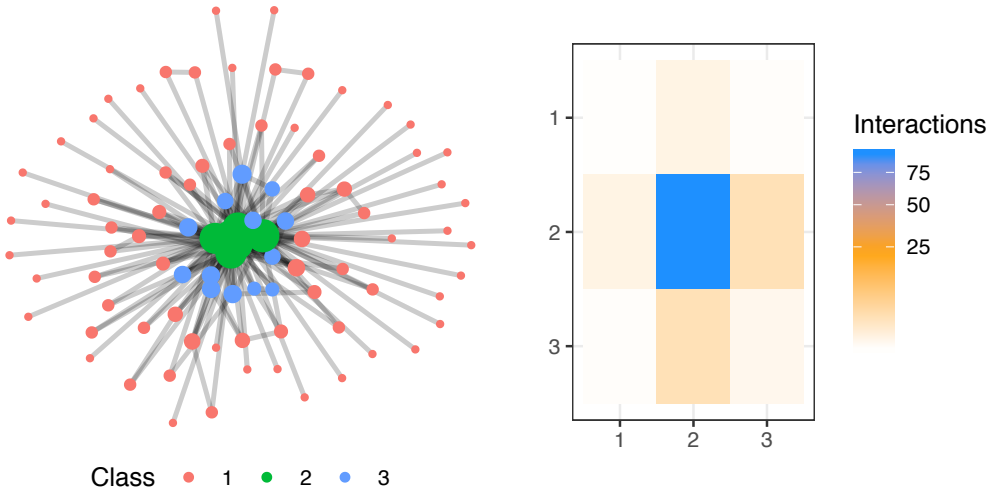



Figure A.50: Season 5 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

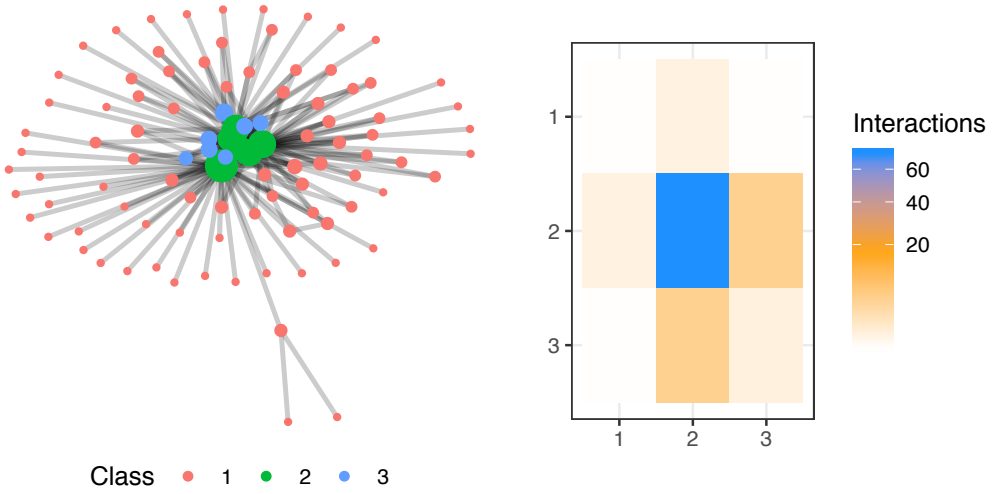



Figure A.51: Season 6 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

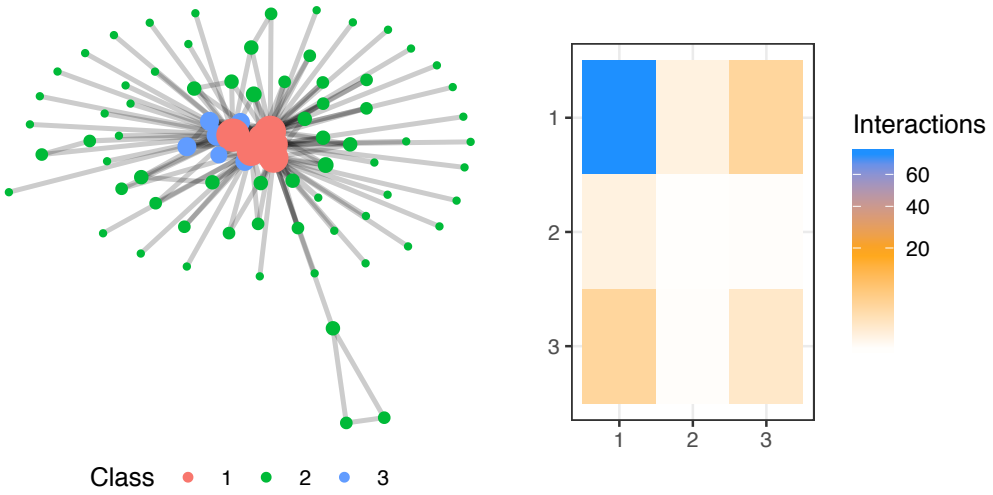



Figure A.52: Season 7 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

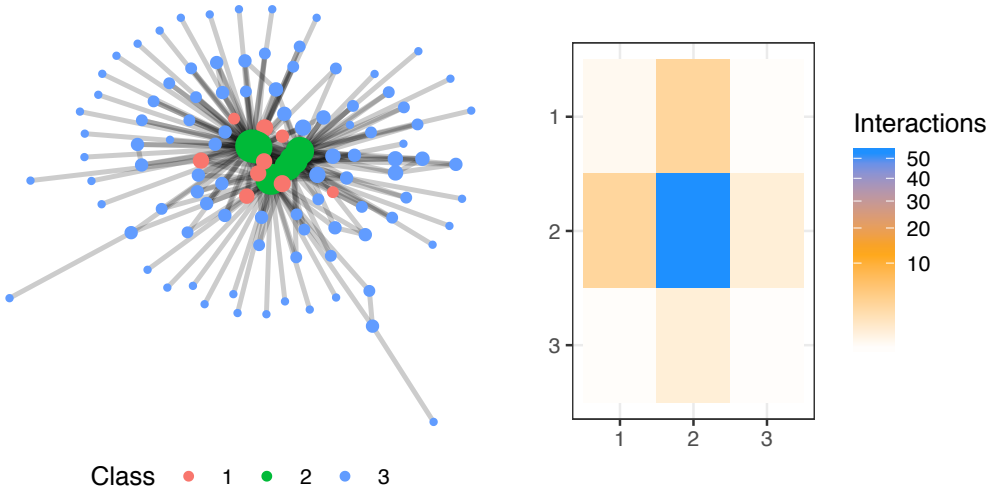



Figure A.53: Season 8 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

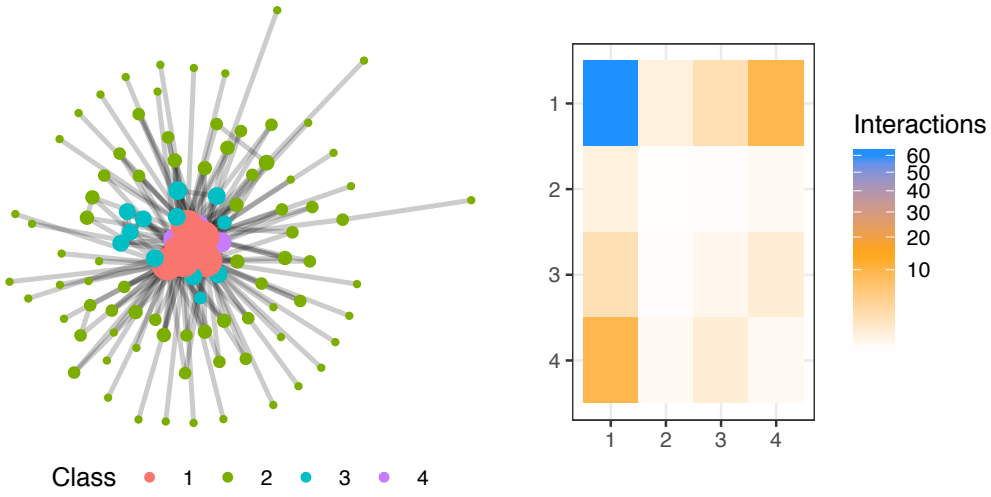



Figure A.54: Season 9 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

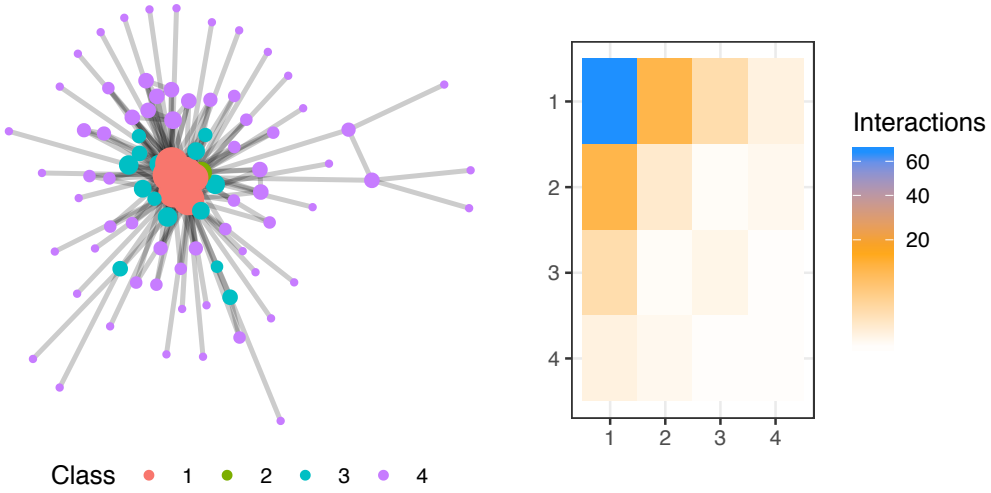



Figure A.55: Season 10 network with nodes coloured by class, and heatmap for the stochastic block model $\boldsymbol{\lambda}$ parameters for the **manual** networks.

## A.13 *Seinfeld* words per season

Figure A.56 shows the number of words spoken in each season of *Seinfeld*. We see that the number of words spoken increases over time, and we quantify this by fitting a linear model in Appendix B.3.

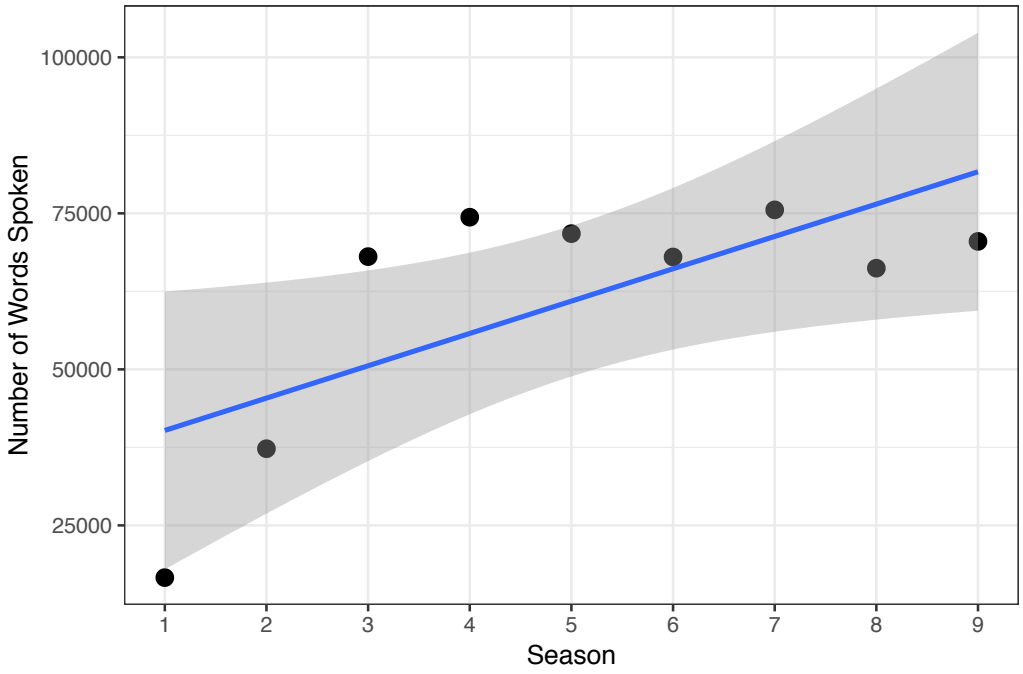



Figure A.56: Scatterplot of the number of words spoken in the 9 seasons of *Seinfeld*. The blue line shows the linear model trend line, with the confidence interval shown by the shaded region.

## A.14 *The Walking Dead* words per season

Figure A.57 shows the number of words spoken by the character Daryl Dixon in *The Walking Dead* of 8 seasons. We see that the number of words spoken decreases over time, and we quantify this by fitting a linear model in Appendix B.4.

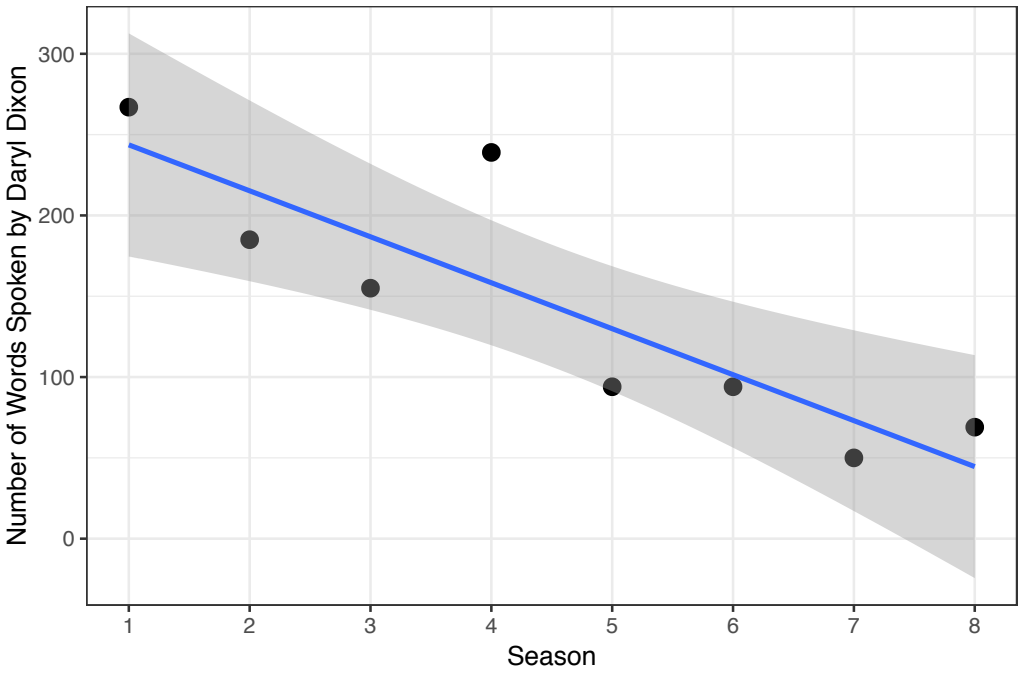



Figure A.57: Scatterplot of the number of words spoken by the character Daryl Dixon in the 8 seasons of *The Walking Dead*. The blue line shows the linear model trend line, with the confidence interval shown by the shaded region.

## A.15 Linear model variables

Figure A.58 shows histograms of the numerical variables used in the linear model in Section 6.4 in the main text. We use these histograms to look for outliers and the general shape of our data.

Figure A.59 shows scatterplots of numerical variables used in the the linear model in Section 6.4 in the main text against the episode number. Here we look for possible outliers, and non-linear trends. We also notice trends that appear linear, but we test for significant linear trends using the linear model.

Figure A.60 shows boxplots of numerical variables used in the the linear model in Section 6.4 in the main text grouped by season. We look for any outliers or non-linear trends.

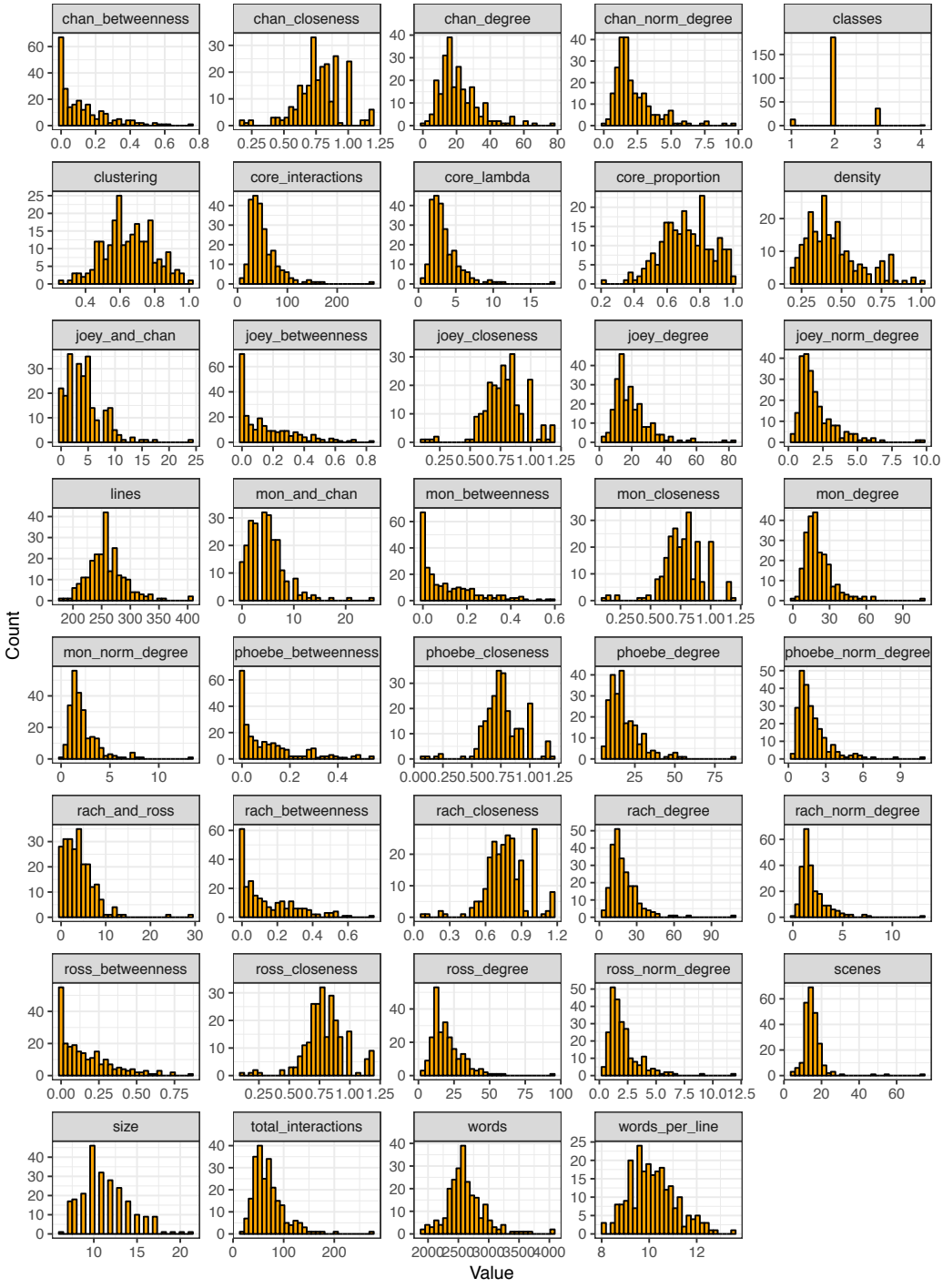



Figure A.58: Histograms of numeric variables for linear model with time using the **manual** episode networks.

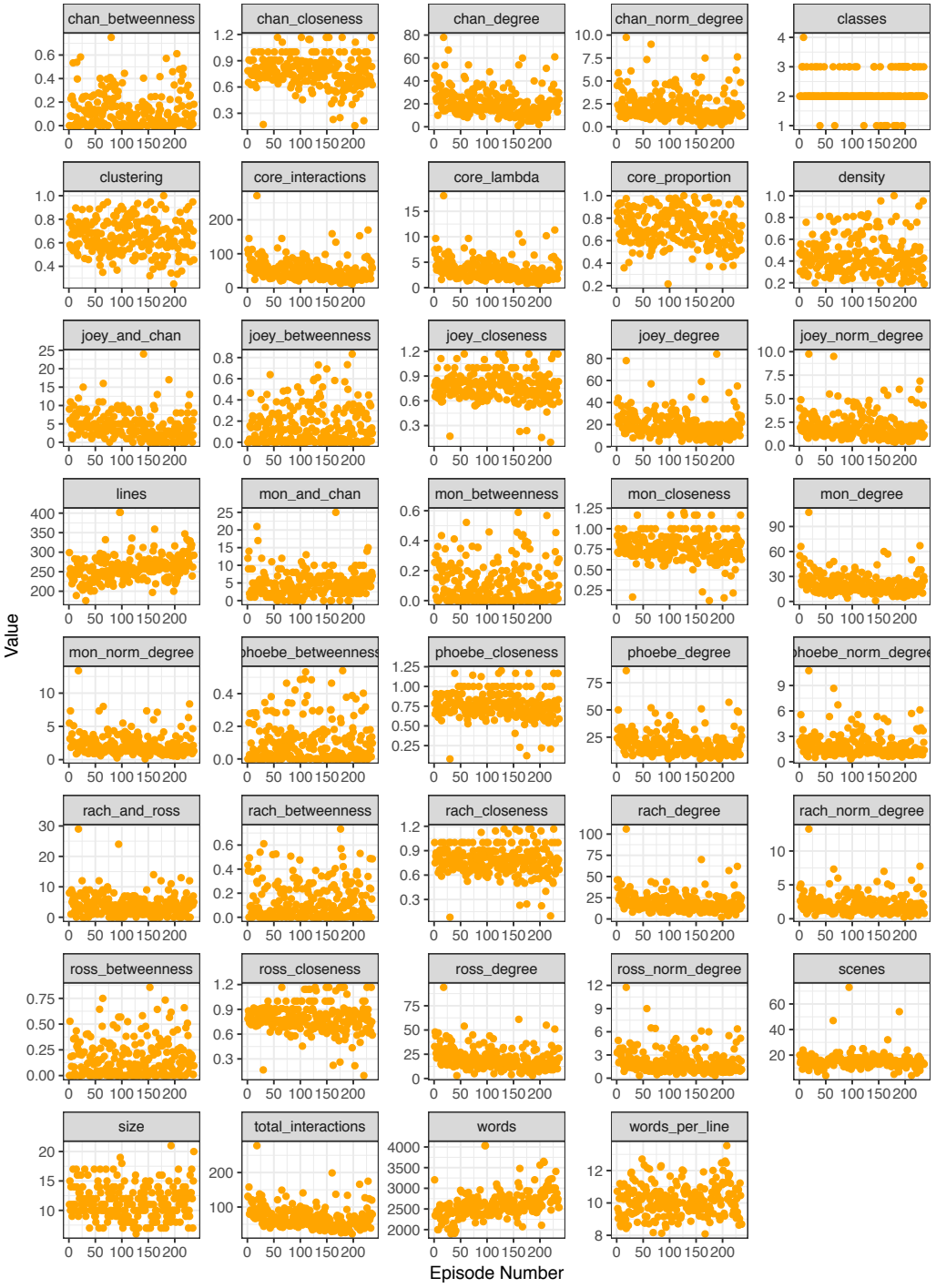



Figure A.59: Scatterplots of numeric variables against episode number for linear model using the **manual** episode networks.

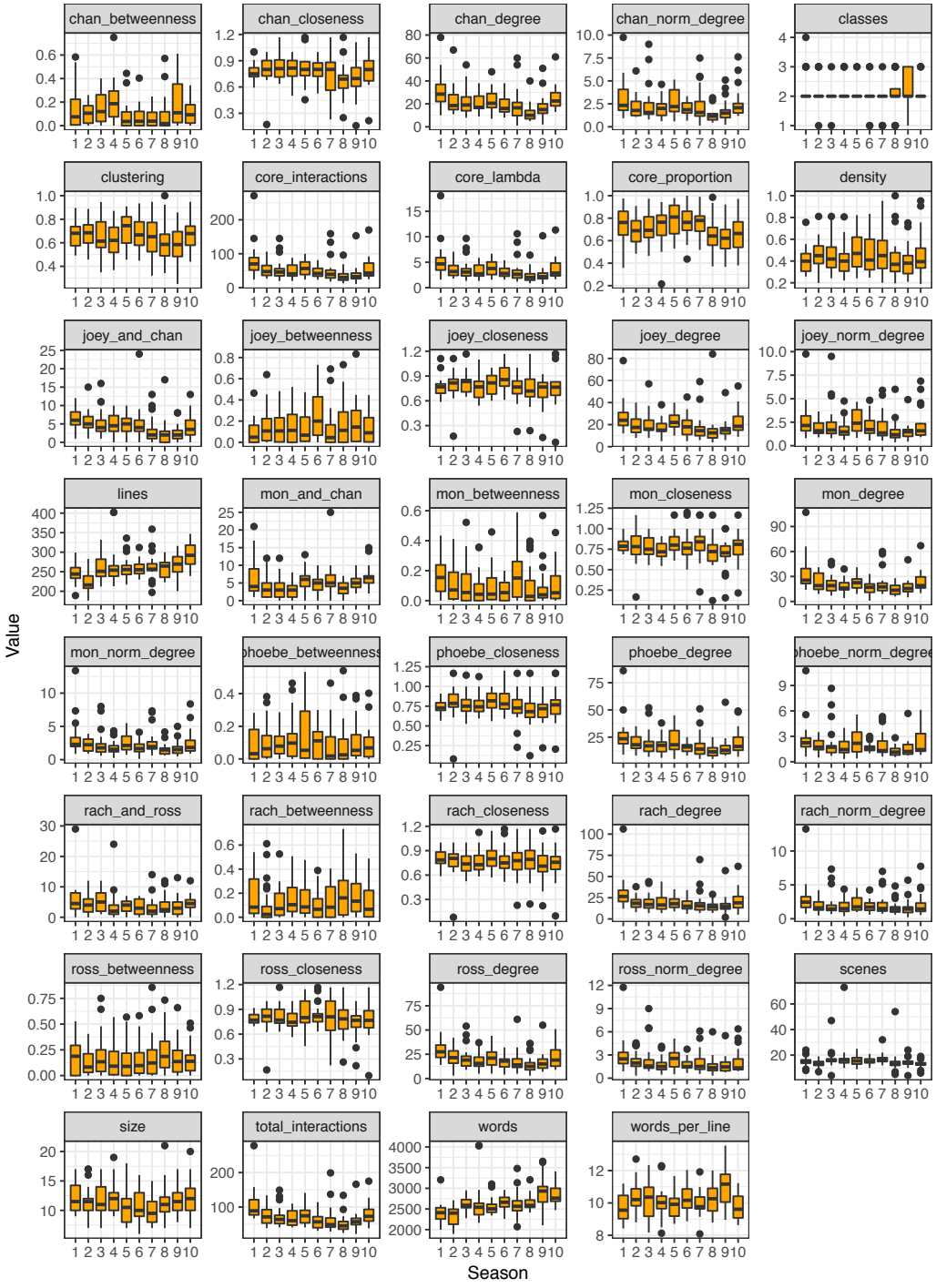



Figure A.60: Boxplots of numeric variables against season number for linear model using the **manual** episode networks.

## A.16 Bivariate model with time for co-occurrence networks

Here, we fit linear models to metrics of the **co-occurrence** networks against episode number to find trends over time. We use the same process as in Section 6.4. Figure A.61 shows histograms of the numeric variables we will fit for the **co-occurrence** linear model. Figure A.62 shows scatterplots of the numeric variables to be fit in the linear model against episode number, and Figure A.63 shows the same datapoints, but grouped into seasons via boxplots.

Figure A.64 shows the adjusted p-values of all predictors for episode number using the **co-occurrence** episode networks. The p-values are adjusted using the false discovery rate correction. In total there are 21 significant predictors at the 5% level. We plot the coefficients of the (normalised) significant predictors in Figure A.65 and find that the predictors with the greatest effect are the same as in the model using the **manual** network.

Figure A.66 shows the adjusted p-values of all predictors for season number using the **co-occurrence** episode networks. The p-values are adjusted using the false discovery rate correction. In total there are 20 significant predictors at the 5% level. We plot the coefficients of the (normalised) significant predictors in Figure A.67 and find that the predictors with the greatest effect are the same as in the model using the **manual** network.

Figure A.68 shows the adjusted p-values of all predictors for season number using the **co-occurrence** season networks. The p-values are adjusted using the False Discovery Rate correction. In total there are 7 significant predictors at the 5% level. We plot the coefficients of the (normalised) significant predictors in Figure A.69 and find that the predictors with the greatest effect are the same as in the model using the **manual** network.

One of the significant predictors for season number is `core_interactions`. Figure A.70 shows a scatterplot of the number of core interactions against season number in the **co-occurrence** network. We see the same trend as in the **manual** networks – that the *Friends* get less friendly over the series.

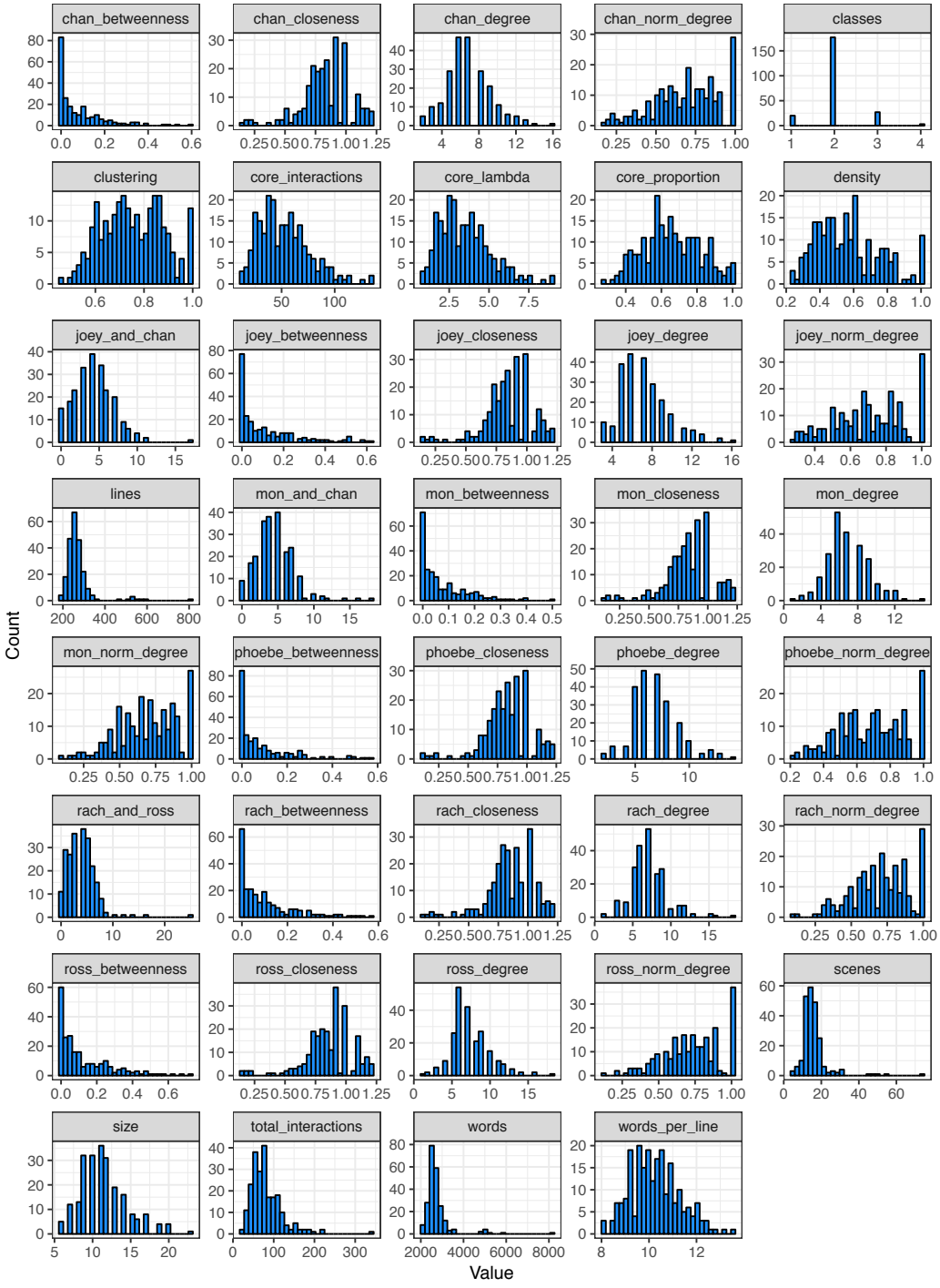



Figure A.61: Histograms of numeric variables for linear model with time using the **co-occurrence** episode networks.

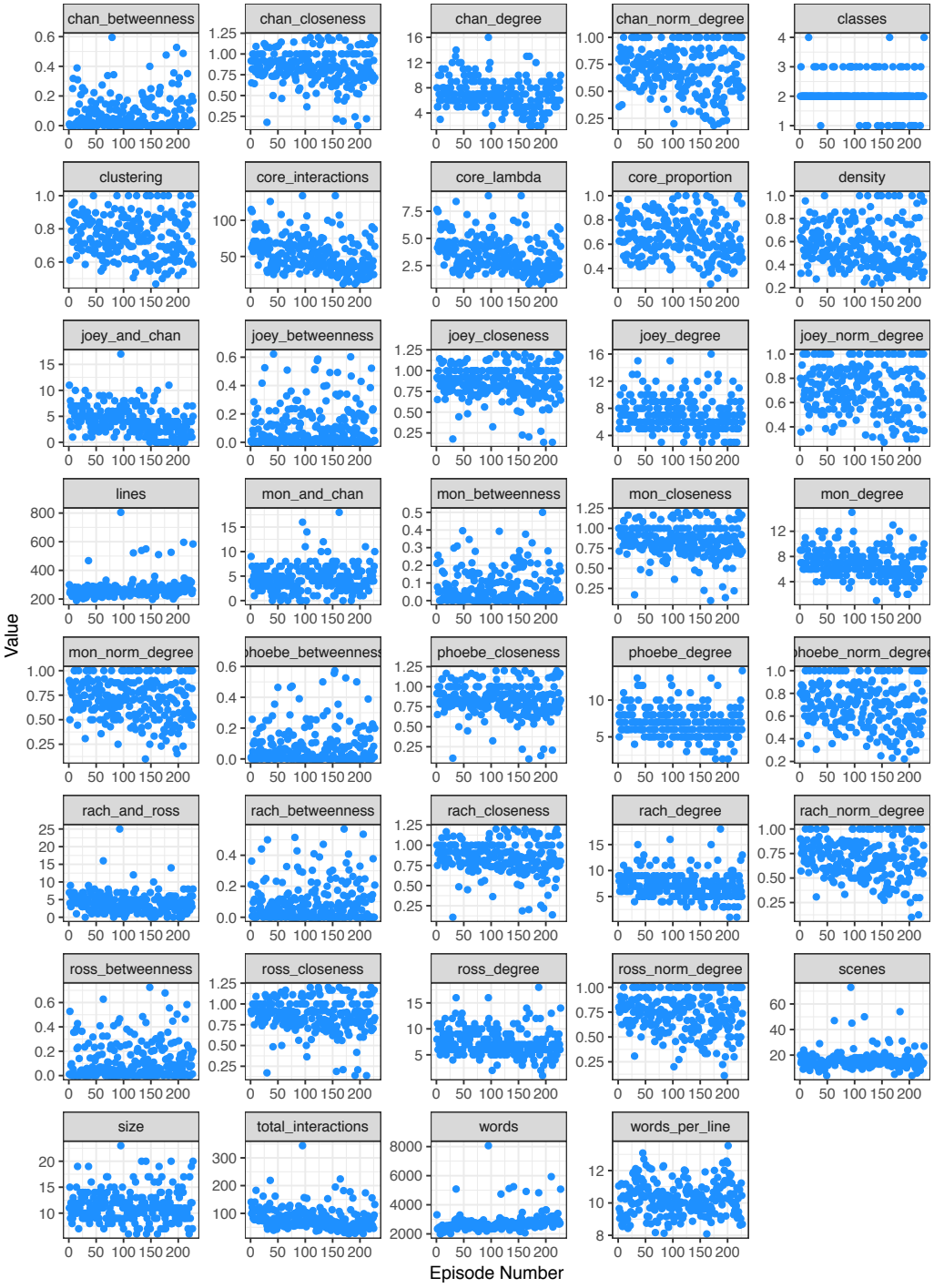



Figure A.62: Scatterplots of numeric variables against episode number for linear model using the **co-occurrence** episode networks.

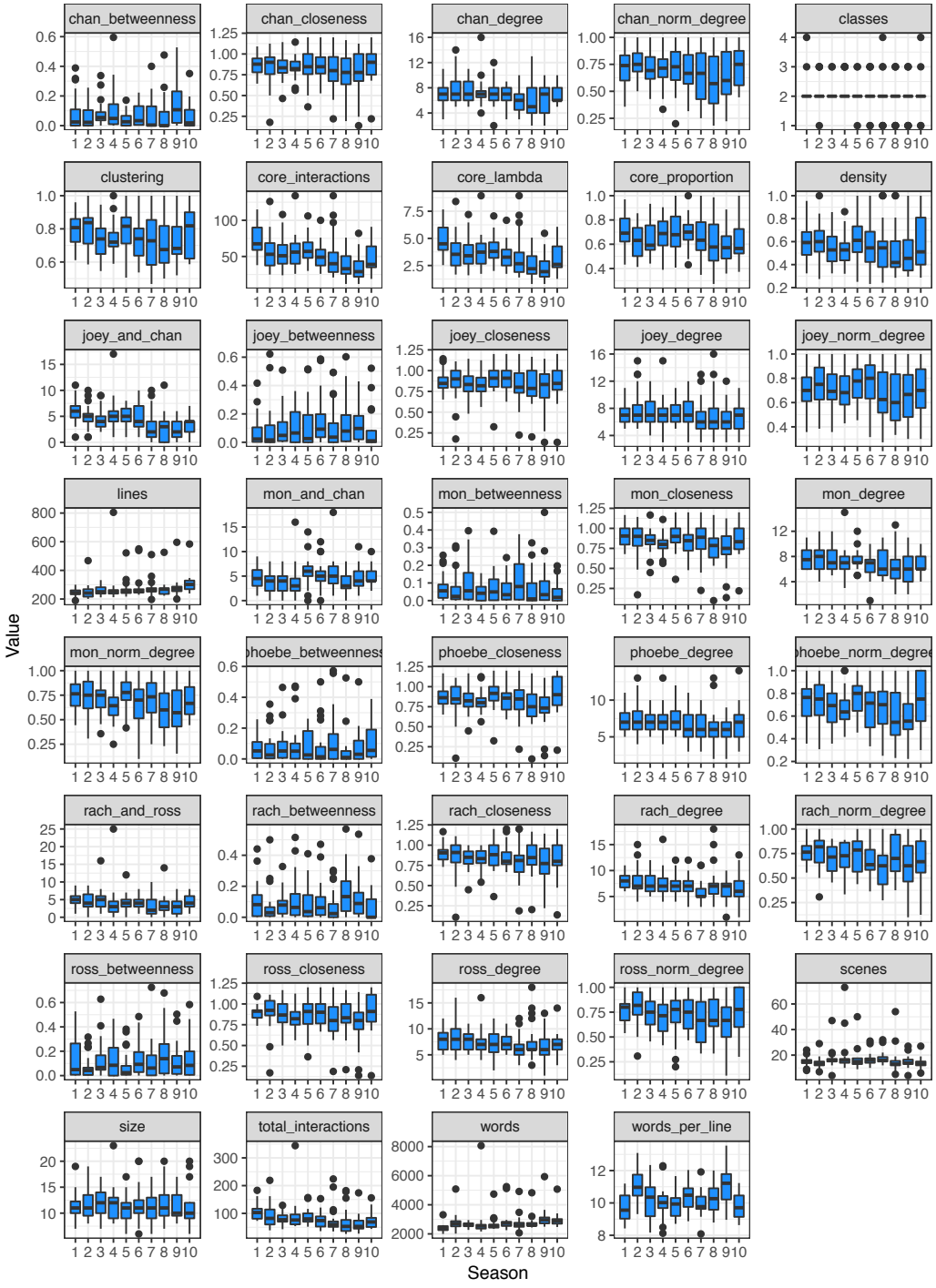



Figure A.63: Boxplots of numeric variables against season number for linear model using the **co-occurrence** episode networks.

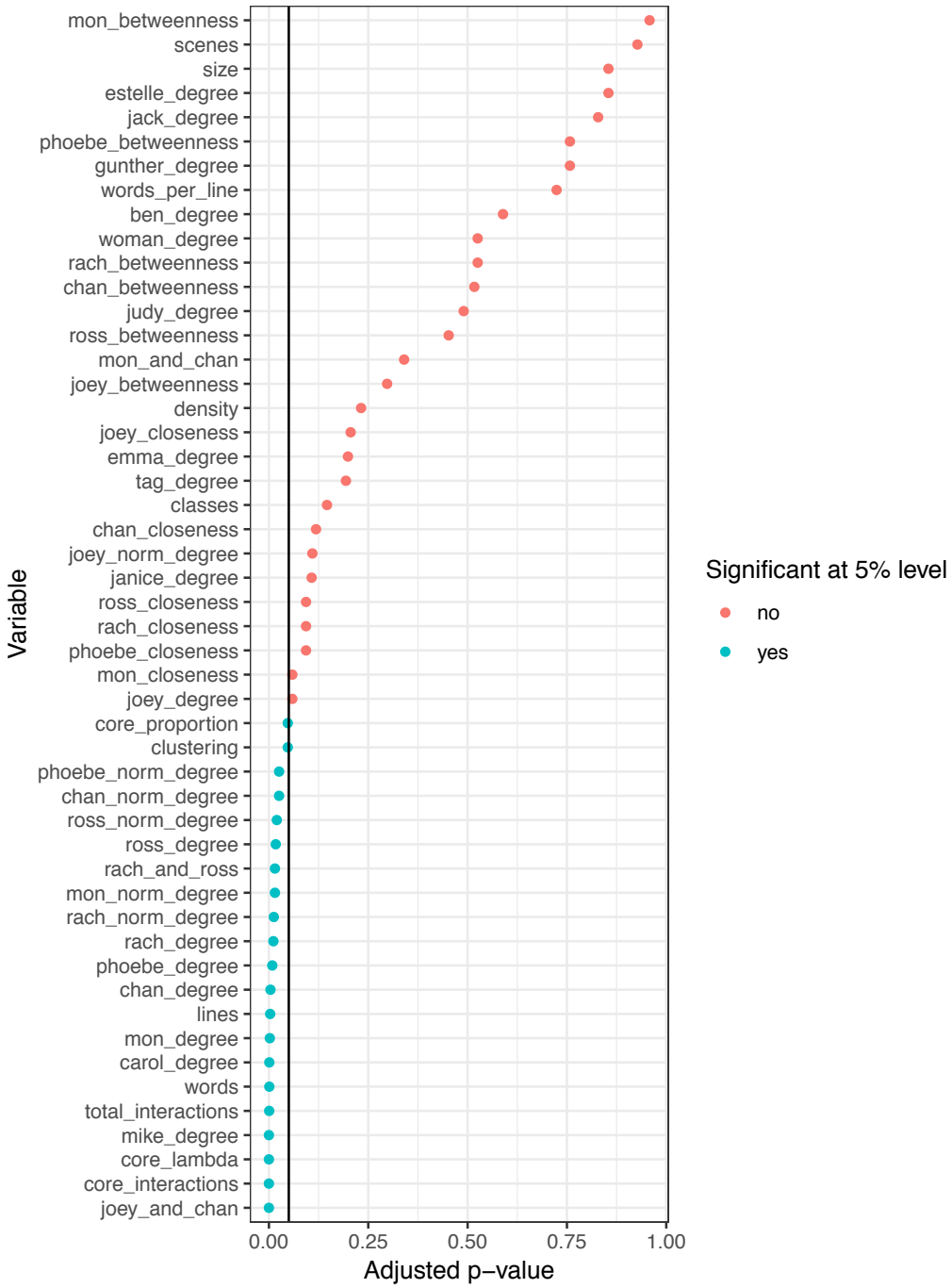



Figure A.64: Scatterplot of adjusted p-values for the linear models of each variable with episode number for the **co-occurrence** episode networks. The black line is at 0.05, which is the cut-off for p-values at the 5% significance level. Significant variables are coloured in blue, and insignificant variables are red.

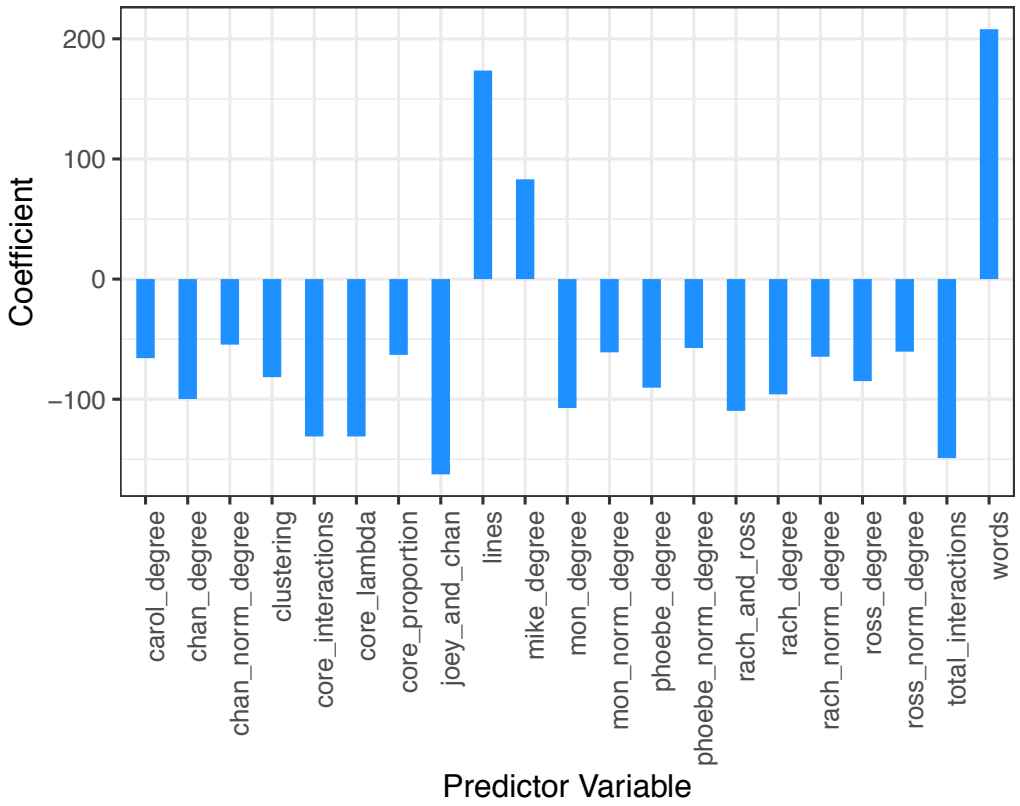



Figure A.65: Line range plot of coefficients of normalised significant predictors for episode number for the **co-occurrence** episode networks.

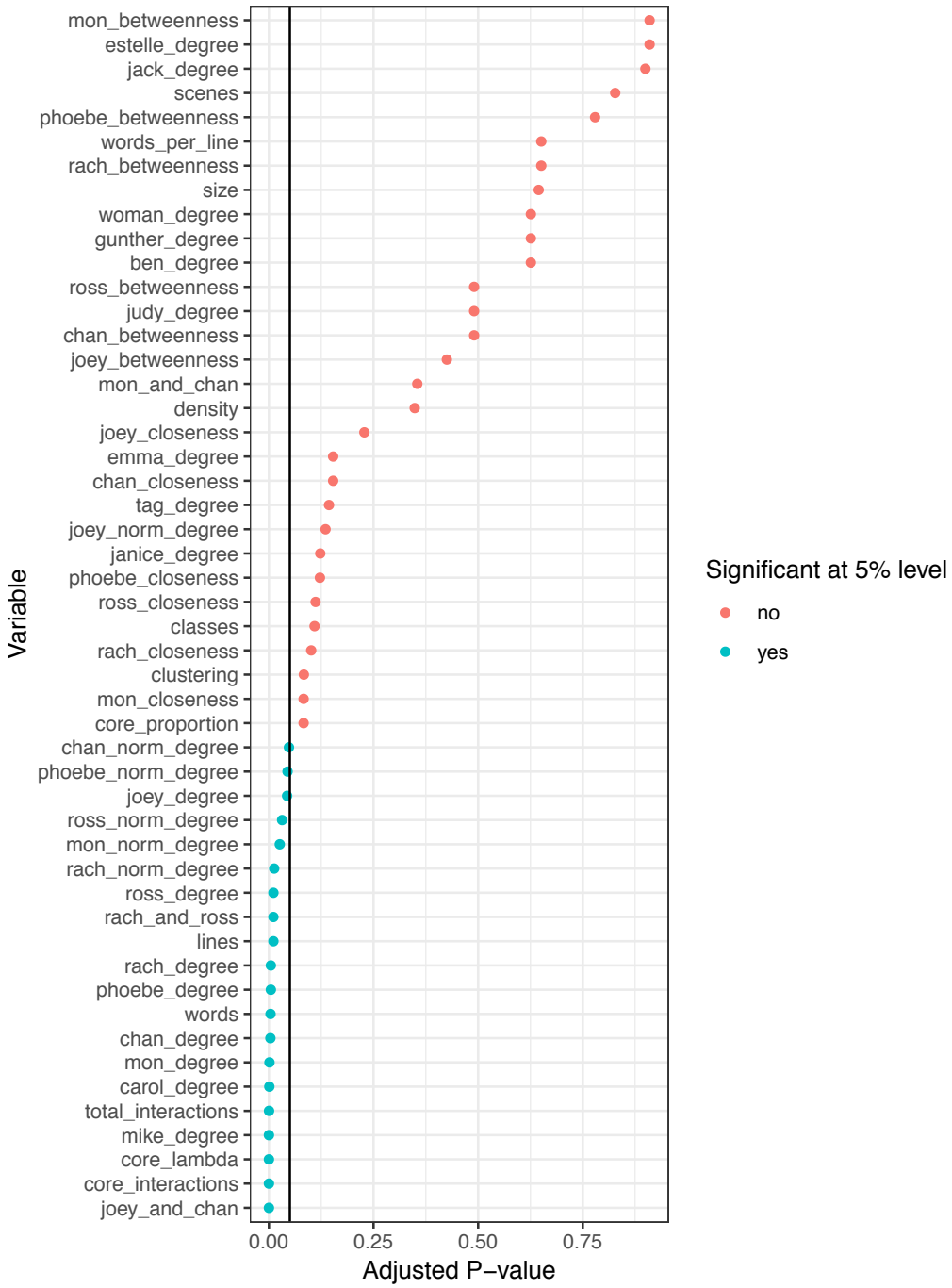



Figure A.66: Scatterplot of adjusted p-values for the linear models of each variable with season number for the **co-occurrence** episode networks. The black line is at 0.05, which is the cut-off for p-values at the 5% significance level. Significant variables are coloured in blue, and insignificant variables are red.

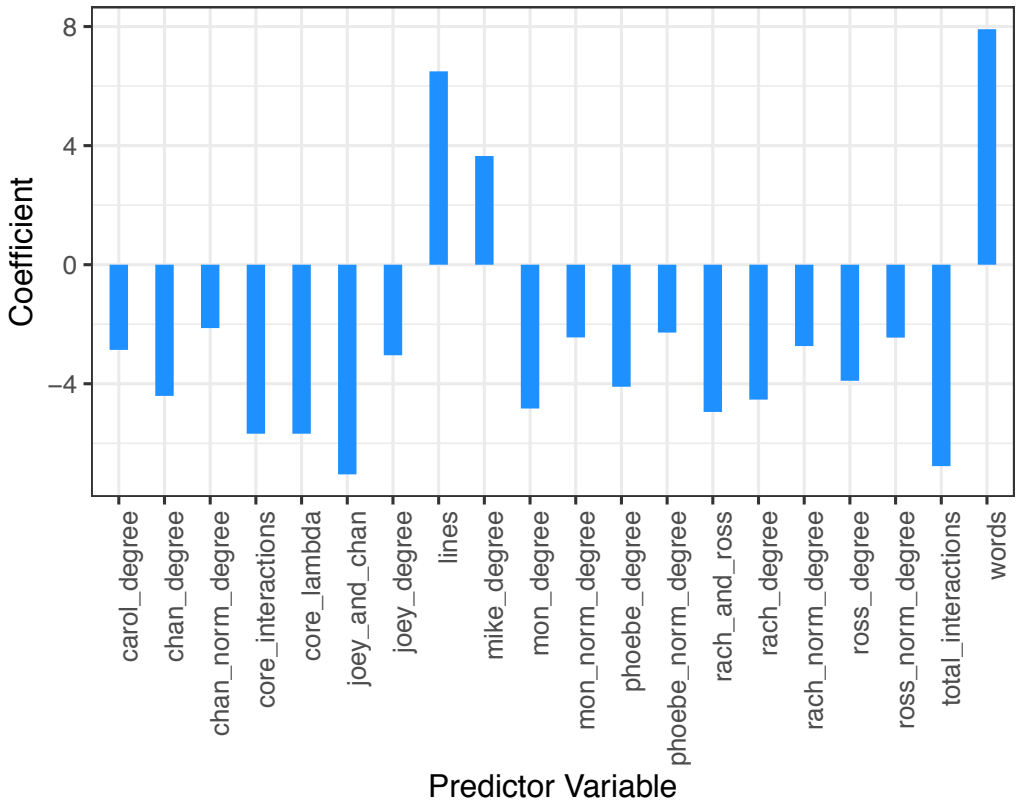



Figure A.67: Line range plot of coefficients of normalised significant predictors for season number for the **co-occurrence** episode networks.

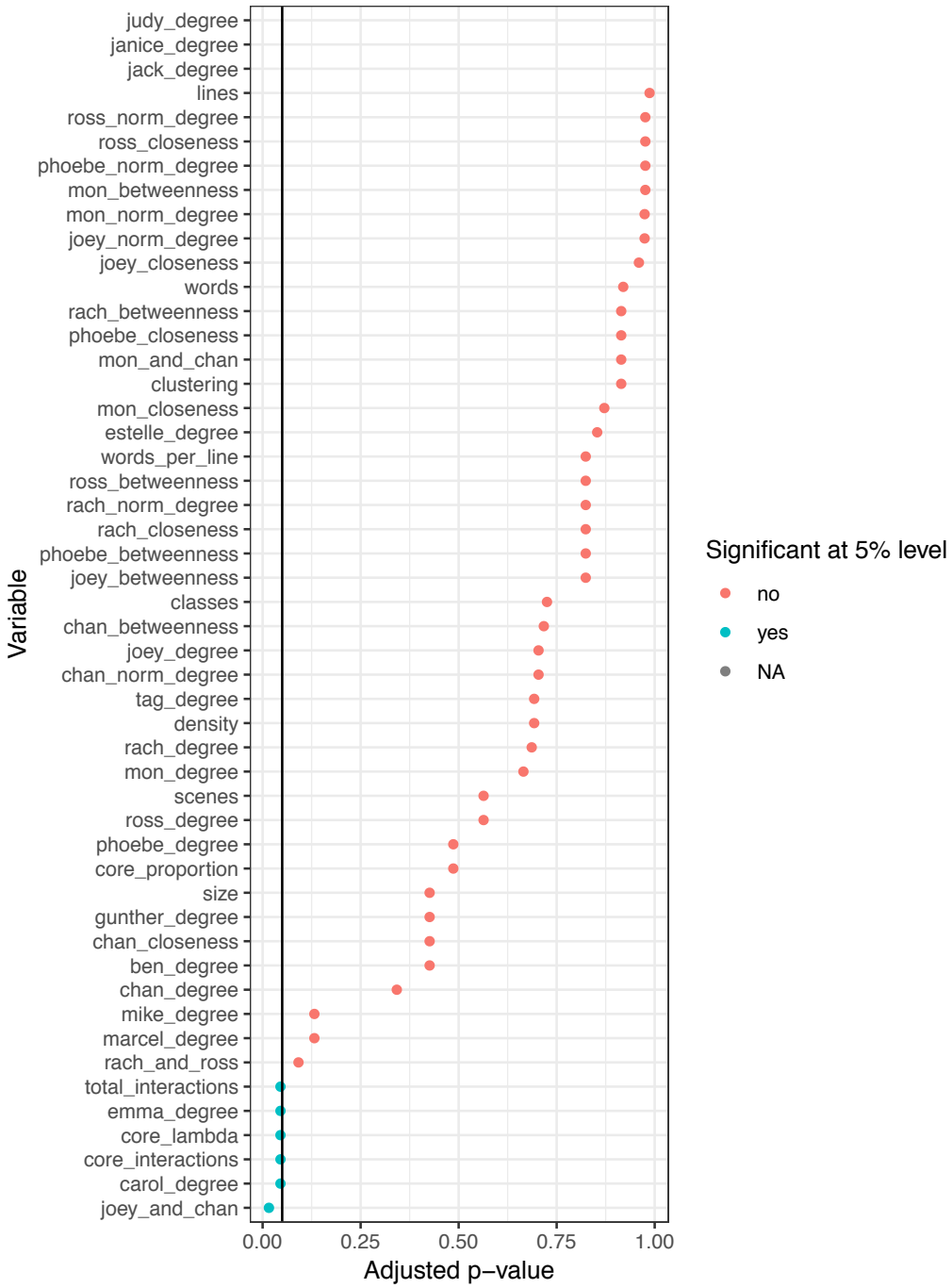



Figure A.68: Scatterplot of adjusted p-values for the linear models of each variable with season number for the **co-occurrence** season networks. The black line is at 0.05, which is the cut-off for p-values at the 5% significance level. Significant variables are coloured in blue, and insignificant variables are red.

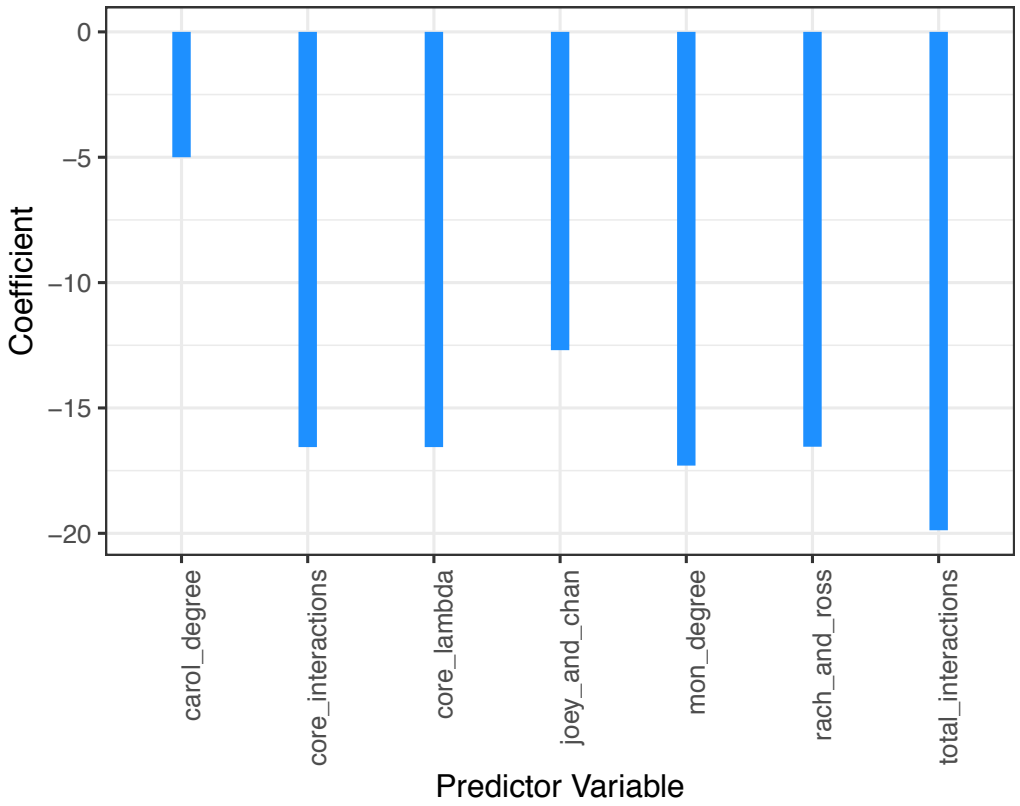



Figure A.69: Line range plot of coefficients of normalised significant predictors for season number for the **co-occurrence** season networks.

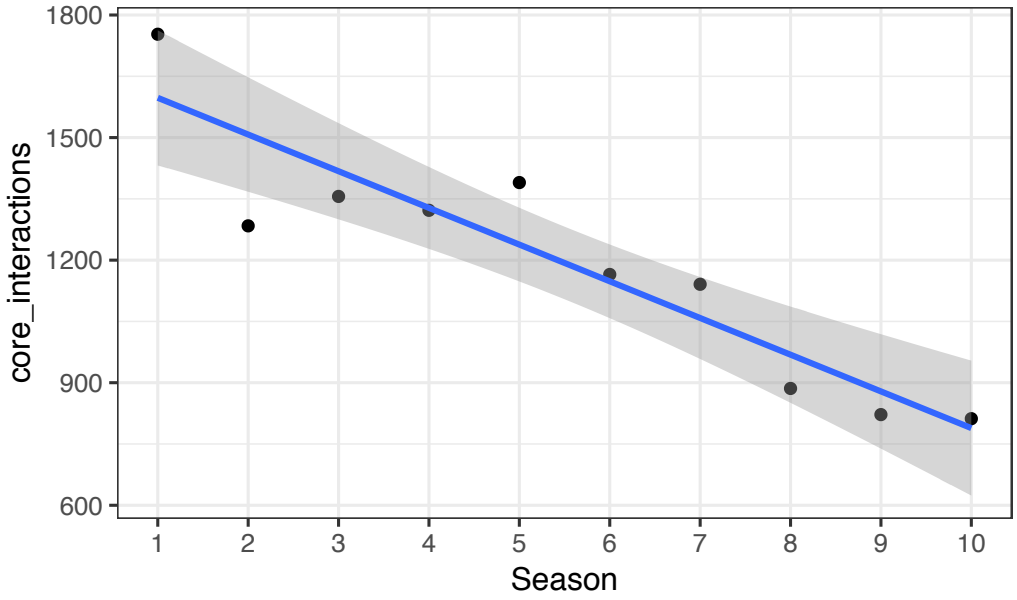



Figure A.70: Scatterplot of the number of interactions between core characters (`core_interactions`) with season number for the **co-occurrence** season networks. The line represents the predicted mean and the shaded region represents the confidence interval for the mean.

## A.17 Bivariate model with ratings

Figure A.71 shows scatterplots of the numeric variables used in the linear model to predict the rating of an episode from the **manual** network features. We look for any outliers and non-linear trends.

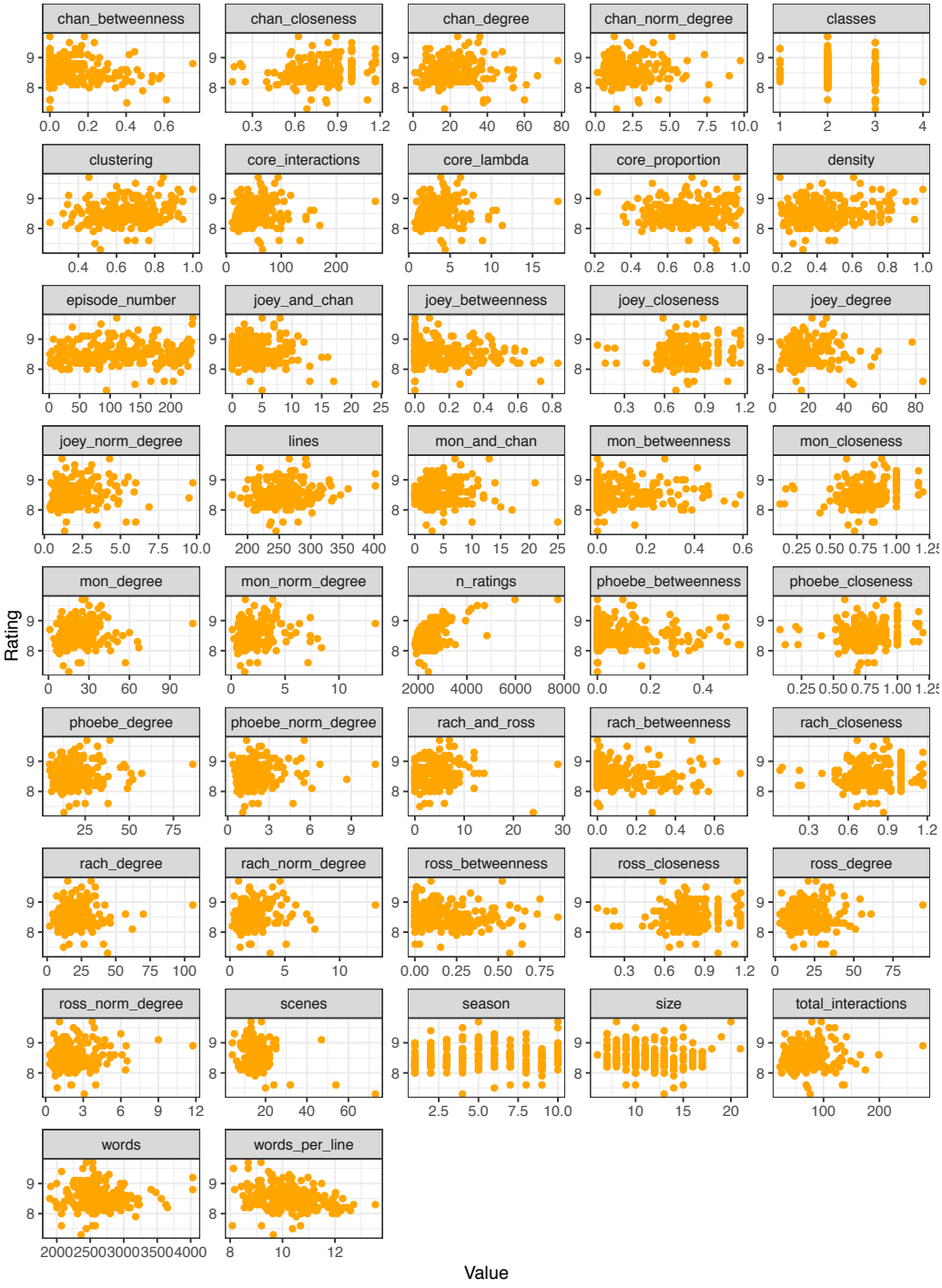



Figure A.71: Scatterplots of numeric variables against IMDb rating for multivariate linear model using the **manual** episode networks.

# Appendix B

# Code

## B.1 Name change dictionary

The following function, written for Python, was used to extract the **co-occurrence** *Friends* networks from the script. See Chapter 3 for details. The name changes were necessary to ensure characters mentioned by more than one name were recorded as single characters.

The string before the colon is the name as appears in the script, and the string after the colon is the name to be changed to. We also change the symbols "/", "&" and "-" in character names to the word "and", as they indicate more than one character speaking.

```
def get_name_changes():
    name_changes = {
        'mnca': 'monica',
        'phoe': 'phoebe',
        'chan': 'chandler',
        'rach': 'rachel',
        'rtst': 'mr. ratstatter',
        'estl': 'estelle',
        'fbob': 'fun bobby',
        'robin williams': 'robin',
        'billy crystal': 'billy',
        'phoebe sr.': 'phoebe sr',
        'dr. timothy burke': 'tim',
        'big nosed rachel': 'rachel',
        'fat monica': 'monica',
        'french phoebe': 'phoebe',
        'janice\'s voice': 'janice',
```

```
        'rache': 'rachel',
        'joey on tv': 'joey',
        'ross lays head on table)': 'ross',
        'gunter': 'gunther',
        'young ross': 'ross',
        'young monica': 'monica',
        'past life phoebe': 'phoebe',
        'dream joey': 'joey',
        'dream monica': 'monica',
        'present chandler\'s voice': 'chandler',
        'mike\'s mom': 'mike\'s mother',
        'mike\'s dad': 'mike\'s father',
        'agency guy': 'adoption agency guy',
        'a waiter': 'waiter',
        'the waiter': 'waiter',
        'a waiter in drag': 'waiter in drag',
        'prof. sherman': 'professor sherman',
        'racel': 'rachel',
        'dr horton': 'dr. horton',
        'same man\'s voice': 'man\'s voice',
        'mr.heckles': 'mr. heckles',
        'a drunken gambler': 'drunken gambler',
        'phoebe[cutting in]': 'phoebe',
        'rachel [cutting in]': 'rachel',
        'chandlers': 'chandler',
        'mr. greene': 'dr. greene',
        'jack': 'mr. geller',
        'judy': 'mrs. geller'

    }
    return name_changes


always_change = {
    '/': ' and ',
    '&': 'and',
    '-': ' and '
}
```

## B.2 Clustering coefficient t-tests

The one sample t-test for clustering of 100 simulated GER random networks with the same number of nodes and edges as the **co-occurrence** static network is below. The t-test shows that if a network with this many nodes and edges is random, the probability of observing a clustering coefficient as high as the 0.0551, or more extreme, is minute. Hence the **co-occurrence** static network has significantly more clustering than a random network.

```
One Sample t-test

data:  sapply(1:100, function(i) {
    return(transitivity(erdos.renyi.game(vcount(g), ecount(g),
         type = "gnm")))})
t = -332.34, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.05510536
95 percent confidence interval:
 0.01213884 0.01264885
sample estimates:
 mean of x
0.01239385
```

The one sample t-test for clustering of 100 simulated GER random networks with the same number of nodes and edges as the **manual** static network is below. The t-test shows that if a network with this many nodes and edges is random, the probability of observing a clustering coefficient as high as the 0.0335, or more extreme, is minute. Hence the **manual** static network has significantly more clustering than a random network.

```
One Sample t-test

data:  sapply(1:100, function(i) {
    return(transitivity(erdos.renyi.game(vcount(ga), ecount(ga),
         type = "gnm")))})
t = -165.9, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.03354714
95 percent confidence interval:
 0.005412769 0.006077803
sample estimates:
  mean of x
0.005745286
```

## B.3 *Seinfeld* words per season linear model

In *Seinfeld* (1989 – 1998), the number of words spoken by any character in each season increases significantly. The linear model results below show that the coefficient of the number of words is unlikely to be 0, given the data collected from `reddit.com` [9]. As the coefficient is positive, we conclude the number of words significantly increases. This is the same as the pattern we observe in the number of words spoken in each season of *Friends*.

```
Call:
lm(formula = seas ~ n, data = seinfeld)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6813 -1.0333  0.2325  0.6027  3.0858

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.220e-01  2.328e+00  -0.353   0.7344
n            9.554e-05  3.647e-05   2.620   0.0344 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.08 on 7 degrees of freedom
Multiple R-squared:  0.495,Adjusted R-squared:  0.4229
F-statistic: 6.862 on 1 and 7 DF,  p-value: 0.03443
```

## B.4 *The Walking Dead* words per season linear model

In *The Walking Dead* (2010 – present), the average number of words spoken by Daryl Dixon per episode in each season decreases significantly. The linear model results below show that the coefficient of the number of words is unlikely to be 0, given the data collected from `reddit.com` [10]. As the coefficient is negative, we conclude the number of words significantly decreases. This is the opposite of the pattern we observe in the number of words spoken in each season of *Friends*.

```
Call:
lm(formula = seas ~ n, data = daryldixon)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4249 -0.9172 -0.1219  0.5173  1.9953

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.290602   1.013538    8.18  0.00018 ***
n           -0.026301   0.006232   -4.22  0.00556 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.328 on 6 degrees of freedom
Multiple R-squared:  0.748,Adjusted R-squared:  0.706
F-statistic: 17.81 on 1 and 6 DF,  p-value: 0.00556
```

# Bibliography


[1] Internet movie database. URL www.imdb.com. Accessed: 13/11/2018.

[2] Neilsen Rating. URL https://www.nielsen.com/ssa/en/solutions/measurement/television.html. Accessed: 20/12/2018.

[3] Project Gutenberg. URL www.gutenberg.org. Accessed: 7/1/2019.

[4] Friends. URL https://www.stan.com.au/watch/friends. Accessed: 20/12/2018.

[5] Friends Soundtrack, 2004. URL https://www.what-song.com/Tvshow/100288/Friends/s/100954. Accessed: 20/12/2018.

[6] Crazy For Friends, 2004. URL http://www.livesinabox.com/friends/scripts.shtml. Accessed: 27/7/2017.

[7] The women missing from the silver screen and the technology used to find them, March 2017. URL https://www.google.com/about/main/gender-equality-films. Accessed: 1/6/2017.

[8] Lines Spoken Per Episode By the Four Most Popular Characters From the Office [OC], 2017. URL https://www.reddit.com/r/dataisbeautiful/comments/6qz4b9/lines_spoken_per_episode_by_the_four_most_popular. Accessed: 11/12/2018.

[9] Seinfeld: Percent of words per character per season [OC], 2018. URL https://www.reddit.com/r/dataisbeautiful/comments/8yundp/seinfeld_percent_of_words_per_character_per. Accessed: 11/12/2018.

[10] Number of Words Spoken by Daryl Dixon per Season, November 2018. URL https://www.reddit.com/r/thewalkingdead/comments/

9pw9kc/number_of_words_spoken_by_daryl_dixon_per_season. Accessed: 11/12/2018.

[11] Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. Social Network Analysis of Alice in Wonderland. In *Workshop on Computational Linguistics for Literature*, pages 88–96, 2012.

[12] Apoorv Agarwal, Anup Kotalwar, and Owen Rambow. Automatic Extraction of Social Networks from Literary Text: A Case Study on Alice in Wonderland. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1202–1208, 2013.

[13] Jae-wook Ahn, Meirav Taieb-Maimon, Awalin Sopan, Catherine Plaisant, and Ben Shneiderman. Temporal Visualization of Social Network Dynamics: Prototypes for Nation of Neighbors. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 309–316. Springer, 2011.

[14] Christopher Aicher, Abigail Z. Jacobs, and Aaron Clauset. Adapting the Stochastic Block Model to Edge-Weighted Networks. arXiv:1305.5782v1 [stat.ML], May 2013.

[15] Hirotugu Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. In B. N. Petrov and F. Csaki, editors, *Proceedings of the 2nd International Symposium on Information Theory*, pages 267–281, 1973.

[16] Hirotugu Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

[17] Ricardo Alberich, José Miró-Julià, and Francesc Rossello. Marvel Universe looks almost like a real social network. arXiv:cond-mat/0202174v1 [cond-mat.dis-nn], February 2002.

[18] Alexandra Albright. The One With All The Quantifiable Friendships, January 2015. URL https://thelittledataset.com/2015/01/20/the-one-with-all-the-quantifiable-friendships. Accessed: 7/7/2017.

[19] Hannah Anderson and Matt Daniels. Film Dialogue from 2,000 screenplays, Broken Down by Gender and Age, April 2016. URL https://pudding.cool/2017/03/film-dialogue/index.html. Accessed: 1/6/2017.

[20] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, October 1999.

[21] Ana L. C. Bazzan. I will be there for you: six friends in a clique. arXiv:1804.04408v1 [cs.SI], April 2018.

[22] Allison Bechdel. *Dykes to Watch Out For*. Firebrand Books, October 1986.

[23] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[24] Andrew Beveridge and Jie Shan. Network of Thrones. *Math Horizons*, 23(4):18–22, 2016.

[25] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a Mixture Model for Clustering with the Integrated Classification Likelihood. Technical Report RR-3521, INRIA, October 1998.

[26] Lawrence A. Birnbaum, Kristian J. Hammond, Nicholas D. Allen, and John R. Templon. System and method for using data and angles to automatically generate a narrative story. Patent, October 2018.

[27] Ben Blatt. Which Friends on Friends Were the Closest Friends?, May 2014. URL `http://www.slate.com/articles/arts/culturebox/2014/05.html`. Accessed: 7/6/2017.

[28] Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics Theory and Experiment*, 2008.

[29] Tjeerd W. Boonstra, Mark E. Larsen, Samuel Townsend, and Helen Christensen. Validation of a smartphone app to map social networks of proximity. *PLoS ONE*, 12(12):1–13, December 2017. doi: 10.1371/journal.pone.0189877. URL `https://doi.org/10.1371/journal.pone.0189877`.

[30] Xavier Bost, Vincent Labatut, Serigne Gueye, and Georges Linarès. Narrative Smoothing: Dynamic Conversational Network for the Analysis of TV Series Plots. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1111–1118. IEEE, 2016.

[31] Moses Boudourides and Sergios Lenis. Network Analysis of Shakespeare's Macbeth, October 2015. URL https://mboudour.github.io/2015/10/28/Shakespeare's-Macbeth-Network.html. Accessed: 9/8/2017.

[32] Larry Brody. *Television writing from the inside out: Your channel to success*. New York : Applause Theatre and Cinema Books, 1st edition, 2003.

[33] Joseph E. Cavanaugh. Unifying the derivations of the Akaike and corrected Akaike information criteria. *Statistics and Probability Letters*, 33(2):201–208, April 1997.

[34] Asli Celikyilmaz, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. The Actor-Topic Model for Extracting Social Networks in Literary Narrative. In *NIPS Workshop: Machine Learning for Social Computing*, 2010.

[35] Snigdha Chaturvedi, Shashank Srivastava, and Dan Roth. Where Have I Heard This Story Before? Identifying Narrative Similarity in Movie Remakes. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2 of *Short Papers*, pages 673–678, 2018.

[36] Rex H.G. Chen, C.-C. Chen, and Chi-Ming J. Chen. Unsupervised cluster analyses of character networks in fiction: Community structure and centrality. *Knowledge-Based Systems*, 2018. doi: 10.1016/j.knosys.2018.10.005.

[37] Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Ting-Hao Huang, Huang, and Lun-Wei Ku. EmotionLines: An Emotion Corpus of Multi-Party Conversations. arXiv:1802.08379v2 [cs.CL], February 2018.

[38] Yu-Hsin Chen and Jinho D. Choi. Character Identification on Multiparty Conversation: Identifying Mentions of Characters in TV Shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles, 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-3612.

[39] Gerda Claeskens and Nils L. Hjort. *Model Selection and Model Averaging*. Cambridge University Press, 2012.

[40] James J. Collins and Carson C. Chow. It's a small world. *Nature*, 393: 409–410, June 1998.

[41] David Crane, Kevin S. Bright, and Marta Kauffman. Friends: Final Thoughts. (DVD), 2004.

[42] Daniel D'Addario. Friends Is Headed to Netflix. *Time*, October 2014.

[43] Henrique Ferraz de Arruda, Filipi Nascimento Silva, and Vanessa Queiroz Marinho. Representation of texts as complex networks: a mesoscopic approach. Technical report, Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil. São Carlos Institute of Physics, University of São Paulo, São Carlos, SP, Brazil, 2017.

[44] Léa A. Deleris, Francesca Bonin, Elizabeth Daly, Stéphane Deparis, Yufang Hou, Charles Jochim, Yassine Lassoued, and Killian Levacher. Know Who Your Friends Are: Understanding Social Connections from Unstructured Text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HTL 2018, New Orleans, Louisiana, USA, June 2-4, 2018, Demonstrations*, pages 76–80, 2018. URL `https://aclanthology.info/papers/N18-5016/n18-5016`.

[45] Arthur P. Dempster, Natalie M. Laird, and Donald B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[46] David K. Elson, Nicholas Dames, and Kathleen R. McKeown. Extracting Social Networks from Literary Fiction. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 138–147. Association for Computational Linguistics, 2010.

[47] Philippe Ercolessi, Christine Sénac, and Hervé Bredin. Toward plot de-interlacing in TV series using scenes clustering. In *2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, June 2012.

[48] P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[49] Vincent Fortuin, Romann M. Weber, Sasha Schriber, Diana Wotruba, and Markus Gross. InspireMe: Learning Sequence Models for Stories.

In *AAAI Conference on Artificial Intelligence*, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16100.

[50] Katie Franklin, Simon Elvery, and Ben Spraggon. Star Wars: every scene from I-VI charted, October 2016. URL http://www.abc.net.au/news/2015-12-16/star-wars-every-scene/7013826. Accessed: 9/8/2017.

[51] Lyle Friedman, Matt Daniels, and Ilia Blinderman. Hollywood's Gender Divide and its Effect on Films, March 2017. URL https://pudding.cool/2017/03/bechdel. Accessed: 1/6/2017.

[52] Thomas M.J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11): 1129–1164, 1991.

[53] Evelina Gabasova. The Star Wars social network, December 2015. URL http://evelinag.com/blog/2015/12-15-star-wars-social-network. Accessed: 17/5/2017.

[54] Edgar N. Gilbert. Random Graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, December 1959.

[55] Shirin Glander. Network analysis of Game of Thrones family ties, May 2017. URL https://shiring.github.io/networks/2017/05/15/got_final. Accessed: 1/3/2018.

[56] Pablo M. Gleiser. How to become a superhero. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(7), September 2007.

[57] Jakub Glinka. Sentiment Analysis of The Lord Of The Rings with tidytext, March 2017. URL http://www.jakubglinka.com/2017-03-01-text_mining_part1. Accessed: 1/6/2017.

[58] Philip J. Gorinski and Mirella Lapata. Movie Script Summarization as Graph-based Scene Extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, 2015.

[59] Martin Grandjean. Network visualization: mapping Shakespeare's tragedies, December 2015. URL http://www.martingrandjean.ch/network-visualization-shakespeare. Accessed: 1/6/2017.

[60] Mark Granroth-Wilding and Stephen Clark. What Happens Next? Event Prediction Using a Compositional Neural Network Model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2727–2733, Phoenix, Arizona, 2016. AAAI Press.

[61] Yuval Noah Harari. *Sapiens: A Brief History of Humankind*. Harper, 2011.

[62] Hua He, Denilson Barbosa, and Grzegorz Kondrak. Identification of Speakers in Novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1312–1320, 2013.

[63] Paul Hoffman. *The Man Who Loved Only Numbers: The Story of Paul Erdos and the Search for Mathematical Truth*. Hyperion, New York, first edition, 1998.

[64] INRA and Jean-Benoist Leger. *blockmodels: Latent and Stochastic Block Model Estimation by a 'V-EM' Algorithm*, 2015. URL `https://CRAN.R-project.org/package=blockmodels`. R package version 1.1.1.

[65] Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544, 2016.

[66] Milan Janosov. Network Science Predicts Who Dies Next in Game of Thrones, July 2017. URL `https://networkdatascience.ceu.edu/article/2017-07-08/network-science-predicts-who-dies-next-game-thrones`. Accessed: 22/6/2018.

[67] Natalie Kalin. Top 10 Most Watched TV Finales Ever. *Huffington Post*, 2015. URL `https://www.huffingtonpost.com/natalie-kalin/top-ten-most-watched-tv-finales-ever_b_6760238.html?ec_carp=4255013076915788504`.

[68] Frigyes Karinthy. Chains (láncszemek). Short story, 1929.

[69] Natalie Kerracher, Jessie Kennedy, and Kevin Chalmers. The Design Space of Temporal Graph Visualisation. In *EuroVis*, volume 14, pages 7–11, 2014.

[70] Andrey N. Kolmogorov. Three Approaches to the Quantitative Definition of Information. *Problems of Information Transmission*, 1965.

[71] Molly Leecaster, Damon J. A. Toth, Warren B. P. Pettey, Jeanette J. Rainey, Hongjiang Gao, Amra Uzicanin, and Matthew Samore. Estimates of Social Contact in a Middle School Based on Self-Report and Wireless Sensor Data. *PLoS ONE*, 11(4), 2016. doi: 10.1371/journal.pone.0153690.

[72] Zhongyang Li, Xiao Ding, and Ting Liu. Constructing Narrative Event Evolutionary Graph for Script Event Prediction. arXiv:1805.05081v2 [cs.AI], May 2018.

[73] Benjamin Lind. Lessons on exponential random graph modeling from Grey's Anatomy hook-ups, September 2012. URL `http://badhessian.org/2012/09`. Accessed: 1/6/2017.

[74] Markus Luczak-Roesch, Adam Grener, and Emma Fenton. Not-so-distant reading: A dynamic network approach to literature. *De Gruyter Oldenbourg*, 60(1):29–40, 2018.

[75] Pádraig Mac Carron and Ralph Kenna. Universal properties of mythological networks. *EPL (Europhysics Letters)*, 99(28002), July 2012.

[76] Pádraig Mac Carron and Ralph Kenna. Viking sagas: Six degrees of Icelandic separation Social networks from the Viking era. *Significance*, 10(6):12–17, 2013.

[77] Ajaykumar Manivannan, W. Quin Yow, Roland Bouffanais, and Alain Barrat. Are the different layers of a social network conveying the same information? *EPJ Data Science*, 7(34), 2018.

[78] Mahendra Mariadassou, Stéphane Robin, and Corinne Vacher. Uncovering latent structure in valued graphs: A variational approach. *The Annals of Applied Statistics*, 4(2):715–742, June 2010.

[79] Lisa M. Marshall. *"I'll be there for you" if you are just like me: an analysis of hegemonic social structures in "Friends"*. PhD thesis, Graduate College of Bowling Green State University, August 2007.

[80] Rossana Mastrandrea, Julie Fournet, and Alain Barrat. Contact Patterns in a High School: A Comparison between Data Collected Using Wearable Sensors, Contact Diaries and Friendship Surveys. *PLoS ONE*, 10(9), September 2015. doi: 10.1371/journal.pone.0136497.

[81] Kevin M. McFarland. The Force Awakens and A New Hope Are More Similar Than You Think, April 2016. URL `https://www.wired.com/2016/03/mapping-star-wars-force-awakens-characters#slide-2`. Accessed: 27/3/2017.

[82] Stanley Milgram. The Small-World Problem. *Psychology Today*, 1(1): 61–67, May 1967.

[83] Semi Min and Juyong Park. Mapping Out Narrative Structures and Dynamics Using Networks and Textual Information. arXiv:1808.06945v1 [cs.CL], April 2016.

[84] Ben Misch. Game of Nodes: A Social Network Analysis of Game of Thrones, May 2015. URL `https://gameofnodes.wordpress.com/2015/05/06/game-of-nodes-a-social-network-analysis-of-game-of-thrones`. Accessed: 5/3/2018.

[85] Anders Mollgaard, Ingo Zettler, Jesper Dammeyer, Mogens H. Jensen, Sune Lehmann, and Joachim Mathiesen. Measures of Node Similarity in Multilayer Networks. *PLoS ONE*, 11(6), June 2016. doi: 10.1371/journal.pone.0157436.

[86] Enys Mones, Arkadiusz Stopczynski, and Sune Lehmann. Contact activity and dynamics of the social core. *EPJ Data Science*, 6(6), 2017. doi: 10.1140/epjds/s13688-017-0103-y.

[87] Marcelo A. Montemurro and Damián H. Zanette. Entropic analysis of the role of words in literary texts. *Advances in Complex Systems*, 5(1): 7–17, 2002.

[88] Youssef Mourchid, Benjamin Renoust, Hocine Cherifi, and Mohammed El Hassouni. Multilayer Network Model of Movie Script. In Luca Maria Aiello, Chantal Cherifi, Hocine Cherifi, Renaud Lambiotte, Pietro Lió, and Luis M. Rocha, editors, *Studies in Computational Intellegence*, volume 812 of *Complex Networks and Their Applications VII*, pages 782–796. Complex Networks, Springer, Cham, 2019.

[89] Chang-Jun Nan, Kyung-Min Kim, and Byoung-Tak Zhang. Social Network Analysis of TV Drama Characters via Deep Concept Hierarchies. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 831–836. IEEE, 2015.

[90] Mark E. J. Newman. Finding community structure using the eigenvectors of matrices. *Physical Review E*, 74(3), 2006.

[91] Mark E. J. Newman. *Networks*. Oxford University Press, United Kingdom, 2nd edition, 2018.

[92] Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 2004.

[93] Pascal Pons and Matthieu Latapy. Computing Communities in Large Networks Using Random Walks. In Yolum, T. Güngör, F. Gürgen, and C. Özturan, editors, *Lecture Notes in Computer Science*, volume 3733, pages 284–293. ISCIS 2005, 2005.

[94] Sandra D. Prado, Silvio R. Dahmen, Ana L. C. Bazzan, Padraig Mac Carron, and Ralph Kenna. Temporal Network Analysis of Literary Texts. *Advances in Complex Systems*, 19(03), May 2016. ISSN 1793-6802. doi: 10.1142/s0219525916500053. URL http://dx.doi.org/10.1142/S0219525916500053.

[95] Paulo Quaglio. *Television Dialogue: The sitcom Friends vs. natural conversation*, volume 36. John Benjamins Publishing, 2009.

[96] Usha N. Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 2007.

[97] Andrew J. Reagan, Lewis Mitchell, Dilan Kiley, Christopher M. Danforth, and Peter Sheridan Dodds. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(1):31, 2016.

[98] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1), 2006.

[99] Kay Richardson. *Television Dramatic Dialogue*. Oxford Studies in Sociolinguistics. Oxford University Press, New York, 2010.

[100] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39 (4):1878–1915, 2011.

[101] Ina Rösiger, Sarah Schulz, and Nils Reiter. Towards Coreference for Literary Text: Analyzing Domain-Specific Phenomena. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 129–138, 2018.

[102] Osvaldo A. Rosso, Hugh Craig, and Pablo Moscato. Shakespeare and other English Renaissance authors as characterized by Information Theory complexity quantifiers. *Physica A: Statistical Mechanics and its Applications*, 388(6):916–926, March 2009.

[103] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[104] Todd Schneider. The Simpsons by the Data, September 2016. URL `http://toddwschneider.com/posts/the-simpsons-by-the-data`. Accessed: 17/5/2017.

[105] David Schoch. Friends and Hypergraphs: The One With All The Networks, March 2015. URL `http://mildlyscientific.schochastics.net/2015/03/03/friends-and-hypergraphs-one-with-a`. Accessed: 7/6/2017.

[106] Shasa A. Schriber, Nam W. Kim, Hanspeter Pfister, and Markus Gross. Narrative Visualization System. Patent, August 2018.

[107] Vedran Sekara and Sune Lehmann. The Strength of Friendship Ties in Proximity Sensor Data. *PLoS ONE*, 9(7), July 2014. doi: 10.1371/journal.pone.0100915.

[108] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. arXiv:1507.04808v1 [cs.CL], July 2015.

[109] Srinivas Shakkottai, Marina Fomenkov, Ryan Koga, Dmitri Krioukov, and Kc Claffy. Evolution of the Internet AS-Level Ecosystem. In *Complex Sciences*, volume 5 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 1605–1616, Berlin, Heidelberg, 2009. Springer.

[110] Lucius A. Sherman. *Analytics of Literature: A Manual for the Objective Study of English Prose and Poetry*. Ginn and Company, Boston, USA, 1893.

[111] Julia Silge and David Robinson. *Text Mining with R*, May 2007.

[112] Giora Simchoni. The One With Friends, June 2017. URL http://giorasimchoni.com/2017/06/04/2017-06-04-the-one-with-friends. Accessed: 7/6/2017.

[113] Abhishek K. Singh. Neural Approaches Towards Text Summarization. Master's thesis, International Institute of Information Technology, Hyderabad, Hyderabad - 500 032, India, July 2018.

[114] Jeremy Sklarsky. Lies, Damn Lies, and West Wing Statistics, March 2017. URL https://medium.com/@jeremy.sklarsky/lies-damn-lies-and-west-wing-statistics-9473219434a9. Accessed: 11/12/2018.

[115] Timo Smieszek, Stefanie Castell, Alain Barrat, Ciro Cattuto, Peter J. White, and Gérard Krause. Contact diaries versus wearable proximity sensors in measuring contact patterns at a conference: method comparison and participants' attitudes. *BMC Infectious Diseases*, 16(341), 2016. doi: 10.1186/s12879-016-1676-y.

[116] Sucheta Soundarajan, Acar Tamersoy, Elias B. Khalil, Tina Eliassi-Rad, Duen H. Chau, Brian Gallagher, and Kevin Roundy. Generating graph snapshots from streaming edge data. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 109–110. International World Wide Web Conferences Steering Committee, 2016.

[117] Adam Sternbergh. Is 'Friends' Still the Most Popular Show on TV? *New York Magazine*, March 2016. URL https://www.vulture.com/2016/03/20-somethings-streaming-friends-c-v-r.html.

[118] Scott Stoltzman. Seinfeld Characters – A Post About Nothing, December 2016. URL https://www.stoltzmaniac.com/2016/12. Accessed: 1/6/2017.

[119] Michael Szella, Renaud Lambiotteb, and Stefan Thurner. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences*, 107(31):13636–13641, August 2010.

[120] Melody S. A. Tan, Ephrance A. Ujum, and Kurunathan Ratnavelu. A character network study of two Sci-Fi TV series. In *AIP Conference Proceedings*, volume 1588, pages 246–251. American Institute of Physics, 2014.

[121] The Acropolitan. Sentence Length Has Declined 752017. URL https://medium.com/@theacropolitan. Accessed: 11/12/2018.

[122] Philippe Thoiron. Diversity Index and Entropy as Measures of Lexical Richness. *Computers and the Humanities*, 20(3):197–202, July 1986.

[123] Guido Van Rossum and Fred L. Drake Jr. *Python reference manual*, 1995.

[124] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[125] Michaël C. Waumans, Thibaut Nicodème, and Hugues Bersini. Topology Analysis of Social Networks Extracted from Literature. *PLoS ONE*, 10(6), June 2015.

[126] Chung-Yi Weng, Wei-Ta Chu, and Ja-Ling Wu. Rolenet: Treat a movie as a small society. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 51–60. ACM, 2007.

[127] Chung-Yi Weng, Wei-Ta Chu, and Ja-Ling Wu. Rolenet: Movie Analysis from the Perspective of Social Networks. *IEEE Transactions on Multimedia*, 11(2):256–271, 2009.

[128] Krist Wongsuphasawat. How every #GameOfThrones episode has been discussed on Twitter, May 2016. URL https://interactive.twitter.com/game-of-thrones/#?episode=1. Accessed: 1/6/2017.

[129] Shirley Wu. An Interactive Visualization of An Interactive Visualisation of Every Line in Hamilton, March 2017. URL https://pudding.cool/2017/03/hamilton/index.html. Accessed: 1/6/2017.

[130] Jingjing Xu, Yi Zhang, Qi Zeng, Xuancheng Ren, Xiaoyan Cai, and Xu Sun. A Skeleton-Based Model for Promoting Coherence Among Sentences in Narrative Story Generation. arXiv:1808.06945v1 [cs.CL], August 2018.