МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра Вычислительной техники



**ОТЧЁТ**

**по Расчетно Графическому Заданию**

**по дисциплине: «***Компьютерные технологии анализа и обработки данных***»**

Выполнил:Проверил:

Студенты гр. «АММ2-21», «АВТФ»  
 *«Арнольд Э.В»* *«*к.т.н., доцент*»*

*«Антонов С.С» «Альсова О.К.»*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск

2021

# Анализ сетей в R

*Наука о сетях* (network science) является широким научным подходом, который использует анализ взаимосвязей для изучения и интерпретации биологических, физических, социальных и информационных систем. Основной инструмент для специалистов по сетям – это анализ сетей, который представляет собой набор методов, использующихся для (1) визуализации сетей, (2) описания определенных характеристик структуры сети в целом, а также получения детальной информации об отдельных узлах, связях и подгруппах внутри сетей, (3) создания математических и статистических моделей сетевых структур и сетевой динамики. Поскольку основной вопрос, интересующий науку о сетях, касается взаимосвязей, большинство методов, используемых в анализе сетей, сильно отличается от более традиционных статистических инструментов, используемых социологами и учеными-медиками.

*Анализ сетей* (network analysis)– это отдельное научное направление со своими

собственными теориями и методами, взятыми из других дисциплин, в частности

из теории графов и топологии в математике, анализа систем родственных связей

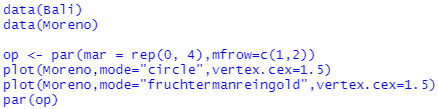
в антропологии, анализа социальных групп и процессов в социологии и психологии.

В данной работе будем рассматривать только визуализацию сетей.

# Визуализация сетей

Анализ сетей начинается с построения и исследования графика сети. Главное предназначение графика сети (как и любой графической информации) состоит в том, чтобы выделить важную информацию, содержащуюся в исходных данных. Однако существует множество способов скомпоновать узлы и связи сети в двумерном пространстве, а также использовать графические элементы (например, размер узла, цвет линии, легенду и т. д.) для представления сетевых данных.

Эффективный график сети показывает важную информацию о социальной сети, например, об общей структуре, расположении ключевых акторов, наличии отдельных подгрупп и т. д. Одновременно с этим график должен максимально минимизировать присутствие нерелевантной информации. Например, длина связи в диаграмме сети является произвольной в том смысле, что она не значима. Эффективный рисунок сети должен быть спроектирован и выведен так, чтобы свести к минимуму вероятность того, что зритель неправильно истолкует значения длины связей.



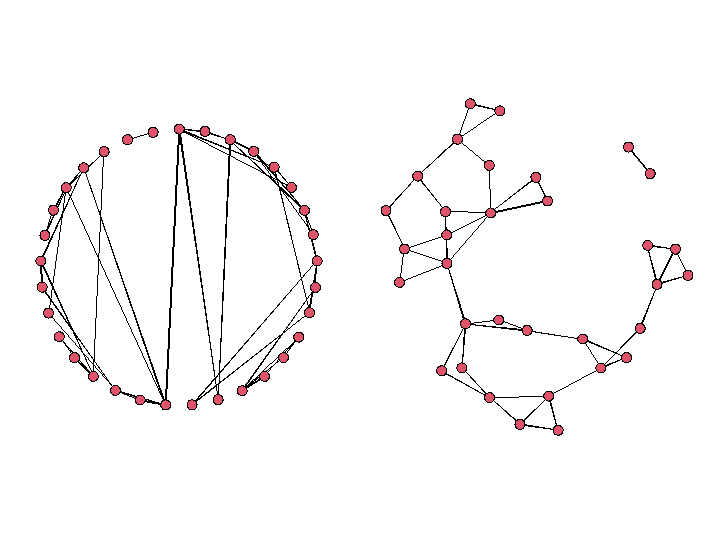


Рис.1 Различные укладки одной сети

Далее мы познакомимся с основными способами графического представления сетей в R и рассмотрим различные параметры для настройки укладки сети (network layout) на экране или странице.

Пример с укладкой сети показывает, как интерпретацию графика сети можно усложнить или упростить с помощью выбора той или иной укладки.

На первый взгляд, может показаться, что на рисунке представлены две совершенно различные сети. Фактически же перед нами – две различные визуализации

одной и той же социальной сети, в данном случае сети дружеских контактов между

учениками 4-го класса.

Несмотря на то что на рисунке представлены одни и те же данные, легко заметить, что график справа интерпретировать проще. В частности, гораздо легче увидеть, что сеть составлена из двух отдельных компонент, а бóльшая по размеру компонента состоит из двух довольно четких подгрупп. Важные структурные характеристики сети легче определить на графике справа.

Несмотря на то что можно разместить сеть в трехмерном пространстве, подавляющее большинство визуализаций сети является двухмерным. Узлы представлены в виде геометрических фигур, обычно в виде кружков, а связи – в виде прямых или иногда изогнутых линий.

Для человека, малознакомого с методами визуализации сети, интерпретация линий может быть сложной. В частности, длина линии не имеет никакого значения. Рассмотрим следующие два графика, которые выводят на экран одну и ту же простую сеть (рис. 4.2).

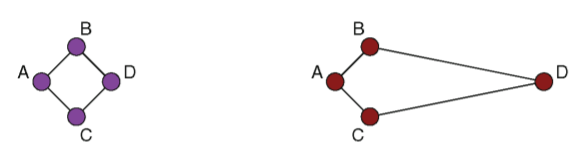


Рис. 2. Длина линии может быть произвольна

На первый взгляд, может показаться, что на графике справа узел D удален от узлов B и C на большее расстояние, чем на графике слева. Но связи просто указывают на то, какие узлы смежны (adjacent) друг с другом, таким образом, длина линии не несет какой-либо

существенной информации.

Однако, как и в случае с сетью дружеских контактов на первом рисунке, несмотря на произвольный характер некоторых элементов укладки, способ графического представления сети может прояснить или исказить важную информацию о ее

структуре.

## Эстетический вид укладок сетей

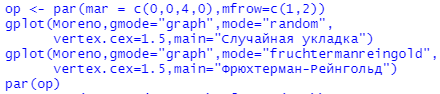
Теоретически количество вариантов отображения сети на экране конечно, но практически неотличимо от бесконечности. В большинстве случаев будут построены уродливые или сбивающие с толку укладки, поэтому возникает необходимость в такой укладке, которая с высокой долей вероятности будет адекватно представлять сеть.

Для того чтобы сделать укладку более понятной был сформулирован ряд эстетических принципов, которые могут использоваться для построения более эффективных графиков сетей. Графики сетей будет проще понять, если при их построении придерживаться следующих пяти принципов:

* минимизировать пересечения ребер;
* максимизировать симметричность укладки узлов;
* минимизировать изменчивость длины ребер;
* максимизировать угол между ребрами, когда они пересекают или соединяют узлы;
* минимизировать общее пространство, использованное для вывода сети.

Для автоматического построения укладок было разработано большое количество методов. Общий класс алгоритмов под названием *force-directed algorithms* (силовые алгоритмы) оказался гибким и мощным методом автоматической укладки сетей. Эти алгоритмы работают итеративно, чтобы уменьшить общую энергию сети, где энергия может быть определена различными способами. Общий подход заключается в том, что между узлами, соединенными ребром, действуют силы притяжения («пружины»), при этом одновременно между всеми парами узлов действуют силы отталкивания. Пружины (springs) в этом алгоритме стараются максимально сблизить смежные узлы, тогда как силы отталкивания максимально удаляют несмежные узлы друг от друга. Получающаяся в результате сеть будет некоторое время меняться, прежде чем обрести устойчивое состояние, при котором произойдет минимизация энергии сети.

Оценить положительные результаты применения одного из этих алгоритмов можно на рис. 3.



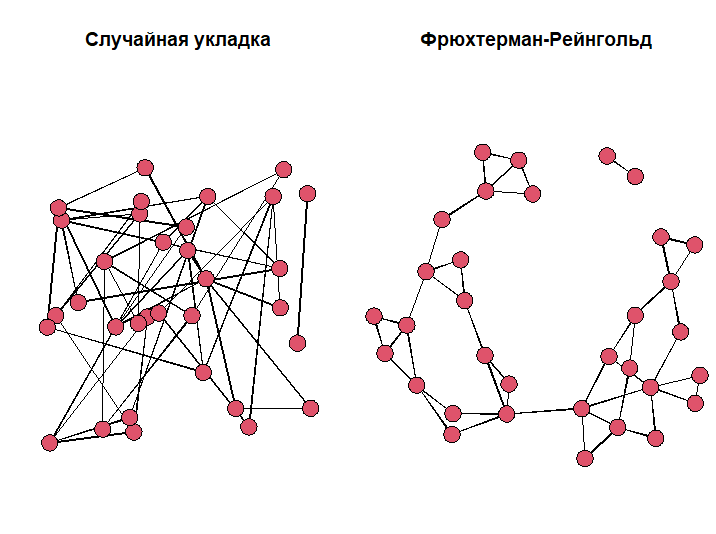


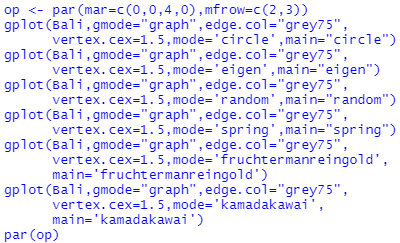
Рис.3 Сеть Moreno, случайная укладка и укладка Фрюхтермана–Рейнгольда

Слева сеть Moreno в случайной укладке. Справа для визуализации сети был использован алгоритм укладки Фрюхтермана и Рейнгольда. Графические функции в комплекте пакетов statnet используют этот алгоритм по умолчанию. Справа узлы уложены более симметрично, мы видим меньшее количество пересечений ребер, и длина связей распределена более равномерно. Все это упрощает интерпретацию структуры сети.

Алгоритм Фрюхтермана–Рейнгольда наряду с другими силовыми методами является итеративным и недетерминированным. Это означает, что каждый раз, запуская алгоритм визуализации, вы не получите точно такой же укладки. Однако вы получите укладку, которая, как правило, симметрична, имеет минимальное количество пересечений ребер и т. д.

# Основные алгоритмы и методы графического представления

Визуализация сети в statnet осуществляется с помощью двух тесно связанных между собой функций plot и gplot. Последняя предлагает большее количество алгоритмов укладки, таким образом, она, возможно, будет более полезной. Чтобы использовать тот или иной алгоритм укладки, нужно задать соответствующую укладку. Рисунок показывает шесть алгоритмов укладки для функции gplot.



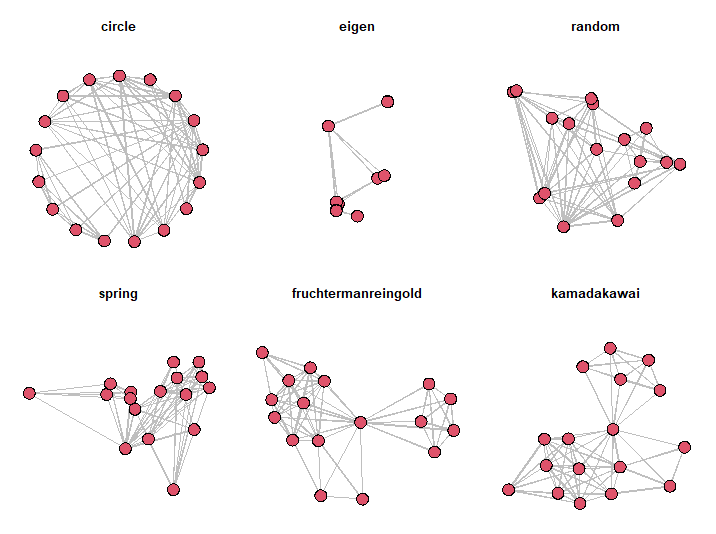
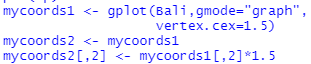


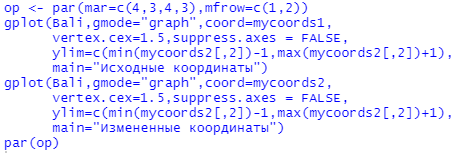
Рис. 4. Алгоритмы укладки сетей

## Более точная настройка укладки сети

Алгоритмы укладки, представленные в statnet и igraph, работают эвристически, с некоторой долей случайности. Даже выбрав один и тот же алгоритм укладки, при каждом построении графика сети мы будем получать разную укладку. Но, в R можно точно настроить координаты укладки. Это позволяет добиться точного позиционирования или сохранить координаты укладки после того, как построен конкретный график сети.

Для этого используется параметр coord в функции gplot. Этому параметру нужна матрица с двумя столбцами. Каждая строка соответствует узлу, первый столбец задает координату X, а второй столбец – координату Y. Результаты, полученные с помощью функции gplot, можно сохранить в объекте. Объект хранит координаты построенного графика.





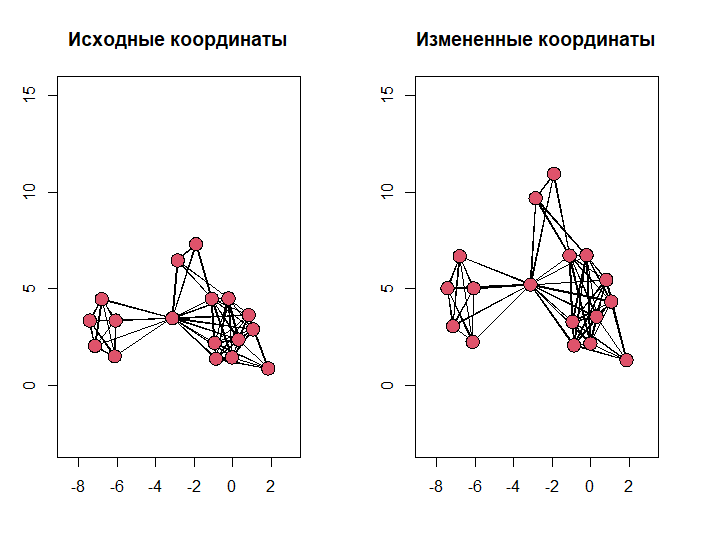
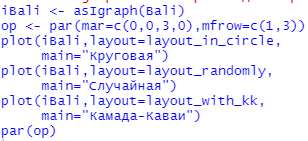


Рис.5 Укладки сетей с измененными координатами

На данном рисунке мы строим первоначальный график сети, сохраняя координаты. Затем, растягиваем граф путем умножения координат Y на константу. Оба графика показаны на рисунке, также приводятся оси, чтобы проще было увидеть, как изменились координаты. Существуют и другие способы задать определенные координаты, но основное применение заключается в том, чтобы сохранить конкретную укладку для построения и анализа графика в будущем.

## Укладки сетей, построенные с помощью igraph

Пакет igraph предлагает пользователю аналогичный набор параметров для настройки укладки сети. Параметр layout используется, чтобы задать алгоритм укладки или указать набор координат вершин.



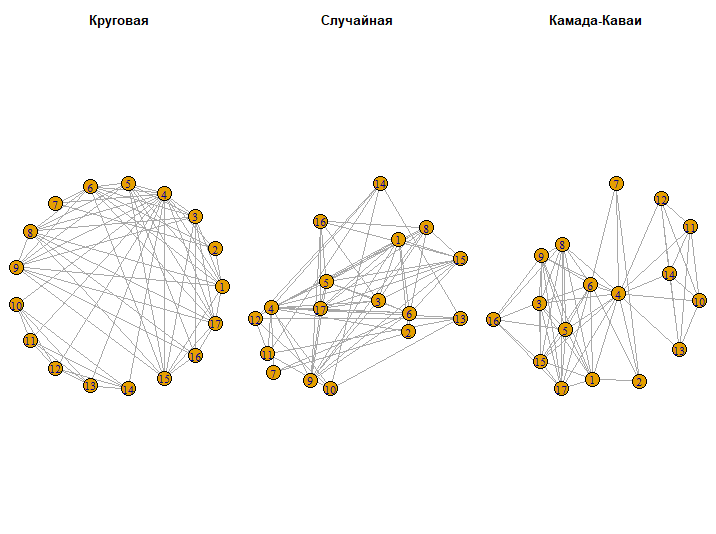


Рис.6 Алгоритмы укладки сетей в igraph

# Эффективный графический дизайн сетей

Создание эффективного графического дизайна сети мало чем отличается от какого-либо другого типа графической информации. «Графическое превосходство состоит в том, чтобы дать зрителю максимальное количество информации в кратчайший срок, используя наименьшее количество чернил на минимальном пространстве» (Эдвард Тафти). У графиков сетей изначально есть важное преимущество в том плане, что они обычно характеризуются высоким соотношением объема информации к объему чернил.

Цель графического дизайна сети должна состоять в том, чтобы построить рисунок, который показывает важную или интересную информацию, содержащуюся в сетевых данных. Для этого аналитик должен определиться с каждым графическим элементом, который может появиться на рисунке. R и графические функции в statnet и igraph предлагают аналитику практически полное программное управление графиками сетей.

Далее будет кратко рассказано о наиболее важных элементах графического дизайна сетей и обсуждено, как их использовать и почему их нужно применять определенным образом.

Как и любой другой тип графической информации, визуализация сети состоит из большого количества отдельных графических элементов. Эти отдельные элементы являются отличительными для того или иного графика сети, например, узлы и связи, а также другие элементы, характерные для большинства графиков, например заголовки, легенды и т. д. Графические функции в statnet и igraph предлагают пользователю полноценное программное управление графиками.

Несмотря на то что для построения обычного графика сети достаточно простого вызова функции графического изображения, практически всегда имеет место ситуация, когда нужно потратить время на настройку соответствующих параметров функций и написать дополнительный программный код R для построения эффективного графика.

Мы кратко познакомимся с графическими элементами, наиболее часто используемыми для визуализации сети. О каждом элементе будет рассказано по порядку.

## Цвет узла

По умолчанию statnet строит график сети, в котором узлы представлены в виде красных кружков. Чтобы задать цвет, используется параметр vertex.col в функции gplot (также используется параметр gmode, который задает тип графа)



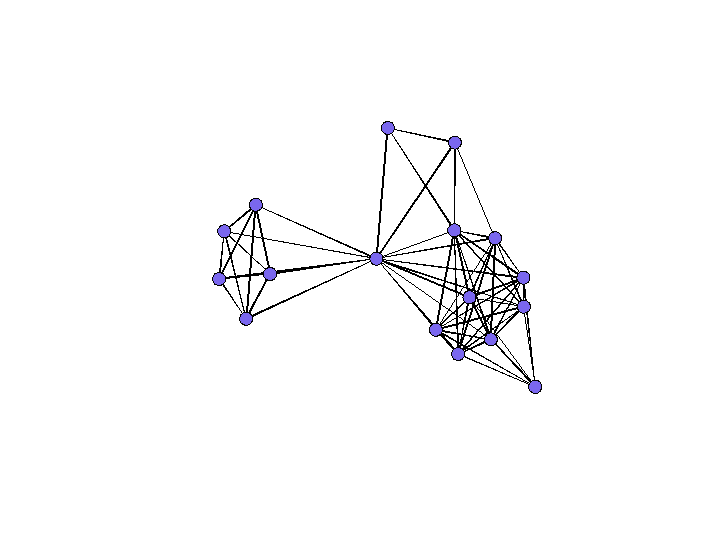


Рис. 7. Сеть Bali со светло-синими узлами

В целом с точки зрения графического представления сетей в R есть все основные параметры для работы с цветом. Цвет может определяться его названием. Чтобы взглянуть на все 657 названий цветов, распознаваемых R, воспользуйтесь командой colors(). Кроме того, цвета можно назначить с помощью RGB-триплетов1 . Помимо этого, цветовую схему RGB можно задать с помощью шестнадцатеричной строки в формате #RRGGBB, где RR, GG и BB – это шестнадцатеричные числа, которые задают интенсивности красной, зеленой и синей составляющих цвета в диапазоне от 00 до FF.

Одна из самых редко используемых настроек для работы с цветом может пригодиться для построения диаграмм больших сетей, где узлы накладываются друг на друга. Обычно цвета полностью непрозрачны, поэтому перекрывающиеся узлы в диаграмме выглядят как огромные цветовые «сгустки» («blobs»), где трудно выделить какие-либо узлы. Однако есть возможность применить частично прозрачные цвета, использовав встроенный альфа-канал прозрачности. Функцию rgb() можно использовать для определения значения прозрачности от 0 (полностью прозрачный) до 1 (полностью непрозрачный).

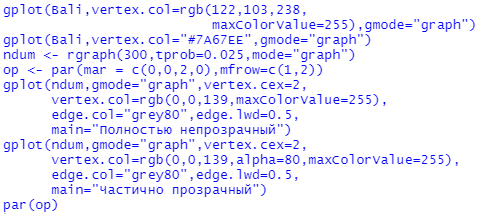


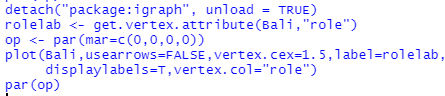


Рис.8 Пример использования альфа-канала прозрачности

Рисунок 8 показывает разницу, которая получается при использовании альфа-канала прозрачности. Оба графика представляют собой одну и ту же случайную сеть, состоящую из 300 узлов. (Укладки отличаются, потому что использовался силовой алгоритм Фрюхтермана–Рейнгольда.) На рисунке слева используется полностью непрозрачный темно-синий цвет. Рисунок справа тоже использует темно-синий цвет, но со значением альфа-канала, равным примерно 30%. Перекрывающиеся узлы становится намного проще увидеть, когда используются прозрачные цвета.

. Более важная задача, решаемая с помощью цвета, заключается в том, чтобы показать определенную характеристику узла или сети, присвоив различным узлам разные цвета. В частности, информацию, хранящуюся в категориальном атрибуте узла, можно передавать с помощью цвета узла.

. Например, сеть террористов Bali имеет атрибут вершин role, в котором хранится категориальная информация о роли каждого участника сети. CT означает член командной группы, BM – изготовитель бомб и т. д. Цвет узла можно использовать для эффективного различения ролей участников сети. Поскольку эта информация уже хранится в атрибуте вершин, statnet может использовать ее для автоматического выбора цвета узла. (Это верно лишь для plot(), но не для gplot().)



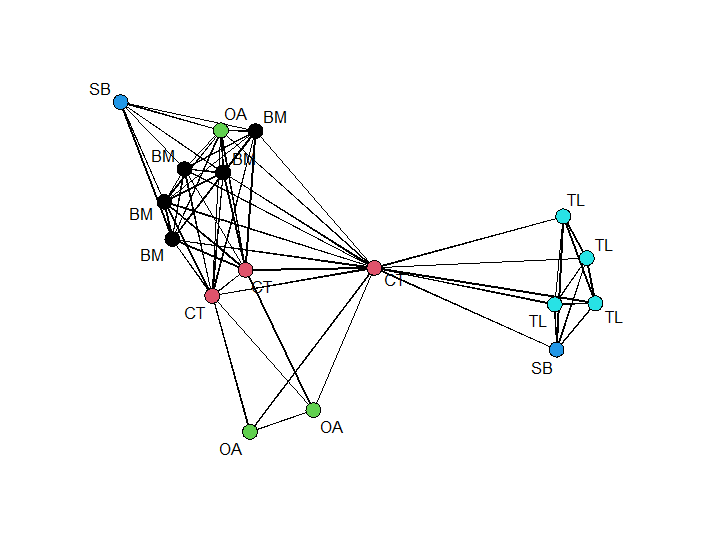


Рис.9 Сеть Bali, роли участников выделены разными цветами

При этом, используя имя заданного атрибута вершин, statnet назначает цвета узлов, согласно текущей палитре цветов, принятой по умолчанию в R. Просматривая эту палитру, мы видим, что «BM» назначается черный цвет, потому что «BM» стоит первым по алфавиту в атрибуте role, а black – первая запись в палитре цветов. «CT» стоит вторым, поэтому назначается красный цвет, который является второй записью в палитре, и т. д.

Использование палитры, принятой по умолчанию, имеет много недостатков. Во-первых, выбор ограничен восемью цветами. (R циклически повторит набор из восьми цветов, если типов узлов, для которых нужно задать цвет, будет больше восьми.) Во-вторых, палитра, принятая по умолчанию, начинается с черного цвета, который, как правило, является не самым лучшим цветом для графика сети.

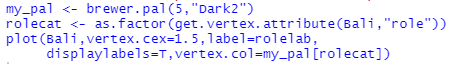
Более гибкий и оптимальный, с эстетической точки зрения, подход заключается в том, чтобы задать собственную цветовую палитру и индекс для выбора цвета. Пакет RColor Brewer предлагает ряд заранее разработанных цветовых палитр, которые очень удобны, если цвет используется для идентификации относительно небольшого количества категорий. Больше подробную информацию о ColorBrewer можно получить по адресу <http://www.colorbrewer.org>. (Приводится информация о различных параметрах выбора цветов и палитр, например посмотрите интерактивный генератор палитр по адресу <http://paletton.com>.)





Рис.10 Набор из пяти цветов, взятых из палитры RColorBrewer

Пользовательская палитра создается путем выбора пяти цветов из палитры большего размера под названием Dark2, входящей в RColorBrewer. Как только палитра задана, ее можно использовать для построения графика сети. Этот подход позволяет получить более приятные цвета и является гораздо более гибким, в отличие от ситуации, когда зависишь от палитры цветов, принятой по умолчанию. Обратите внимание на то, что мы преобразуем символьный вектор атрибута вершин role в фактор, чтобы воспользоваться индексацией.



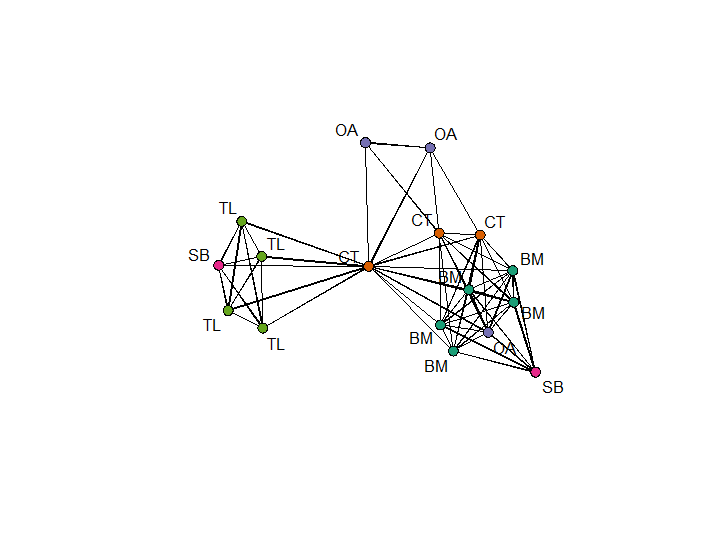
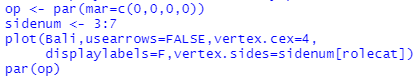


Рис.11 Сеть Bali с более оптимальными цветами узлов

## Форма узла

Помимо использования цвета для идентификации различных типов узлов, в statnet можно задавать разные формы узлов. В основном это удобно тогда, когда используется небольшое количество типов узлов. Кроме того, это особенно удобно, когда у вас нет возможности использовать цвет для идентификации узлов (или помогать дальтоникам).

К сожалению, statnet предлагает скромные возможности по идентификации узлов с помощью формы, автоматически определяя количество сторон, необходимых для построения многоугольника узла (обычно количество сторон равно 50 и строится круг). Если количество сторон равняется 3, вы получаете треугольник, 4 – квадрат и т. д. Это удобно лишь при работе с несколькими типами узлов:



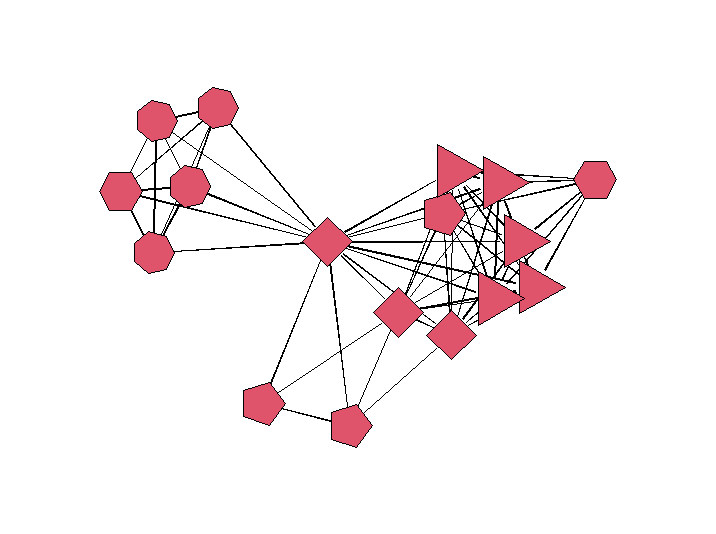
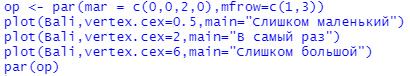


Рис.12 Сеть Bali с различными формами узлов

Если у вас есть конкретная потребность использовать различные формы узлов для графика сети, пакет igraph в этом плане является гораздо более гибким.

## Размер узла

В функциях plot и gplot комплекта пакетов statnet размеры узлов сети задаются параметром vertex.cex (аналогичным образом настраиваются размеры графических элементов в базовой графической системе R). Определяя абсолютные размеры узлов1 , нужно придерживаться правила: узлы должны быть достаточно большими, чтобы их можно было отличить друг от друга, и при этом достаточно маленькими, чтобы они не перекрывали сильно друг друга.



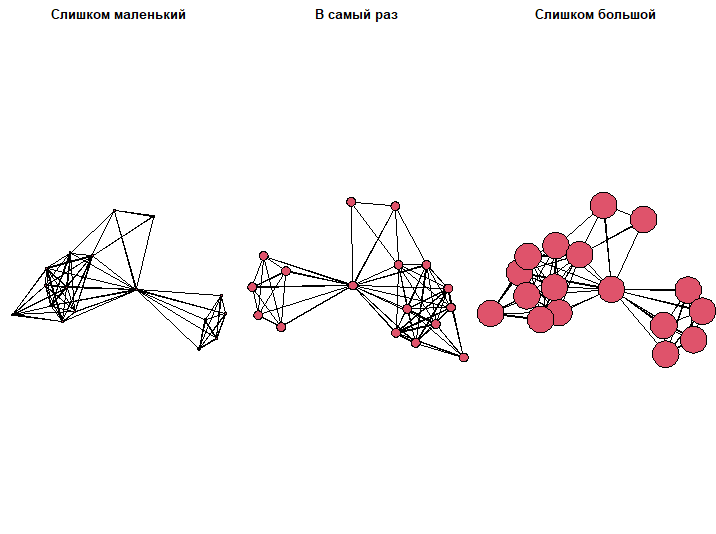
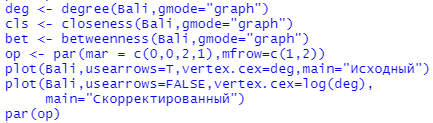


Рис.13 Корректировка абсолютного размера узла

Если у вас есть конкретная потребность использовать различные формы узлов для графика сети, пакет igraph в этом плане является гораздо более гибким.

Вместо того чтобы задавать одинаковый абсолютный размер для каждого узла, часто возникает необходимость использовать размер узла, чтобы отразить какуюту важную количественную характеристику. Например, узлы отличаются по своему расположению в сети. Некоторые узлы являются более центральными, тогда как другие являются более периферийными. Вычислим три различные меры центральности узла. Каждая из нижеперечисленных строк программного кода создает вектор мер центральности для каждого узла, и более высокие числовые значения указывают на бóльшую центральность.

Получив вектор, его можно использовать для определения относительных размеров узлов. Как и ранее, это делается с помощью того же самого параметра vertex.cex, но вместо конкретного числового значения мы задаем полученный вектор deg.



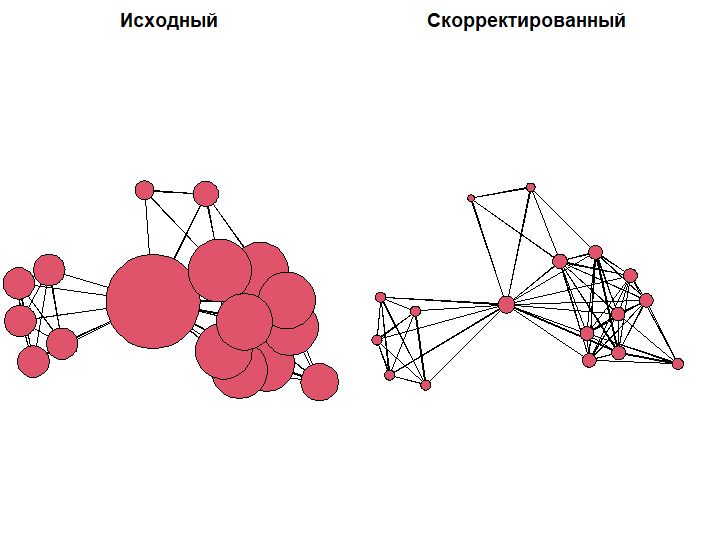
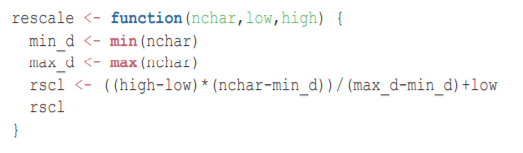


Рис. 14. Корректировка относительного размера узла

Взглянув на график слева, мы видим, что исходные числовые значения вектора deg (degree, или степень) строят слишком большие узлы. Их нужно скорректировать, и в данном случае мы получим подходящие размеры узлов, прологарифмировав значения вектора deg. В результате получаем оптимальный вариант, когда можно легко увидеть узлы с высокой степенью и узлы с низкой степенью.

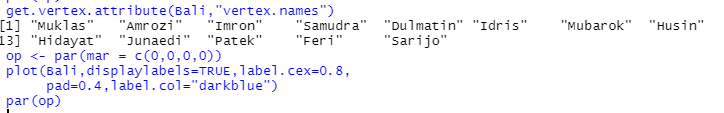
Несмотря на то что R действительно позволяет вам полностью настраивать размеры узлов, корректировка относительных размеров узлов может быть утомительна. Функцию, приведенную ниже, можно использовать, чтобы сэкономить определенное время, которое уходит на определение оптимальных размеров узлов. Функция rescale() берет вектор характеристик узла (им может быть любой числовой вектор) и масштабирует значения так, чтобы они лежали в интервале между значениями low и high.



## Метка узла

Как правило, график сети интереснее и проще интерпретировать, если узлы имеют метки и можно составить представление об участниках сети. Это особенно удобно при работе с небольшими сетями, если же сети становятся слишком большими, метки уже сами по себе могут затруднить интерпретацию сети.

Если объект-сеть в statnet имеет специальный атрибут вершин vertex.names, то его можно использовать для автоматического вывода меток узлов при построении графика. Можно настроить различные характеристики меток узлов, например размер шрифта, цвет и расстояние от узла:



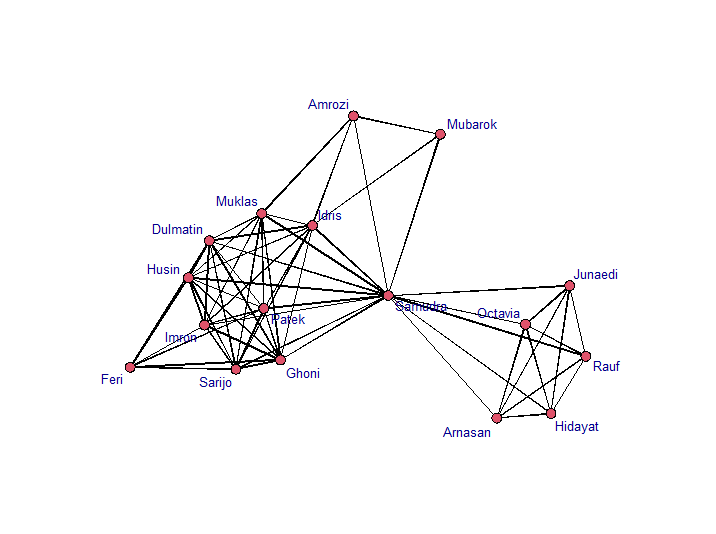


Рис.15 Сеть Bali с метками узлов

Автоматические метки, основанные на информации, хранящейся в атрибуте vertex.names, не всегда отражают самую важную или полезную информацию. Например, при анализе сети Bali подлинные имена террористов не очень интересны аудитории. К счастью, в качестве меток узлов можно использовать другую текстовую информацию. Используем текстовые значения, хранящиеся в атрибуте вершин role. Ключевой момент здесь заключается в использовании параметра label, который задает текстовый вектор для меток.



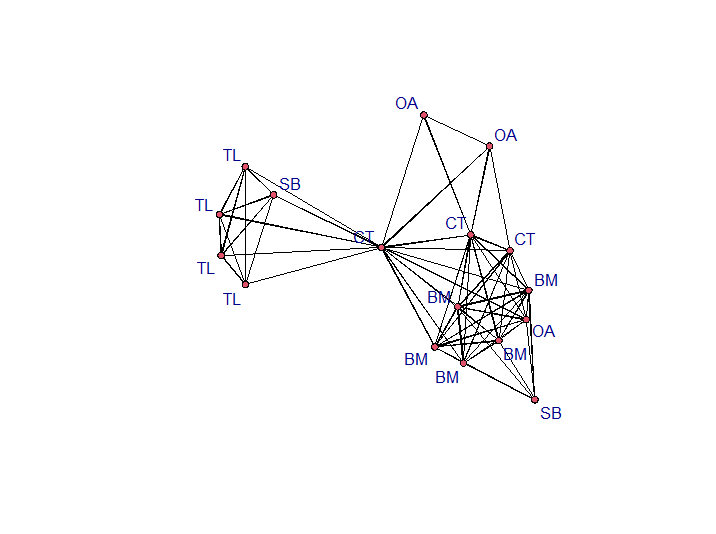
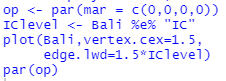


Рис. 16. Сеть Bali с метками-ролями

## Ширина ребра

Если сетевые данные предполагают связи с определенными значениями или вообще какую-либо количественную информацию, которая характеризует связи между узлами, можно отразить эту информацию визуально, изменив ширину связей в графике сети. Например, возможно, есть информация о силе дружеских связей или можно измерить денежную сумму, которая передается от организации к организации в направленной сети. В этих случаях более толстые связи могут означать более тесный контакт или более крупную денежную сумму



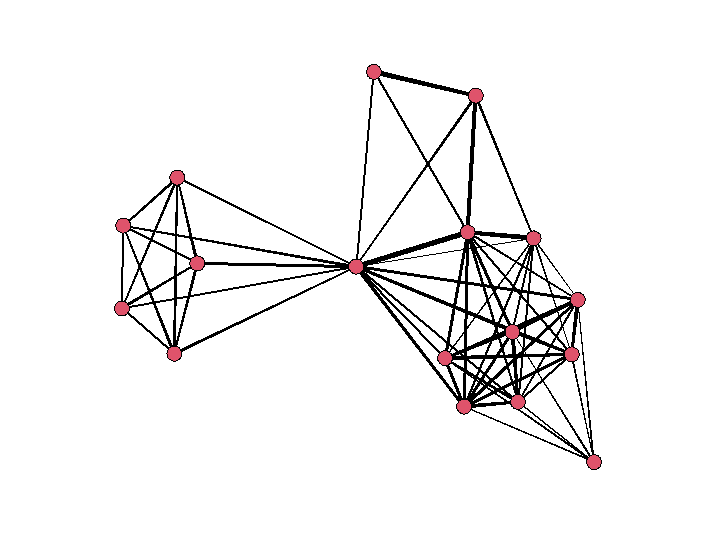
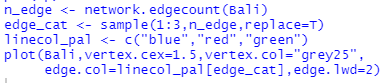


Рис. 17. Сеть Bali, ширина ребра обозначает тесноту контактов

Сеть Bali включает атрибут связей под названием IC, который представляет собой обычную пятиуровневую порядковую шкалу, использующуюся для измерения тесноты взаимодействия между участниками сети. Этот атрибут можно использовать для определения ширины связей. В примере, значения IC извлекаются из сохраненного атрибута ребер, это позволяет нам преобразовать вектор для более четкого разграничения пяти уровней IC (умножив значения вектора на 1,5).

## Цвет ребра

Если ширину ребра можно использовать для представления количественной информации о связях сети, то цвет ребра можно применить для вывода качественной информации о связях, подобным же образом используется цвет узла. Например, вы можете использовать различные цвета линий, чтобы выделить положительные и отрицательные связи в социальной сети.



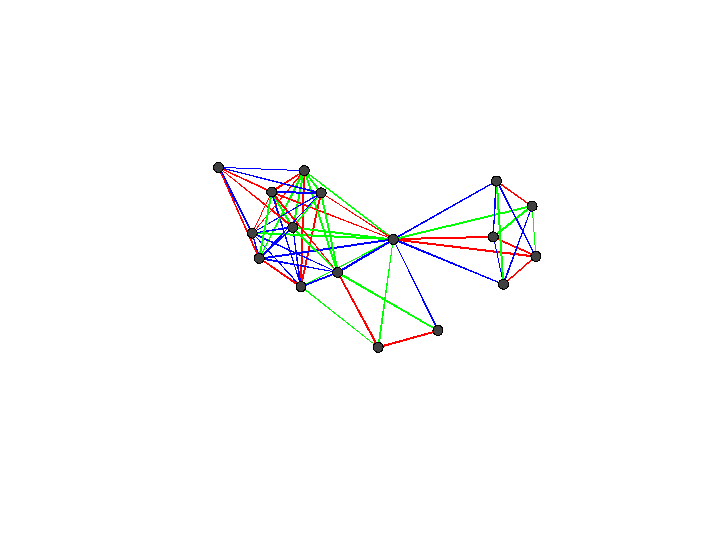


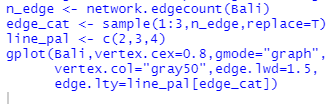
Рис.18 Сеть Bali с разными цветами ребер

В атрибуте ребер сети Bali не содержится категориальной или качественной информации, поэтому здесь мы создаем случайный категориальный вектор, чтобы показать, как можно использовать различные цвета ребер в графике сети. Для этого примера мы задаем палитру цветов, которую можно использовать для индексации нужных цветов, на основе категориального вектора ребер.

## Тип ребра

Если ширину ребра можно использовать для представления количественной информации о связях сети, то цвет ребра можно применить для вывода качественной информации о связях. Например, вы можете использовать различные типы линий, чтобы выделить положительные и отрицательные связи в социальной сети.

В атрибуте ребер сети Bali не содержится категориальной или качественной информации, поэтому здесь мы создаем случайный категориальный вектор, чтобы показать, как можно использовать различные типы ребер в графике сети. В данном случае используются три различных типа линии (2 = тире; 3 = точка; 4 = точка-тире). Кроме того, поскольку при использовании функции plot() различные типы линий не отображаются четко, здесь используется функция gplot():



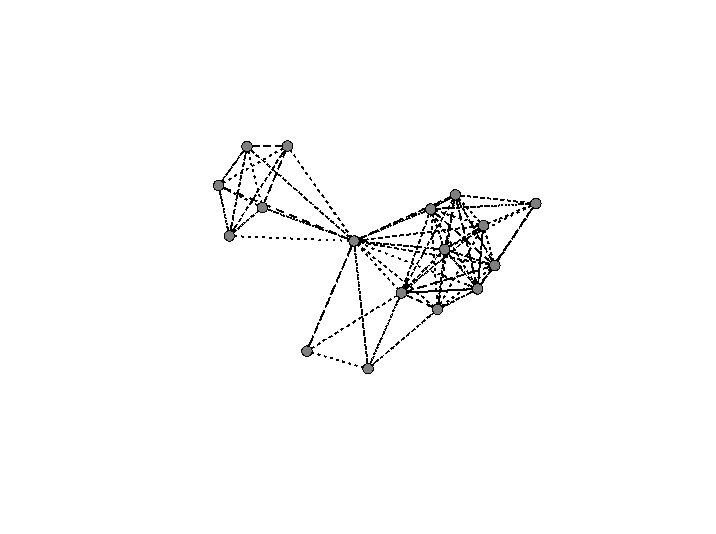
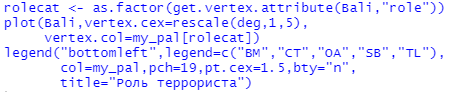


Рис.19 Сеть Bali с различными типами связей

## Легенды

Вышеприведенные примеры показывают, как графические элементы (цвет узла, форма узла, размер узла, тип ребра, ширина ребра) могут использоваться для отражения важных характеристик сети. Как и в случае с другими видами информационной графики, часто бывает полезно создать легенду, чтобы представленная информация стала понятна пользователю.

Базовые графические функции, имеющиеся в statnet, не могут автоматически создать легенду к графику. К счастью, чтобы добавить легенду к графику сети, можно просто воспользоваться функцией legend(), которая имеется в базовом пакете R:



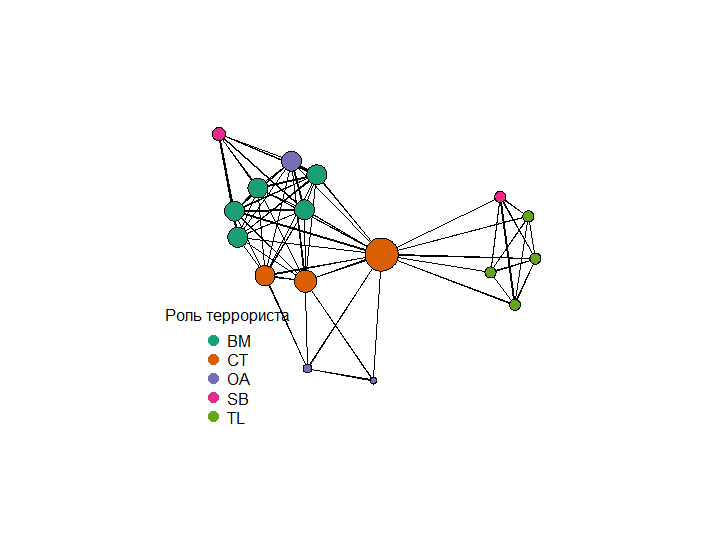


Рис.20 Сеть Bali с различными формами узлов

Если у вас есть конкретная потребность использовать различные формы узлов для графика сети, пакет igraph в этом плане является гораздо более гибким.

# Сложные графики сетей

Как было показано в двух предыдущих главах, statnet и igraph располагают широкими возможностями построения графиков, которые позволяют строить самые различные диаграммы сетей. Однако их графические функции не могут удовлетворить всех потребностей, связанных с анализом или визуализацией данных. В частности, специалистам по анализу сетей могут потребоваться более специализированные графики. Кроме того, хотя statnet и igraph как нельзя лучше подходят для построения высококачественных графиков, если эти графики статичны. К счастью, появилась возможность строить графики сетей и отправлять их в вебприложения, где пользователи могут уже самостоятельно настроить внешний вид диаграмм. Далее рассказывается о нескольких специализированных графиках сетей, а также показано, как построить простые интерактивные вебдиаграммы сетей

## Интерактивные графики сетей

Одна из полезных настроек различных пакетов для анализа сетей, таких как UCINet и Pajek, – это возможность построить графики сетей, которые являются в некотором смысле интерактивными. Например, в Pajek визуализацию сети можно построить в отдельном окне Draw, а затем пользователь может воспользоваться этим окном, для того чтобы различными способами отредактировать или изменить график сети. Эти возможности позволяют исследовать сети, а также точно настроить диаграмму сети для последующей публикации.

Несмотря на то что программная платформа R предлагает детальную настройку всех элементов диаграммы, эта настройка, как правило, не осуществляется интерактивным способом. Существует несколько исключений, а также есть несколько новых пакетов, которые позволяют строить интерактивные графики сети с последующей публикацией в Интернете.

Например, пакет igraph включает функцию tkplot(), которая поддерживает построение простых интерактивных графиков с помощью графического окна Tk. При этом можно изменить лишь некоторые настройки графиков сетей. Обычное использование этой функции заключается в построении интерактивного графика, корректировке позиций узлов для улучшения укладки сети, сохранении координат позиций узлов и построении итоговой (неинтерактивной) диаграммы сети на основе сохраненных координат.

## Специализированные диаграммы сетей

Традиционно диаграммы сетей строятся для иллюстрации фундаментальных характеристик узлов и сети, например важности. Однако существует ряд специализированных графиков, которые используются для выделения других значимых или интересных характеристик сетей. Мы рассмотрим три таких графика: дуговые диаграммы, хордовые диаграммы и теплокарты.

**Дуговые диаграммы** (arc diagrams) могут использоваться, когда структура связей представляет больший интерес, чем расположение узлов в сети. Для этого примера нам нужно загрузить все необходимые библиотеки, затем создать объект (список ребер) для функции arcplot().

Рис.21 Сеть контактов

**Хордовые диаграммы** (chord diagrams) – это особый тип графика, в котором для отображения взаимосвязей между узлами используется круговая укладка. Они стали особенно популярными в генетических исследованиях. Поскольку информацию о сети можно хранить в матрице, хордовые диаграммы представляют интерес при построении графиков сетей. Особенно это верно для взвешенных (т. е. имеющих числовые значения) и направленных сетей, где сумма и направление «потоков» представляют интерес для исследователя.

В пакете circlize, разработанном Цзугуанем Гу (Zuguang Gu), реализованы различные круговые графики, включая хордовые диаграммы. Пакет имеет множество функций, предоставляя пользователю максимальные возможности по настройке внешнего вида графика.

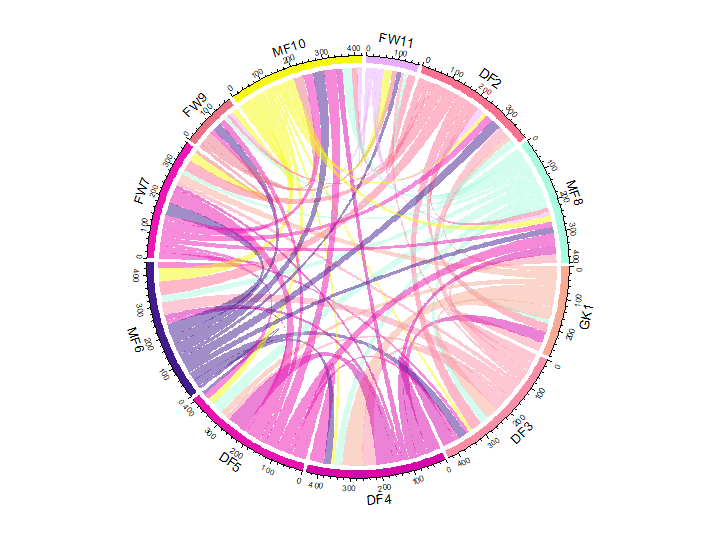


Рис. 22. Хордовая диаграмма для сети игроков сборной Нидерландов по футболу 2010 года, параметры по умолчанию

**Теплокарты** – еще один пример специализированного графика, который можно использовать для визуализации сети, особенно когда речь идет о сети с числовыми значениями (т. е. взвешенной сети). В данном случае теплокарта строится, чтобы выделить игроков, которые передают или получают наибольшее количество пасов в команде.

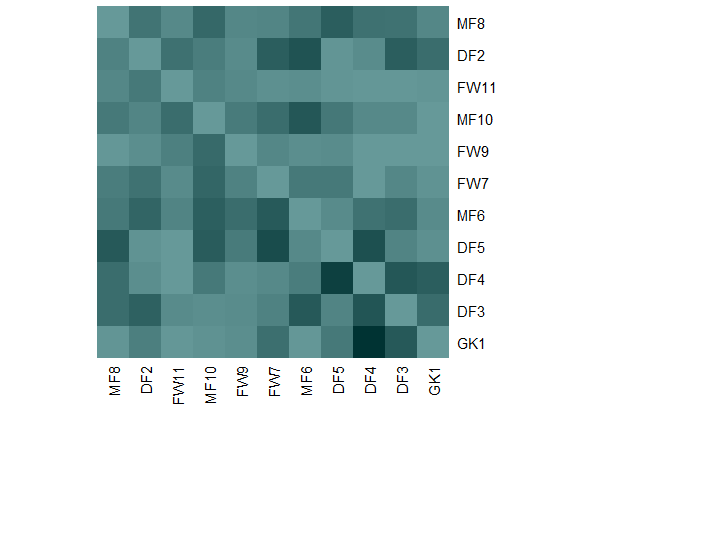


Рис. 23 Теплокарта для сети игроков сборной Нидерландов по футболу 2010 года, интенсивность цвета соответствует количеству пасов

# Выводы по работе

В среде R существует множество настроек внешнего вида сетей, которые помогают упростить анализ этих сетей. При правильной настройке визуальные данные будут помогать определять параметры сети, при использовании интерактивных графиков можно даже экспортировать их в веб.

Рассмотренная книга помогает разобраться во множестве библиотек и функций для визуализации сетей в среде R, делая это в том числе на наглядных примерах.