

*Посвящаю эту книгу
своей жене Светлане*

Оглавление

Предисловие	7
1 Введение	10
1.1 Что представляет собой <code>ggplot2</code> ?	10
1.2 Установка <code>ggplot2</code> и начало работы	10
1.3 Грамматика графических элементов	11
1.4 Данные, используемые в примерах	12
2 Функция <code>qplot()</code>: быстрое решение для задач визуализации	15
2.1 Аргументы функции <code>qplot()</code>	15
2.2 Построение диаграмм рассеяния с помощью <code>qplot()</code>	16
2.3 Другие примеры использования <code>qplot()</code>	19
2.3.1 Линии тренда	20
2.3.2 Одномерные диаграммы рассеяния	21
2.3.3 Диаграммы размахов	22
2.3.4 Гистограммы, кривые плотности вероятности, полигоны частот	25
2.3.5 Столбиковые диаграммы	30
2.4 Категоризованные графики	31
3 Построение графиков слой за слоем	34
3.1 Аргументы функции <code>ggplot()</code>	34
3.2 Слои	35
3.3 Требования к данным	38
3.4 Присваивание эстетических атрибутов	39
3.5 Группирование данных	41
3.6 Геометрические объекты, реализованные в <code>ggplot2</code>	43
3.7 Статистические преобразования	46
4 Основные типы статистических графиков	49
4.1 Общие аргументы <code>geom</code> - и <code>stat</code> -функций	49
4.2 Визуализация одномерных распределений	50
4.2.1 Точечные диаграммы Уилкинсона: <code>geom_dotplot()</code>	50
4.2.2 Столбиковые диаграммы: <code>geom_bar()</code>	54
4.2.3 Гистограммы: <code>geom_histogram()</code>	60
4.2.4 Полигоны частот: <code>geom_freqpoly()</code>	62
4.2.5 Кривые плотности вероятности: <code>geom_density()</code>	64

4.2.6	Кумулятивные функции распределения: <code>geom_step()</code>	67
4.2.7	Квантильные графики: <code>stat_qq()</code>	71
4.3	Визуализация 2D- и 3D-распределений	73
4.3.1	Контуры плотности вероятности: <code>geom_density2d()</code>	74
4.3.2	Изолинии: <code>geom_contour()</code>	76
4.3.3	Сотовые диаграммы: <code>geom_hex()</code>	78
4.4	Визуализация сводной статистической информации о количественных переменных	79
4.4.1	Диаграммы диапазонов: <code>geom_linerange()</code> , <code>geom_pointrange()</code> , <code>geom_errorbar()</code> , <code>geom_crossbar()</code>	79
4.4.2	Диаграммы размахов: <code>geom_boxplot()</code>	85
4.4.3	Скрипичные диаграммы: <code>geom_violin()</code>	89
4.5	Визуализация зависимостей	92
4.5.1	Диаграммы рассеяния: <code>geom_point()</code>	93
4.5.2	Линии тренда: <code>geom_smooth()</code>	95
4.5.3	Линии квантильной регрессии: <code>geom_quantile()</code>	99
4.6	Визуализация временных рядов	101
4.6.1	Функция <code>geom_line()</code>	102
4.6.2	Функция <code>geom_ribbon()</code>	104
4.7	Тепловые карты: <code>geom_tile()</code>	105
4.8	Другие геометрические объекты	107
4.8.1	«График-щетка»: <code>geom_rug()</code>	107
4.8.2	Горизонтальные и вертикальные линии: <code>geom_hline()</code> , <code>geom_vline()</code>	109
4.8.3	Прямоугольные области: <code>geom_rect()</code>	111
4.8.4	Отрезки: <code>geom_segment()</code>	112
4.8.5	Ломаные линии: <code>geom_path()</code>	113
4.8.6	Многоугольники: <code>geom_polygon()</code>	116
4.8.7	Площадь под кривой: <code>geom_area()</code>	118
4.8.8	Текстовые аннотации: <code>geom_text()</code>	119
4.9	Географические карты: <code>geom_map()</code>	124
4.10	Добавление слоев при помощи функций семейства <code>stat</code>	131
5	Шкалы	134
5.1	Шкалы и их основные типы	134
5.2	Аргументы, общие для всех <code>scale</code> -функций	137
5.3	Шкалы положения	140
5.3.1	Шкалы положения для количественных переменных	140
5.3.2	Шкалы положения для дат и времени	145
5.3.3	Шкалы положения для качественных переменных	147
5.4	Цветовые шкалы	149
5.4.1	Цветовые шкалы для количественных переменных	150
5.4.2	Цветовые шкалы для качественных переменных	154
5.5	Пользовательские шкалы для качественных переменных	158
5.6	Тождественные шкалы	160

6	Системы координат	162
6.1	Декартова система и ее разновидности	163
6.2	Полярная система	166
6.3	Картографические проекции	168
7	Подробнее о категоризованных графиках	171
7.1	Два способа организации панелей	171
7.2	Функция <code>facet_grid()</code>	172
7.3	Функция <code>facet_wrap()</code>	180
8	Подготовка графиков к публикации	183
8.1	Стили	183
8.2	Создание составных рисунков	201
8.2.1	Использование окон просмотра	203
8.2.2	Использование пакета <code>gridExtra</code>	206
8.3	Экспорт графиков из среды R	208
9	Дополнительные ресурсы для изучения ggplot2	212
9.1	Литература	212
9.2	Онлайн-ресурсы	213
9.3	Расширения, созданные на основе <code>ggplot2</code>	214
	Предметный указатель	217

Предисловие

*«...нет статистического метода более мощного,
чем хорошо подобранный график.»*
(Chambers et al., 1983¹)

Визуализация данных играет важную роль на всех этапах статистического анализа — от первичного ознакомления со свойствами данных до диагностики качества построенных моделей и представления полученных результатов. Существует много компьютерных программ для выполнения сложных статистических расчетов и создания не менее сложных графиков. Из всего этого разнообразия выделяется R — интенсивно развивающаяся и свободно распространяемая система статистических вычислений, в которой реализовано множество классических и современных методов анализа данных. Язык R имеет почти полувековую историю. Он был создан в середине 1990-х г. в Университете Окленда (Unverity of Auckland) Робертом Джентельменом (Robet Gentleman) и Росом Ихаккой (Ross Ihaka) на основе языка S², который, в свою очередь, был разработан в AT&T Bell Laboratories Джоном Чемберсом (John Chambers), Риком Бекером (Rick Becker) и их коллегами в 1976 году³.

Программные реализации алгоритмов, входящих в ядро R, проверены на практике не одним поколением пользователей и ученых. Кроме того, пользователи R постоянно разрабатывают многочисленные дополнения (т.н. «пакеты») для этой системы⁴. Представляемая вашему вниманию книга посвящена `ggplot2` — одному из таких пакетов, который значительно расширяет и без того богатые базовые возможности R по визуализации данных. Основные создатели `ggplot2` — Хэдли Уикхэм⁵ (Hadley Wickham)

¹ Chambers J. M., Cleveland W. S., Tukey P. A., Kleiner B. (1983) Graphical Methods for Data Analysis. Duxbury Press.

² <http://cran.r-project.org>.

³ Подробнее о создании языка S можно узнать из очень интересного выступления Рика Бекера «Forty Years of S» (конференция UseR, Стэнфорд, 2016 г.): <http://bit.ly/291KfNm>.

⁴ В сентябре 2016 г. количество пакетов в хранилище CRAN превысило 9000. Актуальную статистику можно всегда узнать на странице <http://bit.ly/2d6ugbn>.

⁵ <http://had.co.nz>.

и Уинстон Ченг⁶ (Winston Chang) — проделали огромную работу, результатами которой сегодня пользуются сотни тысяч людей⁷. Популярность пакета обусловлена несколькими причинами, среди которых можно отметить эстетическую привлекательность и пригодное для публикации качество получаемых с его помощью графиков, возможность создавать пользовательские типы диаграмм, а также широкий набор инструментов для настройки внешнего вида графиков. Кроме того, логика и синтаксис команд `ggplot2` базируются на интуитивно понятных идеях «грамматики графических элементов» (Wilkinson, 1999⁸), что облегчает программирование.

К сожалению, информации о `ggplot2` на русском языке крайне мало. Цель данной книги — заполнить этот информационный пробел, представив описание основных возможностей пакета. Следует, однако, подчеркнуть, что у меня не было намерения дать сколь-либо исчерпывающее описание `ggplot2`: такими полными источниками всегда будут книги Х. Уикхэма (Wickham, 2009, 2016⁹). Кроме того, важным справочным источником является официальная онлайн-документация по `ggplot2`¹⁰, которую нет смысла дублировать на бумаге.

Настоящая книга предназначена для широкой аудитории — для всех, кто сталкивается с необходимостью визуализации данных и интересуется соответствующими методами и средствами. Предполагается, что читатель имеет некоторое представление об основных статистических понятиях (на уровне вводного университетского курса статистики) и обладает уверенными навыками работы с R. Последнее обстоятельство особенно важно, поскольку команды, не имеющие непосредственного отношения к `ggplot2`, подробно здесь не обсуждаются. Читателям, не знакомым с R, можно порекомендовать следующие книги на русском языке:

- Зарядов И. С. (2010) Введение в статистический пакет R: типы переменных, структуры данных, чтение и запись информации, графика. Российский университет дружбы народов.
- Шипунов А. Б., Балдин Е. М., Волкова П. А., Коробейников А. И., Назаров С. А., Петров С. В., Суфиянов В. Г. (2012) Наглядная статистика. Используем R! ДМК Пресс.
- Мاستицкий С. Э., Шитиков В. К. (2015) Статистический анализ и визуализация данных с помощью R. ДМК Пресс.

Я благодарен Дмитрию Мовчану и всей команде «ДМК Пресс» за помощь с подготовкой и изданием этой книги, а также Петру Валь, Алексею Кожевину, Игорю Магдееву, Александру Маслову, Виталию Скальскому и Тимуру Шеферу за внимательное прочтение рукописи и высказанные ими предложения по улучшению текста.

⁶ <https://github.com/wch>.

⁷ В 2015 г. пакет был установлен более миллиона раз (Wickham, 2016).

⁸ Wilkinson L. (1999) *The Grammar of Graphics*. Springer.

⁹ Wickham H. (2009) *ggplot2: Elegant graphics for data analysis*. Springer. Второе издание книги вышло в июне 2016 г.

¹⁰ <http://docs.ggplot2.org/current>.

Эту книгу следует считать приложением к моему блогу «R: Анализ и визуализация данных»¹¹, целью которого является популяризация R среди русскоязычных пользователей. Все примеры кода и данные можно найти на GitHub-странице книги¹². Любые замечания и пожелания вы можете направлять по электронной почте `rtutorialsbook@gmail.com`.

*Сергей Мاستицкий,
Лондон, сентябрь 2016 г.*

¹¹ <http://r-analytics.blogspot.com>.

¹² <https://github.com/ranalytics/ggplot2-ru>.

Глава 1

Введение

1.1 Что представляет собой ggplot2?

Пакет является дополнением к системе статистических вычислений R и служит для визуализации данных. В основе `ggplot2` лежат идеи «грамматики графических элементов» Лиланда Уилкинсона (Leland Wilkinson), что проявляется в концептуальной целостности и логичном синтаксисе этого пакета. В своей работе Л. Уилкинсон (Wilkinson, 1999¹) описал набор элементарных графических компонентов, комбинируя которые, можно послойно создавать самые замысловатые графики. Все эти компоненты и операции для их комбинирования доступны в `ggplot2`, благодаря чему пользователь практически не ограничен в выборе заранее определенных типов графиков и имеет возможность изображать данные в точном соответствии со своими потребностями. При этом тщательно продуманные и заданные по умолчанию настройки `ggplot2` позволяют создавать эстетически привлекательные графики с использованием лаконичного синтаксиса.

1.2 Инсталляция ggplot2 и начало работы

Инсталляция пакета `ggplot2` не составляет никакого труда. Убедитесь, что на вашем компьютере установлена последняя версия R² и что он подключен к сети Интернет, после чего выполните следующую обычную в таких случаях команду:

```
install.packages("ggplot2")
```

Перед использованием установленного пакета `ggplot2` необходимо загрузить его при помощи команды

```
library("ggplot2")
```

¹ Wilkinson L. (1999) The Grammar of Graphics. Springer.

² Приведенные в книге примеры были созданы с использованием R v3.3.1 и `ggplot2` v2.0.0.

1.3 Грамматика графических элементов

Согласно Л. Уилкинсону (Wilkinson, 1999), статистический график — это результат преобразования исходных данных в *геометрические объекты* (например, точки, линии, столбцы), обладающие определенными *эстетическими атрибутами* (цвет, форма, размер). График строится с в рамках некоторой *системы координат* и дополнительно может изображать результаты *статистических преобразований* исходных данных (например, средние значения, доверительные интервалы, сглаживающие кривые и т. п.). Для одновременной визуализации сгруппированных данных можно разбить область рисунка на отдельные ячейки («панели») и разместить в них графики, соответствующие каждой группе. Комбинирование перечисленных компонентов и дает возможность создавать разнотипные статистические графики.

Ниже приведены основные термины «грамматики графических элементов», понимание которых важно для освоения пакета `ggplot2`:

- **data** — подлежащие визуализации *данные*;
- **mapping** — процедура присваивания *координат, формы, размера и цвета* изображаемому на графике объектам в соответствии со значениями анализируемых переменных;
- **geom** (сокращение от «*geometric object*») — «*геометрические объекты*», используемые для изображения данных (точки, линии, многоугольники, и т. п.). Между типом этих объектов и типом графика существует тесная связь. Так, на диаграммах рассеяния для изображения данных обычно используют точки, для построения гистограмм — прямоугольные столбики, а для изображения временных рядов применяют линии. В состав пакета `ggplot2` входят более 35 типов геометрических объектов³, которые можно комбинировать в любых сочетаниях;
- **stat** (сокращение от «*statistical transformations*») — «*статистические преобразования*», применяемые к данным для обобщения заключенной в них информации. Объединение значений количественных переменных в дискретные классы для построения гистограмм и представление связи между двумя переменными в виде линии регрессии служат примерами таких преобразований. Статистические преобразования не являются обязательными элементами графиков, однако во многих случаях они оказываются очень полезными;
- **scale** (дословно «*шкала*») — функция, выполняющая отображение пространства данных на пространство эстетических атрибутов. Результатом работы таких функций является преобразование данных в то, что мы можем воспринять визуально, — координаты, форма, размер, цвет, тип линии и т. д.;
- **coord** — *система координат*, в которой строится график. Обычно используется декартова система координат, однако в пакете `ggplot2`

³ Этот список (доступен по команде `??ggplot2::geom`) растет с каждой новой версией пакета.

реализованы и другие системы (например, полярная система координат и разнообразные картографические проекции);

- **facet** — сокращение от «*faceting*», что означает *разбиение данных на группы* и изображение графиков для каждой из этих групп на одном рисунке. В русскоязычной литературе по статистике такие графики часто называют «*категоризованными*». В англоязычной литературе используются также термины «*lattice plots*» и «*trellis plots*».

1.4 Данные, используемые в примерах

Примеры в книгах и статьях по R часто строятся на одних и тех же наборах данных (*iris*, *mtcars* и т. п.). Я решил изменить этой старой доброй традиции и привнести некоторое разнообразие, заодно рассказав читателям о том, чего они раньше, возможно, не знали.

Помните, как в школе на уроках биологии вы изучали инфузорию-туфельку? Оказывается, в природе существует и множество других видов инфузорий. При этом многие из них, в отличие от инфузории-туфельки, живут не просто в воде луж, прудов и озер, а населяют внутренние полости тела других водных животных. Один из таких видов — это *конхофтирус остроконечный* (научное название *Conchophthirus acuminatus*). Конхофтирус обитает в полостях тела *дрейссены речной* (научное название *Dreissena polymorpha*) — моллюска, который широко распространен в пресноводных водоемах Европы и Северной Америки (рис. 1.1)⁴. Никакого вреда эти инфузории не оказывают — они просто подхватывают и с удовольствием поедают остатки пищи (водоросли, бактерии, органические частички и т. п.), отфильтрованной хозяином из толщи воды⁵.

Большинство используемых в книге примеров основано на моих собственных данных по количеству инфузорий *C. acuminatus*, обнаруженных в дрейссене из трех озер — Нарочь, Мясстро и Баторино (Республика Беларусь). Подробнее об этом исследовании можно узнать в статье Mastitsky (2012)⁶. Полученные данные опубликованы на сайте сервиса «figshare»⁷, откуда их можно загрузить непосредственно в R при помощи команды

```
dreissena <- read.delim(
  "http://files.figshare.com/1360878/Dreissena.txt")
```

В состав таблицы *dreissena* входят следующие переменные:

- **Month** — качественная переменная с тремя уровнями, соответствующими времени отбора проб дрейссены: *May* (май), *July* (июль) и *September* (сентябрь);
- **Day** — день отбора проб (с даты начала проведения исследований);

⁴ Подробнее см. статью в Википедии: <http://bit.ly/2cdwukq>.

⁵ Такие взаимоотношения между двумя видами называются *комменсализмом*.

⁶ Mastitsky S. E. (2012) Infection of *Dreissena polymorpha* (Bivalvia: Dreissenidae) with *Conchophthirus acuminatus* (Ciliophora: Conchophthiridae) in lakes of different trophy. *BioInvasions Records* 1(3): 161–169.

⁷ Mastitsky S. (2012) Infection of the zebra mussel with its commensal ciliate *Conchophthirus acuminatus*. *figshare*:<http://dx.doi.org/10.6084/m9.figshare.95449>.



Рисунок 1.1. Дрейссена речная (*Dreissena polymorpha*) — пресноводный моллюск родом из низовьев рек, впадающих в Черное и Азовское моря. Один из наиболее активно расселяющихся чужеродных видов в Европе и Северной Америке. На фотографии видно несколько десятков моллюсков, почти полностью облепивших небольшой булыжник

- **Lake** — качественная переменная с тремя уровнями, обозначающими изученные озера: Batorino, Myastro и Naroch;
- **Site** — качественная переменная с девятью уровнями, обозначающими места отбора проб (S1 – S9). В каждом озере моллюсков собирали на трех постоянных станциях;
- **Length** — длина раковины моллюсков (мм);
- **Infection** — количество инфузорий, обнаруженных в каждом моллюске (далее по тексту будут использоваться также термины «интенсивность инвазии» и «уровень инвазии»).

Со структурой этих данных можно ознакомиться при помощи стандартной команды `str()`:

```
str(dreissena)
```

```
'data.frame':      476 obs. of  6 variables:
 $ Month   : Factor w/ 3 levels "July","May","September": 2 ...
 $ Day     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Lake    : Factor w/ 3 levels "Batorino","Myastro",...: 1 1 ...
 $ Site    : Factor w/ 9 levels "S1","S2","S3",...: 3 3 3 3 ...
 $ Length  : num  14.9 14 13 14 12 14 12 19 16.5 18 ...
 $ Infection: int  36 30 331 110 4 171 31 887 525 497 ...
```

Набор данных `dreissena` очень удобен для иллюстрации возможностей `ggplot2`, поскольку обладает небольшим размером (476 наблюдений) и содержит 5 разнотипных переменных (количественные и качественные), чьи

значения изменяются во времени. Для правильного отображения хронологической последовательности месяцев на графиках, которые мы будем строить в дальнейшем, необходимо сообщить R о том, что `Month` является качественной переменной с упорядоченными (англ. *ordered*) уровнями. Для этого следует выполнить следующую команду:

```
dreissena$Month <- factor(dreissena$Month, ordered = TRUE,  
                          levels = c("May", "July", "September"))
```

Хотя обычно R без труда «понимает» кириллические текстовые выражения, определенные настройки операционной системы компьютера могут сопровождаться некорректным распознаванием таких выражений. Поэтому почти на всех приведенных в книге графиках имена переменных из таблицы `dreissena`, а также значения качественных переменных из этой таблицы на русский язык не переводятся (например, `Lake`, а не `Озеро`, `May`, а не `май`, и т. д.). Выбор в пользу оригинальных имен был сделан осознанно, чтобы исключить обусловленные кодировкой проблемы при воспроизведении примеров.

Другие использованные в книге наборы данных будут описаны непосредственно в ходе рассмотрения соответствующих примеров.

Глава 2

Функция `qplot()`: быстрое решение для задач визуализации

Функция `qplot()` получила свое название от двух слов — *quick* и *plot*, что значит «быстрый» и «график» соответственно. Название этой функции полностью соответствует ее назначению — она позволяет строить самые разнообразные статистические графики с использованием одной–двух строк кода. Если вы уже знакомы с `plot()` — базовой графической функцией R, то освоение `qplot()` не составит никакого труда.

2.1 Аргументы функции `qplot()`

Функция `qplot()` имеет следующие основные аргументы:

- `x` и `y` — переменные X и Y соответственно;
- `data` — таблица данных («*data frame*» в терминах R), содержащая переменные X и Y . Если этот аргумент не указан, то функция `qplot()` попытается автоматически извлечь векторы `x` и `y` из текущей рабочей среды и объединить их в таблицу;
- `facets` — формула, определяющая способ разбиения рисунка на отдельные подобласти при создании категоризованных графиков (см. разд. 2.4 и главу 7);
- `margins` — аргумент, используемый при создании категоризованных графиков. Позволяет включать (`TRUE`) или отключать (`FALSE`) отображаемые по краям графика названия уровней качественной переменной, в соответствии с которыми рисунок разбивается на подобласти;
- `geom` — текстовый вектор с названиям геометрических объектов, используемых для изображения данных. Если на функцию `qplot()`

поданы две переменные — X и Y , то аргумент `geom` по умолчанию примет значение `"point"` («точка»). Если же подана только количественная переменная Y , то значением по умолчанию будет `"histogram"` («гистограмма»). Возможно совмещение нескольких типов геометрических объектов на одном рисунке;

- `stat` — текстовый вектор, определяющий тип статистического преобразования данных;
- `xlim` и `ylim` — задают границы значений переменных X и Y соответственно (в виде `c(нижняя граница, верхняя граница)`);
- `log` — позволяет логарифмически «растянуть» ось X (`log = "x"`), ось Y (`log = "y"`), или обе оси одновременно (`log = "xy"`);
- `main` — текстовый вектор и (или) математическое выражение, образующие заголовок графика;
- `xlab` и `ylab` — текстовые векторы и (или) математические выражения, образующие подписи осей X и Y соответственно;
- `asp` — число, задающее отношение длины X к длине оси Y .

2.2 Построение диаграмм рассеяния с помощью `qplot()`

Первые два аргумента `qplot()` — `x` и `y` — задают переменные, значения которых будут отложены по соответствующим координатным осям создаваемого графика. Если `x` и `y` представляют собой самостоятельные векторы, то функция `qplot()` попытается автоматически объединить их в одну таблицу. Такой подход не защищен от возникновения непредвиденных ошибок, в связи с чем рекомендуется всегда предварительно объединять необходимые данные в одну таблицу и далее ссылаться на нее при помощи аргумента `data`. Ниже приведен пример обычной *диаграммы рассеяния*, построенной с помощью функции `qplot()` по данным из таблицы `dreissena` (рис. 2.1).

— Код для рис. 2.1 —

```
qplot(x = Length, y = Infection, data = dreissena)
```

На рис. 2.1 видно, что количество инфузорий *C. acuminatus* положительно и нелинейно связано с длиной раковины моллюсков *D. polymorpha*. Мы можем «выровнять» эту зависимость путем логарифмирования обеих переменных (рис. 2.2).

— Код для рис. 2.2 —

```
qplot(x = log(Length),
      y = log(Infection + 1), data = dreissena)
# поскольку некоторые значения Infection равны 0,
# логарифмирование выполнено для (Infection + 1)
```

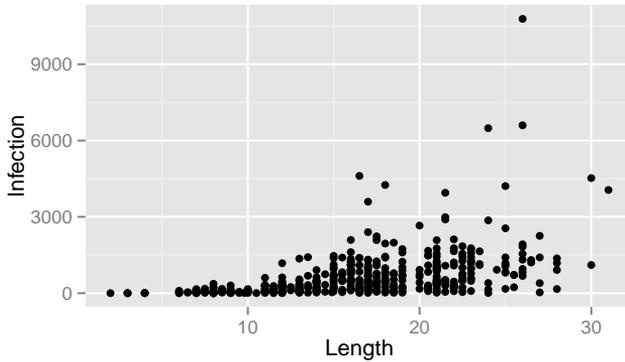


Рисунок 2.1. Пример диаграммы рассеяния, построенной с помощью функции `qplot()`. Изображена связь между длиной раковины дрейссены и интенсивностью инвазии *C. acuminatus*

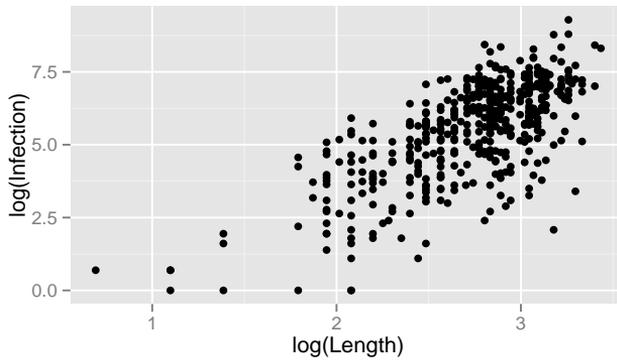


Рисунок 2.2. То же, что и на рис. 2.1, но после логарифмирования исходных данных

Одно из существенных отличий функции `qplot()` от базовой R-функции `plot()` состоит в том, каким образом точкам на графике присваиваются эстетические атрибуты, т. е. цвет, размер и форма. В случае с `plot()` пользователь должен самостоятельно конвертировать уровни интересующей его качественной переменной (например, «зима», «весна», «лето», «осень») в соответствующие значения эстетических атрибутов (например, цвет для разных сезонов года: «белый», «голубой», «зеленый», «оранжевый»). Функция же `qplot()` выполняет такие преобразования автоматически, одновременно создавая легенду с цветовой шкалой, которую пользователь может изменить в соответствии со своими требованиями.

На рис. 2.3 показано, как к графику зависимости между двумя количественными переменными можно добавить информацию о третьей — качественной — переменной, изменяя цвет точек (аргумент `colour`) или их форму (аргумент `shape`). Автоматическое присваивание значений эс-

тетических атрибутов можно отменить, воспользовавшись стандартной R-функцией `I()`¹ (например, «вручную» задав значения `colour = I("red")` или `shape = I(2)`).

Код для рис. 2.3

```
qplot(log(Length), log(Infection + 1), data = dreissena,
      colour = Month)
qplot(log(Length), log(Infection + 1), data = dreissena,
      shape = Lake)
```

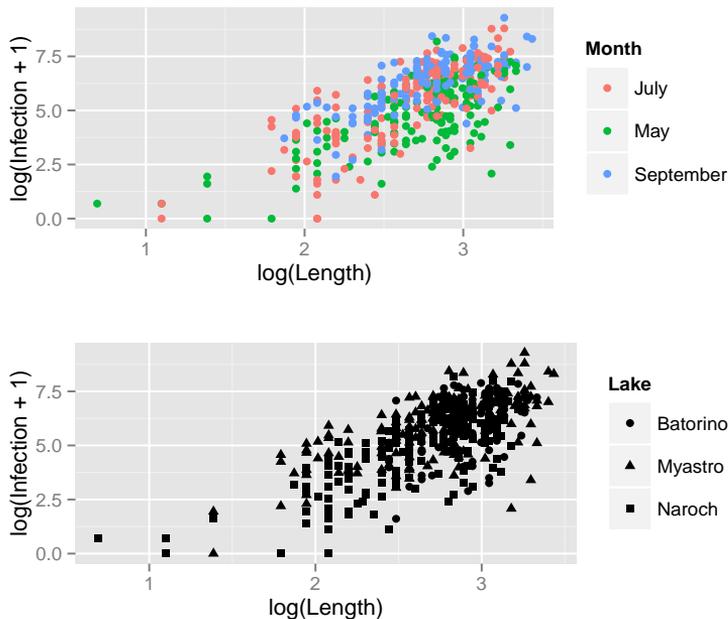


Рисунок 2.3. Примеры присвоения точкам атрибутов «цвет» (верхний график, в соответствии с датой отбора проб) и «форма» (нижний график, в соответствии с водоемом, из которого были отобраны пробы)

Обычно при работе с данными большого объема точки на диаграммах рассеяния накладываются друг на друга, что затрудняет выявление заключенных в данных закономерностей (см., например, рис. 2.1). Полезным приемом для облегчения восприятия таких графиков является использование полупрозрачного цвета (рис. 2.4). Этот прием можно реализовать при помощи аргумента `alpha`, который принимает значения от 0 (полная прозрачность) до 1 (полная непрозрачность). Одним из решений для визуализации данных большого объема является также использование *категоризованных графиков* (см. разд. 2.4 и главу 7).

¹ Функция `I()` подавляет любые преобразования объекта R, возвращая его с сохранением исходного класса.

Код для рис. 2.4

```
qplot(Length, Infection, alpha = I(1/2), data = dreissena)
qplot(Length, Infection, alpha = I(1/4), data = dreissena)
qplot(Length, Infection, alpha = I(1/8), data = dreissena)
```

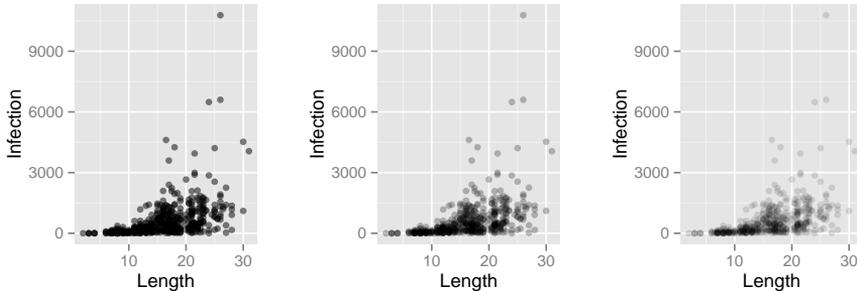


Рисунок 2.4. Примеры использования полупрозрачных точек на диаграммах рассеяния с большим количеством наблюдений. На приведенных графиках (слева направо) значения аргумента `alpha` составляют 0.5, 0.25 и 0.125

Следует помнить, что разные эстетические атрибуты неодинаково хорошо подходят для работы с качественными и количественными переменными. Так, цвет и форма хорошо разграничивают уровни качественных переменных, тогда как атрибут «размер» лучше работает с количественными переменными. Размер точек (и других графических объектов) в `qplot()` задается при помощи аргумента `size` (рис. 2.5).

Код для рис. 2.5

```
qplot(log(Length), log(Infection + 1), data = dreissena,
      size = Day, alpha = I(0.25), colour = I("magenta"))
```

2.3 Другие примеры использования `qplot()`

Безусловно, с помощью функции `qplot()` можно создавать не только диаграммы рассеяния: варьируя значения аргумента `geom`, пользователь получает возможность построить практически все распространенные типы статистических графиков. Аргумент `geom` определяет тип геометрических объектов, используемых для изображения данных. Так, к наиболее часто используемым значениям `geom` относятся следующие:

- `geom = "point"` — изображает данные в виде точек (см., например, рис. 2.1);

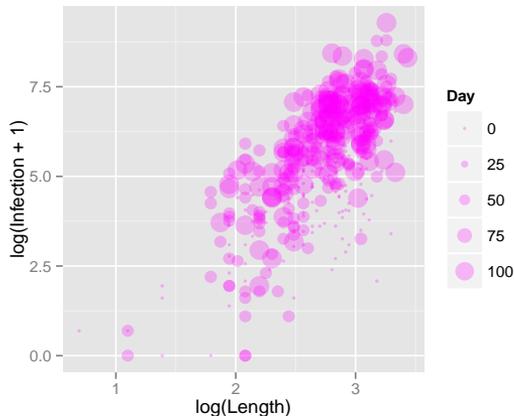


Рисунок 2.5. Диаграмма рассеяния, на которой точкам присвоен атрибут «размер» (`size`) в соответствии со значениями третьей (количественной) переменной

- `geom = "smooth"` — подгоняет сглаживающую кривую к данным и одновременно изображает ее 95%-ную доверительную область;
- `geom = "jitter"` — создает одномерные диаграммы рассеяния;
- `geom = "boxplot"` — создает диаграммы размахов;
- `geom = "path"` и `geom = "line"` — соединяют точки линиями. Традиционно используются для изображения временных изменений количественных переменных (`geom = "line"`). Однако точки могут соединяться не только в соответствии с ходом времени, т. е. слева направо, но и любым другим образом (`geom = "path"`).

При анализе свойств только одной переменной выбор возможных значений аргумента `geom` будет определяться типом этой переменной:

- количественные переменные: значение `geom = "histogram"` приведет к созданию гистограммы, `geom = "freqpoly"` — полигона распределения частот, а `geom = "density"` — кривой плотности вероятности;
- качественные переменные: значение `geom = "bar"` приведет к созданию столбиковой диаграммы.

2.3.1 Линии тренда

Как было отмечено ранее, на диаграммах рассеяния с большим количеством наблюдений бывает сложно увидеть какие-либо четкие закономерности. Помимо использования полупрозрачного цвета точек (см. рис. 2.4 и 2.5), полезным приемом в таких случаях может оказаться также добавление к графику *сглаживающей линии* (англ. *smoother*), или *линии*

тренда. В `ggplot2` для этого служит геометрический объект типа `smooth`. Обратите внимание на то, как в приведенном ниже коде для рис. 2.6 два типа геометрических объектов — точки и сглаживающая линия — были совмещены стандартным для R образом, т. е. с помощью функции конкатенации `c()`. Слои (см. разд. 3.2) с соответствующими геометрическими объектами появятся на графике в порядке перечисления этих объектов в скобках команды `c()`.

Код для рис. 2.6

```
qplot(log(Length), log(Infection + 1),
      geom = c("point", "smooth"), data = dreissena)
```

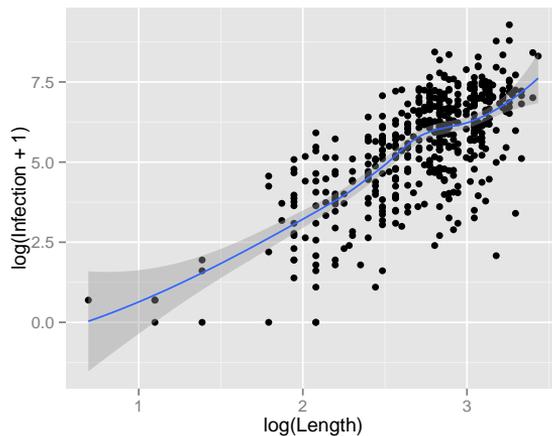


Рисунок 2.6. Пример добавления сглаживающей линии к диаграмме рассеяния

2.3.2 Одномерные диаграммы рассеяния

Одномерная диаграмма рассеяния (англ. *strip chart* или *strip plot*) является подходящим инструментом для визуализации значений какой-либо количественной переменной в соответствии с уровнями качественной переменной. Для создания таких диаграмм в `ggplot2` служит геометрический объект типа `"jitter"`. Во избежание излишнего перекрытия точек на графике к их X -координатам случайным образом добавляется небольшой «шум» (рис. 2.7). Поскольку при этом используется встроенный в R генератор псевдослучайных чисел, каждый раз при выполнении соответствующего кода внешний вид рисунка будет несколько изменяться².

² Для воспроизведения внешнего вида графика при повторном исполнении кода следует воспользоваться функцией `set.seed()`.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru