



# Содержание

От издательства.....	20
Об авторе.....	21
О техническом редакторе.....	22
Благодарности .....	23
Введение.....	24
<b>Часть I. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ ВИЗУАЛИЗАЦИИ ПРИ ПОМОЩИ R</b> .....	<b>41</b>
<b>Глава 1. Грамматика графиков</b> .....	<b>42</b>
Пошаговое создание визуализации в Power BI при помощи R.....	43
Шаг 1. Настройте Power BI .....	43
Шаг 2. Перенесите визуальный элемент R в рабочую область Power BI.....	43
Шаг 3. Определитесь с набором данных.....	43
Шаг 4. Спроектируйте визуальный элемент в среде разработки R.....	44
Шаг 5. Используйте следующий шаблон для разработки элемента на R .....	45
Шаг 6. Добавьте скрипту функциональности .....	45
Рекомендованные шаги по созданию визуального элемента на R при помощи ggplot2.....	46
Шаг 1. Импортируйте нужные для скрипта пакеты .....	46
Шаг 2. Выполните необходимое преобразование исходных данных .....	47
Шаг 3. Создайте визуализацию при помощи функции ggplot() .....	48
Шаг 4. Добавьте нужные геометрии.....	48
Шаг 5. Определите заголовки, подзаголовки и подписи .....	50
Шаг 6. Приведите в порядок оси.....	50
Шаг 7. Примените тему при необходимости .....	53
Шаг 8. Используйте функцию theme() для настройки оформления .....	54
Дополнительный шаг: задайте цвета точек на диаграмме рассеяния .....	55
Важность оперирования «чистыми» данными.....	58
Популярные геометрии.....	58
Управление эстетиками через шкалы .....	64
Встроенные в пакет ggplot2 темы.....	65
Использование визуальных элементов R в службе Power BI.....	66
Вспомогательные пакеты ggplot2.....	66
Заключение .....	67

<b>Глава 2. Создание пользовательских визуализаций на R в Power BI при помощи ggplot2</b> .....	68
<b>Диаграмма с аннотацией</b> .....	69
Шаг 1. Загрузите исходные данные.....	70
Шаг 2. Создайте срез на основании года на панели фильтров.....	71
Шаг 3. Настройте визуальный элемент R в Power BI.....	71
Шаг 4. Экспортируйте данные в R Studio для дальнейшей разработки.....	72
Шаг 5. Загрузите необходимые пакеты.....	73
Шаг 6. Создайте переменные для проверки данных.....	73
Шаг 7. Выполните проверку данных.....	74
Шаг 8. Добавьте столбцы к набору данных, необходимые для нашей визуализации.....	75
Шаг 9. Создайте переменные для динамических составляющих диаграммы.....	76
Шаг 10. Постройте диаграмму при помощи функции ggplot().....	78
Шаг 11. Добавьте слой со столбчатой диаграммой на визуальный элемент.....	78
Шаг 12. Добавьте текстовый слой на визуальный элемент.....	79
Шаг 13. Измените ось y.....	80
Шаг 14. Преобразуйте вертикальную столбчатую диаграмму в горизонтальную.....	81
Шаг 15. Добавьте на диаграмму динамическую аннотацию.....	81
Шаг 16. Добавьте динамические заголовки и подпись на визуальный элемент.....	83
Шаг 17. Удалите метки с осей x и y.....	84
Шаг 18. Удалите легенду с диаграммы.....	84
Шаг 19. Измените внешний вид диаграммы при помощи темы theme_few().....	85
Шаг 20. Расположите заголовки по центру.....	86
Шаг 21. Перенесите код в Power BI.....	87
<b>Пузырьковая диаграмма</b> .....	89
Шаг 1. Загрузите исходные данные.....	90
Шаг 2. Загрузите данные в Power BI.....	90
Шаг 3. Создайте срез на основании года.....	90
Шаг 4. Выполните базовую настройку визуального элемента R.....	91
Шаг 5. Экспортируйте данные в R Studio для разработки элемента.....	91
Шаг 6. Загрузите требуемые пакеты.....	91
Шаг 7. Создайте переменные для проверки данных.....	91
Шаг 8. Создайте код для проверки.....	92
Шаг 9. Определите цвета для конференций и дивизионов.....	92
Шаг 10. Динамически определите заголовок диаграммы.....	93
Шаг 11. Создайте набор данных для диаграммы.....	93
Шаг 12. Создайте диаграмму при помощи функции ggplot.....	94
Шаг 13. Добавьте слой для пузырьковой диаграммы при помощи геометрии geom_point.....	94
Шаг 14. Добавьте метки на диаграмму.....	96

Шаг 15. Измените цвет границ и заливок пузырьков на диаграмме .....	97
Шаг 16. Создайте заголовок диаграммы .....	98
Шаг 17. Задайте тему.....	98
Шаг 18. Перенесите код в Power BI .....	98
Визуализация прогнозирования .....	99
Шаг 1. Загрузите исходные данные .....	101
Шаг 2. Создайте срез по квотербекам на панели фильтров .....	102
Шаг 3. Настройте визуальный элемент R в Power BI.....	102
Шаг 4. Экспортируйте данные в R Studio для дальнейшей разработки.....	102
Шаг 5. Загрузите необходимые пакеты .....	102
Шаг 6. Создайте переменные для проверки данных.....	103
Шаг 7. Выполните проверку данных .....	103
Шаг 8. Создайте динамический заголовок для визуализации.....	104
Шаг 9. Создайте набор данных, необходимый для составления прогноза .....	104
Шаг 10. Постройте прогноз.....	105
Шаг 11. Постройте диаграмму.....	105
Шаг 12. Перенесите код в Power BI .....	106
Линейная диаграмма с затенением .....	107
Шаг 1. Загрузите исходные данные .....	109
Шаг 2. Загрузите данные в Power BI .....	110
Шаг 3. Создайте срезы в отчете .....	111
Шаг 4. Настройте визуальный элемент R в Power BI.....	111
Шаг 5. Экспортируйте данные в R Studio для дальнейшей разработки.....	112
Шаг 6. Загрузите необходимые пакеты .....	112
Шаг 7. Создайте переменные для проверки данных .....	112
Шаг 8. Выполните проверку данных .....	113
Шаг 9. Создайте новый датафрейм на основании датафрейма dataset.....	113
Шаг 10. Создайте переменные для динамических составляющих диаграммы.....	114
Шаг 11. Создайте наборы данных, необходимые для наложения тени .....	114
Шаг 12. Создайте наборы данных, необходимые для отрисовки графика.....	118
Шаг 13. Создайте символьный вектор для хранения цветовой схемы затенения .....	118
Шаг 14. Постройте диаграмму при помощи функции ggplot() .....	119
Шаг 15. Добавьте слой для создания затенения .....	119
Шаг 16. Добавьте линейную диаграмму на основании выбора пользователя .....	120
Шаг 17. Раскрасьте фоновую заливку в соответствии с предопределенной цветовой схемой партий .....	121
Шаг 18. Отформатируйте ось y в соответствии с выбором пользователя .....	121
Шаг 19. Добавьте метки на оси x и y.....	121
Шаг 20. Снабдите диаграмму динамическим заголовком и подзаголовком .....	122
Шаг 21. Измените внешний вид диаграммы в стиле журнала The Economist .....	122

Шаг 22. Перенесите код в Power BI .....	122
Карта .....	123
Шаг 1. Загрузите исходные данные .....	125
Шаг 2. Загрузите данные в Power BI .....	126
Шаг 3. Создайте срез в отчете на основании выбранного в фильтре штата.....	127
Шаг 4. Настройте визуальный элемент R в Power BI.....	127
Шаг 5. Экпортируйте данные в R Studio для дальнейшей разработки.....	127
Шаг 6. Загрузите необходимые пакеты .....	127
Шаг 7. Создайте переменные для проверки данных .....	127
Шаг 8. Выполните проверку данных .....	128
Шаг 9. Создайте переменные для заголовков диаграммы .....	128
Шаг 10. Добавьте к набору данных столбец с квинтилем .....	129
Шаг 11. Создайте символьный вектор для хранения цветовой схемы затенения.....	129
Шаг 12. Постройте диаграмму при помощи функции ggplot() .....	130
Шаг 13. Добавьте слой с картой.....	130
Шаг 14. Отформатируйте оси <i>x</i> и <i>y</i> .....	131
Шаг 15. Раскрасьте округа на основании квинтилей.....	131
Шаг 16. Улучшите отображение карты выбранного штата.....	133
Шаг 17. Снабдите диаграмму динамическим заголовком и подзаголовком .....	133
Шаг 18. Примените тему theme_map() .....	133
Шаг 19. Перенесите код в Power BI .....	133
Диаграмма квадрантов .....	134
Шаг 1. Загрузите исходные данные .....	137
Шаг 2. Загрузите данные в Power BI .....	138
Шаг 3. Создайте срезы в отчете по типу игры и четверти матча .....	138
Шаг 4. Настройте визуальный элемент R в Power BI.....	138
Шаг 5. Экпортируйте данные в R Studio для дальнейшей разработки.....	138
Шаг 6. Загрузите необходимые пакеты .....	139
Шаг 7. Создайте переменные для проверки данных .....	139
Шаг 8. Выполните проверку данных .....	140
Шаг 9. Создайте заголовки диаграммы .....	140
Шаг 10. Добавьте дополнительные столбцы в набор данных .....	140
Шаг 11. Постройте диаграмму при помощи функции ggplot() .....	141
Шаг 12. Используйте геометрию geom_point() для создания диаграммы рассеяния .....	141
Шаг 13. Добавьте метки игроков для всех квадрантов.....	141
Шаг 14. Добавьте горизонтальные и вертикальные линии, проходящие через центр .....	142
Шаг 15. Добавьте на диаграмму заголовки квадрантов .....	143
Шаг 16. Добавьте метки на оси <i>x</i> и <i>y</i> .....	145
Шаг 17. Снабдите диаграмму динамическими заголовками и подписями .....	145
Шаг 18. Примените тему theme_tufte.....	145
Шаг 19. Выполните финальную очистку.....	145

Шаг 20. Перенесите код в Power BI .....	148
Добавление линии регрессии.....	149
Шаг 1. Загрузите исходные данные .....	150
Шаг 2. Загрузите данные в Power BI .....	151
Шаг 3. Настройте визуальный элемент R в Power BI.....	151
Шаг 4. Экпортируйте данные в R Studio для дальнейшей разработки.....	151
Шаг 5. Загрузите необходимые пакеты .....	151
Шаг 6. Создайте переменные для проверки данных.....	151
Шаг 7. Выполните проверку данных .....	152
Шаг 8. Постройте диаграмму при помощи функции ggplot() .....	152
Шаг 9. Используйте геометрию geom_point() для создания диаграммы рассеяния .....	153
Шаг 10. Добавьте на визуализацию слой с линией регрессии .....	153
Шаг 11. Снабдите диаграмму заголовком и подзаголовком .....	154
Шаг 12. Примените тему .....	155
Шаг 13. Выполните финальную очистку.....	155
Шаг 14. Перенесите код в Power BI .....	155

## **Часть II. ЗАГРУЗКА ИНФОРМАЦИИ В МОДЕЛЬ ДАННЫХ POWER BI ПРИ ПОМОЩИ R И PYTHON .....**

### **Глава 3. Чтение файлов CSV .....**

Динамическое объединение файлов .....	158
Пример сценария.....	159
Выбор файлов за скользящий период из 24 месяцев при помощи R .....	159
Шаг 1. Импортируйте необходимые пакеты для скрипта .....	159
Шаг 2. Установите рабочую директорию на папку, содержащую наборы данных о продажах .....	160
Шаг 3. Считайте имена файлов в символьный вектор.....	161
Шаг 4. Создайте вектор дат .....	162
Шаг 5. Создайте датафрейм, состоящий из двух векторов.....	163
Шаг 6. Получите верхнюю и нижнюю границы желаемого диапазона дат.....	164
Шаг 7. Ограничьте датафрейм только нужными нам месяцами .....	164
Шаг 8. Создайте датафрейм на основании объединенных файлов .....	165
Шаг 9. Соберите написанный код и перенесите в редактор скриптов в Power BI.....	166
Выбор файлов за скользящий период из 24 месяцев при помощи Python .....	168
Шаг 1. Создайте скрипт на Python и загрузите необходимые библиотеки .....	168
Шаг 2. Установите рабочую директорию на папку Python_Code .....	168
Шаг 3. Загрузите перечень имен файлов в список .....	169
Шаг 4. Создайте датафрейм pandas с информацией о файлах для объединения.....	169

Шаг 5. Создайте новый столбец с датой в датафрейме.....	170
Шаг 6. Определите границы нужного нам диапазона дат.....	170
Шаг 7. Ограничьте датафрейм нужным диапазоном .....	170
Шаг 8. Объедините файлы в единый датафрейм .....	171
Шаг 9. Перенесите скрипт в Power BI.....	172
Фильтрация строк на основе регулярных выражений .....	174
Использование регулярных выражений в R .....	174
Шаг 1. Загрузите необходимые для работы пакеты .....	175
Шаг 2. Загрузите в R файл с потенциальными избирателями.....	175
Шаг 3. Определите регулярное выражение .....	175
Шаг 4. Исключите неправильные адреса из набора данных.....	176
Шаг 5. Объедините написанный код в один скрипт и перенесите в редактор скриптов в Power BI.....	176
Использование регулярных выражений в Python .....	176
Шаг 1. Загрузите необходимые для работы библиотеки .....	177
Шаг 2. Загрузите в Python файл с избирателями и присвойте его содержимое датафрейму .....	177
Шаг 3. Определите регулярное выражение .....	177
Шаг 4. Исключите неправильные адреса из набора данных.....	177
Шаг 5. Объедините написанный код в один скрипт и перенесите в редактор скриптов Python в Power BI.....	177

## Глава 4. Чтение данных из Microsoft Excel..... 179

Чтение файлов Excel при помощи R .....	180
Шаг 1. Импортируйте пакеты tidyverse и readxl.....	180
Шаг 2. Создайте оболочку функции combine_sheets.....	181
Шаг 3. Получите имена листов для объединения из указанной рабочей книги .....	181
Шаг 4. Преобразуйте символьный вектор, полученный на предыдущем шаге, в именованный символьный вектор.....	181
Шаг 5. Используйте функцию map_dfr() для объединения информации с листов в один датафрейм .....	182
Шаг 6. Верните датафрейм из функции .....	183
Шаг 7. Направьте рабочую директорию на папку с файлами Excel.....	183
Шаг 8. Сохраните в переменной excel_file_paths список файлов для обработки.....	184
Шаг 9. Используйте функцию map_dfr() для применения функции combine_sheets() ко всем выбранным файлам .....	184
Шаг 10. Скопируйте скрипт и вставьте в редактор скриптов R в Power BI через инструмент <b>Получить данные</b> (GetData) .....	184
Чтение файлов Excel при помощи Python.....	185
Шаг 1. Импортируйте библиотеки os и pandas .....	186
Шаг 2. Создайте оболочку функции combine_sheets() .....	186
Шаг 3. Создайте объект Excel на основании пути, переданного в функцию в аргументе excel_file_path .....	186
Шаг 4. Создайте список имен листов в рабочей книге.....	186

Шаг 5. Используйте метод <code>read_excel()</code> из библиотеки <code>pandas</code> для считывания данных в один датафрейм .....	187
Шаг 6. Верните датафрейм <code>df</code> из функции <code>combine_sheets</code> .....	187
Шаг 7. Установите рабочую директорию в папку, в которой находятся файлы Excel.....	187
Шаг 8. Получите список файлов в текущей рабочей директории и присвойте его переменной <code>excel_file_paths</code> .....	188
Шаг 9. Создайте пустой датафрейм и назовите его <code>combined_workbooks</code> ...	188
Шаг 10. Создайте заготовку для цикла <code>for</code> .....	188
Шаг 11. Объедините данные со всех листов в один датафрейм при помощи функции <code>combine_sheets()</code> .....	189
Шаг 12. Добавьте датафрейм <code>combined_workbook</code> к главному датафрейму <code>combined_workbooks</code> .....	189
Шаг 13. Скопируйте скрипт и вставьте в редактор скриптов Python в Power BI через инструмент <b>Получить данные</b> (GetData) .....	190
<b>Глава 5. Чтение данных из SQL Server</b> .....	191
Добавление базы данных AdventureWorksDW_StarSchema к вашему экземпляру SQL Server .....	191
Чтение данных из SQL Server в Power BI при помощи R .....	192
Шаг 1. Создайте DSN для подключения к базе данных SQL Server.....	193
Шаг 2. Создайте таблицу лога в SQL Server .....	196
Шаг 3. Начните написание скрипта на R для загрузки таблицы DimDate .....	197
Шаг 4. Создайте переменную для хранения имени загружаемой таблицы .....	197
Шаг 5. Создайте переменную для хранения SQL-выражения.....	197
Шаг 6. Создайте подключение к SQL Server .....	197
Шаг 7. Извлеките данные из SQL Server и сохраните их в датафрейм.....	198
Шаг 8. Получите текущее время.....	198
Шаг 9. Получите количество прочитанных записей.....	198
Шаг 10. Добавьте в датафрейм запись с информацией для сохранения в лог.....	198
Шаг 11. Сохраните собранную информацию в базе данных .....	199
Шаг 12. Закройте соединение .....	200
Шаг 13. Скопируйте написанный скрипт в Power BI .....	200
Шаг 14. Создайте скрипт для загрузки таблицы DimProduct на базе ReadLog_DimDate.R.....	201
Шаг 15. Создайте скрипт для загрузки таблицы DimPromotion.....	202
Шаг 16. Создайте скрипт для загрузки таблицы DimSalesTerritory на основе ReadLog_DimDate.R.....	202
Шаг 17. Создайте скрипт для загрузки таблицы FactInternetSales на основе ReadLog_DimDate.R.....	203
Чтение данных из SQL Server в Power BI при помощи Python .....	204
Шаг 1. Создайте DSN для SQL Server .....	204
Шаг 2. Создайте таблицу для ведения логов в SQL Server.....	204
Шаг 3. Создайте скрипт для загрузки таблицы DimDate.....	205



Шаг 4. Определите переменную для хранения имени таблицы, предназначенной для загрузки в Power BI.....	205
Шаг 5. Создайте подключение к базе данных с помощью библиотеки sqlalchemy .....	205
Шаг 6. Прочитайте содержимое таблицы DimDate и сохраните его в переменной df_read .....	206
Шаг 7. Получите текущую дату и время и сохраните в переменной timestamp.....	206
Шаг 8. Посчитайте количество записей в таблице DimDate.....	206
Шаг 9. Добавьте запись в датафрейм с информацией для сохранения логов .....	207
Шаг 10. Добавьте информацию, добытую на предыдущем шаге, в таблицу логов .....	207
Шаг 11. Скопируйте скрипт в Power BI .....	208
Шаг 12. Создайте скрипт для загрузки таблицы DimProduct на основе ReadLog_DimDate.py .....	208
Шаг 13. Создайте скрипт для загрузки таблицы DimPromotion на основе ReadLog_DimDate.py .....	209
Шаг 14. Создайте скрипт для загрузки таблицы DimSalesTerritory на основе ReadLog_DimDate.py .....	210
Шаг 15. Создайте скрипт для загрузки таблицы FactInternetSales на основе ReadLog_DimDate.py .....	211
<b>Глава 6. Чтение в модель данных Power BI посредством API.....</b>	<b>212</b>
Чтение и загрузка данных в Power BI из API с помощью R.....	212
Шаг 1. Получите персональный ключ API к Census .....	212
Шаг 2. Загрузите необходимые пакеты R .....	213
Шаг 3. Определите переменные для возврата из вашего набора данных.....	213
Шаг 4. Создайте символьный вектор, содержащий нужные вам переменные .....	214
Шаг 5. Сконфигурируйте функцию get_acs .....	215
Шаг 6. Присвойте переменным (столбцам) осмысленные имена .....	215
Шаг 7. Скопируйте скрипт в Power BI.....	216
Чтение и загрузка данных в Power BI из API с помощью Python.....	217
Шаг 1. Получите персональный ключ API к Census .....	217
Шаг 2. Загрузите необходимые библиотеки Python .....	218
Шаг 3. Определите переменные для возврата из вашего набора данных.....	218
Шаг 4. Создайте список, содержащий нужные вам переменные .....	219
Шаг 5. Создайте список кортежей с географическими фильтрами для набора данных .....	220
Шаг 6. Извлеките данные при помощи функции censusdata.download() ....	220
Шаг 7. Переиндексируйте датафрейм, созданный на шестом шаге.....	221
Шаг 8. Дайте столбцам осмысленные имена .....	221
Шаг 9. Переименуйте столбцы в датафрейме.....	221
Шаг 10. Скопируйте скрипт в Power BI .....	222
Заключение .....	222

## Часть III. ПРЕОБРАЗОВАНИЕ ДАННЫХ ПРИ ПОМОЩИ R И PYTHON ..... 223

### Глава 7. Продвинутое строковое операции и распознавание шаблонов ..... 224

Защита конфиденциальных сведений .....	225
Защита конфиденциальных сведений в Power BI с помощью R .....	225
Шаг 1. Импортируйте пакеты tidyverse и stringr.....	225
Шаг 2. Напишите функцию для очистки данных.....	226
Шаг 3. Считайте комментарии в датафрейм .....	228
Шаг 4. Скройте телефонные номера и номера социального страхования в поле комментария.....	228
Шаг 5. Скопируйте скрипт в Power BI .....	229
Защита конфиденциальных сведений в Power BI с помощью Python.....	229
Шаг 1. Импортируйте библиотеки pandas, os и re .....	230
Шаг 2. Напишите функцию mask_text() .....	230
Шаг 3. Установите рабочую директорию.....	232
Шаг 4. Считайте комментарии в датафрейм .....	232
Шаг 5. Выполните замену телефонных номеров и номеров социального страхования.....	232
Шаг 6. Скопируйте скрипт в Power BI .....	232
Подсчет количества слов и предложений в обзорах.....	233
Подсчет количества слов и предложений в обзорах с помощью R .....	233
Шаг 1. Импортируйте библиотеки tidyverse и stringr.....	233
Шаг 2. Измените рабочую директорию .....	234
Шаг 3. Считайте информацию из файла .....	234
Шаг 4. Ограничьте набор данных требуемыми столбцами.....	234
Шаг 5. Добавьте столбцы с количеством слов и предложений .....	234
Шаг 6. Скопируйте скрипт в Power BI .....	235
Подсчет количества слов в обзорах с помощью Python.....	235
Шаг 1. Импортируйте библиотеки pandas и os.....	235
Шаг 2. Установите рабочую директорию.....	236
Шаг 3. Считайте информацию из файла .....	236
Шаг 4. Создайте в датафрейме столбец word_count .....	236
Шаг 5. Скопируйте скрипт в Power BI .....	236
Удаление имен неподходящего формата .....	237
Удаление имен неподходящего формата с помощью R .....	237
Шаг 1. Импортируйте пакеты tidyverse и stringr.....	237
Шаг 2. Установите рабочую директорию.....	237
Шаг 3. Создайте регулярное выражение с правильным шаблоном имени .....	238
Шаг 4. Считайте данные в датафрейм .....	239
Шаг 5. Выполните обновление столбца Name .....	239
Шаг 6. Скопируйте скрипт в Power BI .....	239
Удаление имен неподходящего формата с помощью Python.....	239
Шаг 1. Импортируйте библиотеки pandas, re и os .....	240

Шаг 2. Установите рабочую директорию.....	240
Шаг 3. Считайте данные из файла DimEmployee.csv в датафрейм .....	240
Шаг 4. Создайте регулярное выражение, соответствующее правильному формату имени .....	240
Шаг 5. Скомпилируйте регулярное выражение .....	241
Шаг 6. Напишите функцию для проверки имен на совместимость с шаблоном.....	241
Шаг 7. Примените функцию к столбцу, чтобы избавиться от лишних имен .....	242
Шаг 8. Скопируйте скрипт в Power BI .....	242
Определение шаблонов в строках на основании условной логики .....	243
Поиск шаблонов в строках на основании условной логики с помощью R.....	244
Шаг 1. Импортируйте пакеты tidyverse и stringr.....	244
Шаг 2. Установите рабочую директорию.....	244
Шаг 3. Напишите функцию для поиска изделий .....	245
Шаг 4. Считайте данные из файла ProductionOrders.csv в датафрейм....	246
Шаг 5. Добавьте в датафрейм df столбец Monitored Products .....	246
Шаг 6. Скопируйте скрипт в Power BI .....	246
Поиск шаблонов в строках на основании условной логики с помощью Python .....	247
Шаг 1. Импортируйте библиотеки pandas, re и os .....	247
Шаг 2. Установите рабочую директорию.....	247
Шаг 3. Скомпилируйте регулярное выражение .....	247
Шаг 4. Напишите функцию для распознавания нужных нам деталей....	248
Шаг 5. Считайте данные в датафрейм Pandas .....	248
Шаг 6. Создайте новый столбец с именем Monitored Products.....	249
Шаг 7. Скопируйте скрипт в Power BI.....	249
Заключение .....	249
<b>Глава 8. Вычисляемые столбцы с помощью R и Python .....</b>	<b>250</b>
Создание ключа Google Geocoding API .....	251
Шаг 1. Зайдите в консоль Google.....	251
Шаг 2. Настройте учетную запись .....	251
Шаг 3. Добавьте новый проект.....	251
Шаг 4. Активируйте API геокодирования.....	252
Геокодирование адресов с помощью R .....	254
Геокодирование адресов с помощью Python .....	256
Вычисление расстояния между точками с помощью пользовательской функции в R .....	258
Вычисление расстояния между точками с помощью пользовательской функции в Python.....	260
Вычисление расстояния между точками с помощью готовой функции в R...	263
Вычисление расстояния между точками с помощью готовой функции в Python.....	264
Заключение .....	266

## Часть IV. МАШИННОЕ ОБУЧЕНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В POWER BI ПРИ ПОМОЩИ R И PYTHON .....267

### Глава 9. Применение методов машинного обучения и искусственного интеллекта к моделям данных Power BI..... 268

Применение алгоритмов машинного обучения к набору данных перед загрузкой в модель Power BI.....	269
Прогнозирование цен на недвижимость с помощью R.....	270
Шаг 1. Пусть аналитик данных сохранит для вас модель.....	270
Шаг 2. Загрузите пакет tidyverse.....	270
Шаг 3. Загрузите объект модели и набор данных для оценки.....	271
Шаг 4. Ограничьте датафрейм столбцами, необходимыми для вашей модели.....	271
Шаг 5. Примените модель машинного обучения к своему набору данных для составления прогноза цен на недвижимость.....	271
Шаг 6. Добавьте прогноз к исходному набору данных.....	272
Шаг 7. Скопируйте скрипт в Power BI.....	272
Прогнозирование цен на недвижимость с помощью Python.....	272
Шаг 1. Пусть аналитик данных сохранит для вас модель.....	273
Шаг 2. Загрузите необходимые библиотеки.....	273
Шаг 3. Загрузите объект модели и актуальный набор данных.....	273
Шаг 4. Извлеките нужную информацию из датафрейма.....	274
Шаг 5. Примените модель к подготовленному набору данных для расчета прогноза.....	274
Шаг 6. Добавьте прогнозные данные к исходному набору данных.....	275
Шаг 7. Скопируйте скрипт в Power BI.....	275
Использование готовых моделей ИИ для расширения функционала моделей данных в Power BI.....	276
Настройка Cognitive Services в Azure.....	277
Виртуальная машина для анализа данных (Data Science Virtual Machine – DSVM).....	277
Анализ тональности текста в Microsoft Cognitive Services при помощи Python.....	278
Шаг 1. Загрузите набор данных с отзывами Yelp с сайта Kaggle.....	279
Шаг 2. Загрузите нужные библиотеки, модули и функции для скрипта.....	279
Шаг 3. Инициализируйте переменные для работы скрипта.....	280
Шаг 4. Считайте фрагмент файла с отзывами в датафрейм.....	280
Шаг 5. Преобразуйте датафрейм к формату, приемлемому для службы Microsoft Cognitive Services.....	281
Шаг 6. Оцените отзывы посетителей при помощи метода sentiment().....	281
Шаг 7. Создайте датафрейм, содержащий оценки отзывов.....	282
Шаг 8. Скопируйте скрипт в Power BI.....	282
Применение сторонних моделей машинного обучения к моделям данных Power BI.....	283

Конфигурирование средства анализа настроения текста в IBM Watson.....	284
Шаг 1. Заведите аккаунт в IBM Cloud.....	284
Шаг 2. Выполните вход в IBM Cloud.....	284
Шаг 3. Перейдите на страницу Tone Analyzer.....	284
Шаг 4. Настройте службу Tone Analyzer.....	284
Шаг 5. Получите ключ API.....	285
Написание скрипта на Python для анализа настроения текста в IBM Watson.....	286
Шаг 1. Импортируйте необходимые библиотеки и модули.....	286
Шаг 2. Создайте экземпляр класса IAMAuthenticator.....	286
Шаг 3. Создайте экземпляр класса ToneAnalyzerV3.....	286
Шаг 4. Установите ссылку на службу для созданного объекта.....	287
Шаг 5. Создайте датафрейм с исходными данными для анализа.....	287
Шаг 6. Создайте основу для датафрейма с оценочными данными.....	287
Шаг 7. Определите циклическую конструкцию для отправки документов в службу IBM Watson.....	288
Шаг 8. Отформатируйте и оцените настроение текста в документе.....	288
Шаг 9. Извлеките результат анализа текста и инициализируйте переменные исходными значениями.....	289
Шаг 10. Пройдите по тонам и присвойте их значения соответствующим переменным.....	290
Шаг 11. Создайте датафрейм на основе списка listReturnedUtterance.....	291
Шаг 12. Объедините датафреймы dfReturnedUtterance и dfDocuments.....	291
Шаг 13. Скопируйте скрипт в Power BI.....	292

## **Глава 10. Создание моделей анализа данных и скриптов для обработки информации.....**

Прогнозирование цен на недвижимость в Power BI с помощью R со службой SSMLS.....	295
Написание скрипта на языке R для добавления модели в SQL Server.....	295
Шаг 1. Загрузите необходимые пакеты.....	296
Шаг 2. Загрузите модель R в вашу сессию.....	296
Шаг 3. Подключитесь к базе данных.....	296
Шаг 4. Определите переменные модели.....	296
Шаг 5. Напишите выражение на T-SQL для добавления модели в базу данных.....	297
Шаг 6. Добавьте код, необходимый для запуска выражения T-SQL из R.....	297
Шаг 7. Сохраните скрипт.....	298
Использование SSMLS совместно с R для оценки данных.....	298
Шаг 1. Запустите SQL Server Management Studio.....	298
Шаг 2. Создайте подключение к серверу, который хотите использовать.....	298
Шаг 3. Добавьте базу данных BostonHousingInfo на ваш сервер.....	298
Шаг 4. Добавьте модель в базу данных.....	299
Шаг 5. Создайте в базе данных хранимую процедуру для прогноза.....	299

Шаг 6. Извлеките данные с прогнозами из SQL Server в Power BI.....	301
Прогнозирование цен на недвижимость в Power BI с помощью Python со службой SSMLS.....	303
Написание скрипта на языке Python для добавления модели в SQL Server .....	303
Шаг 1. Подберите версии библиотек.....	303
Шаг 2. Создайте окружение conda.....	304
Шаг 3. Напишите код для загрузки модели в SQL Server .....	305
Использование SSMLS совместно с Python для оценки данных.....	307
Шаг 1. Запустите SQL Server Management Studio .....	307
Шаг 2. Создайте подключение к серверу, который хотите использовать .....	307
Шаг 3. Добавьте базу данных BostonHousingInfo на ваш сервер .....	307
Шаг 4. Добавьте модель в базу данных.....	307
Шаг 5. Создайте в базе данных хранимую процедуру для прогноза .....	308
Шаг 6. Извлеките данные с прогнозами из SQL Server в Power BI.....	310
Анализ тональности текста в Power BI с помощью R со службой SSMLS.....	311
Добавление готовых моделей R в SSMLS с помощью PowerShell .....	311
Шаг 1. Проверьте, установлены ли предварительно обученные модели.....	311
Шаг 2. Откройте PowerShell от имени администратора .....	312
Шаг 3. Загрузите скрипт PowerShell .....	312
Шаг 4. Запустите загруженный скрипт в PowerShell .....	312
Решение проблем .....	312
Использование готовой модели R в SSMLS для анализа тональности текста в Power BI .....	312
Шаг 1. Определите хранимую процедуру .....	313
Шаг 2. Определите переменные .....	313
Шаг 3. Инициализируйте переменную @Query .....	313
Шаг 4. Инициализируйте переменную @RScript .....	314
Шаг 5. Сконфигурируйте процедуру sp_execute_external_script .....	315
Шаг 6. Определите выходные данные .....	316
Шаг 7. Создайте хранимую процедуру в базе данных.....	316
Шаг 8. Вызовите процедуру из Power BI.....	317
Анализ тональности текста в Power BI с помощью Python со службой SSMLS .....	318
Добавление готовых моделей Python в SSMLS.....	319
Шаг 1. Проверьте, установлены ли предварительно обученные модели.....	319
Шаг 2. Откройте PowerShell от имени администратора .....	319
Шаг 3. Загрузите скрипт PowerShell .....	319
Шаг 4. Запустите загруженный скрипт в PowerShell .....	319
Решение проблем .....	320
Использование готовой модели Python в SSMLS для анализа тональности текста в Power BI .....	320
Шаг 1. Определите хранимую процедуру .....	320
Шаг 2. Определите переменные .....	321

---

Шаг 3. Инициализируйте переменную @Query .....	321
Шаг 4. Инициализируйте переменную @PythonScript.....	321
Шаг 5. Сконфигурируйте процедуру sp_execute_external_script .....	322
Шаг 6. Определите выходные данные .....	322
Шаг 7. Создайте хранимую процедуру в базе данных.....	322
Шаг 8. Вызовите процедуру из Power BI.....	323
Вычисление расстояния между точками в Power BI с помощью R со службой SSMLS.....	324
Шаг 1. Убедитесь, что в SSMLS загружен пакет dplyr.....	325
Шаг 2. Запустите SSMS и подключитесь к SQL Server .....	325
Шаг 3. Добавьте базу данных CalculateDistance на ваш сервер.....	325
Шаг 4. Создайте хранимую процедуру для расчета расстояний.....	326
Шаг 5. Вызовите процедуру из Power BI .....	328
Вычисление расстояния между точками в Power BI с помощью Python со службой SSMLS.....	329
Шаг 1. Запустите SSMS и подключитесь к SQL Server .....	330
Шаг 2. Добавьте базу данных CalculateDistance на ваш сервер.....	330
Шаг 3. Создайте хранимую процедуру для расчета расстояний.....	331
Шаг 4. Вызовите процедуру из Power BI .....	333
<b>Предметный указатель.....</b>	<b>335</b>

# Об авторе



Райан Уэйд (Ryan Wade) является профессиональным аналитиком данных с более чем 20-летним стажем. Своему образованию и опыту работы Райан обязан тем, что приобрел целостное понимание аналитических процессов как с технической, так и с организационной точки зрения. Является обладателем сертификата MCSE в области бизнес-аналитики и Microsoft R и профессионально программирует на R, Python, DAX, T-SQL, M и VBA применительно к локальным и облачным решениям в аналитике на *платформе данных Microsoft (Microsoft Data Platform)*.

Райан принимает активное участие в открытых мероприятиях по R и Python, а также выступает на конференциях SQLSaturdays, TDWI, BDPA и PASS Summit с лекциями по анализу данных. Разработал полноценный онлайн-курс для ExcelTv, в рамках которого демонстрирует применение языка R в Power BI для проведения углубленного анализа данных и их визуализации.



# Введение

Microsoft Power BI является одним из наиболее популярных инструментов в области бизнес-аналитики. За последние годы этот программный комплекс опередил своих прямых конкурентов QlikView и Tableau и прочно занял лидирующее положение на рынке. Одним из главных преимуществ Power BI является то, что он представляет собой нечто гораздо большее, чем просто инструмент визуализации данных. Вот лишь несколько явных преимуществ Power BI:

- встроенный язык запросов *DAX*, позволяющий крайне эффективно извлекать информацию из модели данных с применением сложной бизнес-логики;
- интегрированный инструмент подготовки и преобразования данных *Power Query*, при помощи которого можно легко извлекать и трансформировать исходную информацию в вид, пригодный для анализа;
- движок *Vertipaq*, позволяющий хранить данные в оптимальном для формирования отчетов виде и быстро и эффективно обрабатывающий сложные вычисления;
- заранее подготовленные пакеты интерактивных элементов визуализации, помогающие представлять данные в понятной и четкой форме.

Глядя на этот список преимуществ, вы вполне можете задаться вопросом, зачем же столь мощному инструменту понадобилась помощь языков программирования R и Python. Ответ прост – чтобы заполнить области, в которых встроенные средства недостаточно хороши. Вот лишь несколько примеров применения этих языков программирования в рамках Power BI:

- создание пользовательских элементов визуализации без особых усилий;
- реализация интеллектуальной обработки данных, методов машинного обучения и искусственного интеллекта без необходимости приобретать дорогостоящую подписку на *Power BI Premium*;
- применение продвинутых методов обработки текстовой информации с использованием техник, недоступных в *Power Query* и *DAX*;
- взаимодействие со службами *Microsoft Cognitive Services* без необходимости приобретать подписку на *Power BI Premium*;
- взаимодействие со сторонними интерфейсами API с целью эффективного обогащения моделей данных Power BI;
- и многое другое...

В данной книге мы подробно расскажем о том, как использовать на практике языки программирования R и Python для обеспечения всей перечисленной выше функциональности в Power BI. Язык R идеально подходит для Power BI по причине того, что он был создан специально для анализа данных. Уже долгие годы аналитики активно используют R для преобразования и визуализации информации. Так что то немногое, на что не способна программная среда Power BI, может быть с лихвой компенсировано при помощи языка R.

Что касается Python, то этот язык программирования приобрел чрезвычайную популярность в области анализа данных в последнее десятилетие. Одним

из главных преимуществ Python является то, что он подходит для решения не только аналитических задач, но и общих задач программирования. К примеру, взаимодействие с интерфейсами API довольно легко осуществить при помощи Python, тогда как посредством Power Query это будет сделать не так-то просто.

Все эти особенности делают языки R и Python идеально подходящими для такого мощного аналитического инструмента, как Power BI. И в книге, которую вы держите в руках, мы проиллюстрируем на примерах все перечисленные выше возможности и техники. При этом все решения и сценарии будут сопровождаться подробными описаниями, чтобы вы досконально поняли используемую реализацию.

Но перед тем как приступить к конкретным примерам, необходимо для начала настроить среду выполнения. Давайте пройдемся по соответствующим пунктам и подготовимся к работе.

## НАСТРОЙКА ВАШЕГО ОКРУЖЕНИЯ AZURE

В данной книге мы будем часто упоминать и пользоваться различными составляющими облачной платформы *Microsoft Azure*. В частности, для внедрения механизмов искусственного интеллекта в модели данных Power BI мы будем использовать набор служб *Microsoft Cognitive Services*. Также для работы с примерами из данной книги рекомендуется настроить *виртуальную машину для анализа данных* (Data Science Virtual Machine – DSVM). Это не обязательное условие, но желательное. Кроме того, для комфортной работы с тем, что мы будем обсуждать, вам необходимо настроить окружение со следующими инструментами:

- SQL Server 2017 или выше;
- SQL Server Machine Learning Services 2017 или выше с поддержкой R и Python;
- дистрибутив Python из Anaconda;
- R;
- R Studio;
- VS Code;
- Power BI Desktop.

Сконфигурировать такое окружение вручную – задача не из легких, но если вы установите виртуальную машину для анализа данных, большинство настроек будет выполнено за вас автоматически. В следующих разделах мы поговорим о том, как настроить *Azure*, чтобы можно было пользоваться службами *Microsoft Cognitive Services*. Кроме того, вы узнаете, как развернуть *DSVM*.

## Подписка на Azure

Оформить подписку на *Azure* можно по адресу <https://azure.microsoft.com/en-us/free>. В результате вы получите 12 месяцев на пользование выборочными службами плюс кредит на сумму \$200 на первый месяц.

## Подписка на Microsoft Cognitive Services

Мы будем использовать службу *Microsoft Cognitive Services* для проведения *анализа тональности текста* (sentiment analysis) в Power BI с помощью Python. Для начала вам необходимо будет настроить службу *Microsoft Cognitive Services* в Azure, после чего можно будет обращаться к ней из Power BI. Для настройки службы нужно выполнить следующие шаги.

1. Войдите на портал Azure.
2. Введите в строке поиска *Cognitive Services* и нажмите **Enter**.
3. Вы должны оказаться на странице *Cognitive Services*. Нажмите кнопку создания для запуска процесса регистрации.
4. Введите следующую информацию:
  - имя;
  - подписка;
  - расположение;
  - ценовая категория;
  - группа ресурсов.
5. Установите флажок, оповещающий о том, что вы прочитали и согласны с условиями.
6. Нажмите кнопку создания.

Учтите, что использование службы *Microsoft Cognitive Services* является платным. Чтобы получить информацию о ценах, перейдите к ресурсу *Microsoft Cognitive Services*, введите в окно поиска текст *Ценовая категория* и откройте результаты поиска. Выберите подходящую вам категорию. Вы будете перенаправлены на страницу с информацией о ценах согласно выбранному вами региону. На примеры, показанные в данной книге, вам с лихвой хватит выданного вам на первый месяц кредита.

## Создание виртуальной машины для анализа данных (DSVM)

Предпочтительным вариантом будет создание виртуальной машины и добавление ресурсов, не установленных в ней по умолчанию. Я рекомендую идти этим путем, поскольку полностью ручная настройка окружения для выполнения примеров из этой книги может занять уйму времени. Использование виртуальной машины позволит сконфигурировать необходимое окружение за несколько минут – притом что в ручном режиме у вас бы это могло отнять много дней проб и ошибок. Если вы решите пойти длинным путем, позже я опишу примерный план действий. Сейчас же приведу инструкции по настройке виртуальной машины.

### *Шаги создания виртуальной машины в Azure*

1. Перейдите на портал <https://portal.azure.com>. Если система попросит, введите данные учетной записи, созданной ранее.

2. Нажмите на кнопку **Создать ресурс** (Create a resource) в левом верхнем углу.
3. В строке поиска введите **Data Science Virtual Machine – Windows 2019**.
4. Нажмите на кнопку **Создать** (Create). Появится форма для ввода информации о конфигурации виртуальной машины, в которой будет открыта вкладка **Основные** (Basics). В следующих шагах вы узнаете, как заполнить эту форму.
5. **Подписка** (Subscription): выберите подписку, которую собираетесь использовать. По умолчанию будет выбрана подписка, настроенная ранее.
6. **Группа ресурсов** (Resource group): если у вас уже есть группа ресурсов, которую вы желаете использовать, выберите ее. В противном случае создайте новую для своей виртуальной машины.
7. **Имя виртуальной машины** (Virtual machine name): название, которое вы хотите присвоить виртуальной машине.
8. **Регион** (Region): ближайший к вам географический регион Azure.
9. **Image**: убедитесь, что в данном списке выбран пункт **Data Science Virtual Machine – Windows 2016**.
10. **Размер** (Size): я использую B4ms, поскольку это самый дешевый вариант для 16 Гб оперативной памяти. Этот параметр очень важен для R, Python и Power BI.
11. **Имя пользователя** (Username): придумайте имя пользователя.
12. **Пароль** (Password): введите пароль.
13. **Подтвердите пароль** (Confirm password): введите пароль еще раз.
14. Перейдите на вкладку **Диски** (Disks).
15. **Тип диска ОС** (OS disk type): укажите **Стандартный SSD** (Standard SSD) – это будет оптимальный для нашего случая.
16. Перейдите на вкладку **Сетевые подключения** (Networking).
17. Убедитесь, что все нужные поля заполнены. Обязательные к заполнению поля помечены звездочкой. В эти поля должны быть введены значения по умолчанию. Если таких значений нет, нажмите на ссылку **Создать** (Create new) под соответствующим полем и введите недостающий элемент.
18. Перейдите на вкладку **Управление** (Management).
19. Оставьте все по умолчанию и перейдите на вкладку **Дополнительно** (Advanced).
20. Примите значения по умолчанию и перейдите на вкладку **Теги** (Tags).
21. Откройте вкладку **Просмотр и создание** (Review + create).
22. Вы увидите сводную информацию о создаваемой виртуальной машине. Кроме того, для вас будет рассчитана стоимость ее использования. Если вы согласны с введенными данными, нажмите кнопку **Создать** (Create).

*Не забывайте останавливать виртуальную машину после каждого использования, чтобы не платить лишние деньги.* На всякий случай вы можете настроить автоматическое отключение машины в определенное время. Для этого необходимо сделать следующее.

1. Перейдите к своей виртуальной машине на портале Azure.
2. Введите текст **Автозавершение работы** (auto-shutdown) в строку поиска и перейдите в соответствующий раздел.
3. Переведите переключатель **Включено** (Enabled) в положение **Вкл** (On).
4. Выберите время, в которое машина будет выключаться, в поле **Запланированное завершение работы** (Scheduled shutdown).
5. В выпадающем списке **Часовой пояс** (Time zone) выберите нужную зону.
6. Если хотите, чтобы система отправляла вам уведомление о выключении виртуальной машины, установите переключатель в разделе **Отправлять уведомление перед автоматическим завершением работы** (Send notification before auto-shutdown) в положение **Да** (Yes). Уведомление будет отправлено на адрес, введенный в поле **Адрес электронной почты** (Email address).

## Настройка R на виртуальной машине

Мы будем использовать другой дистрибутив языка R по сравнению с установленным. Нашим выбором будет дистрибутив *Microsoft R Open (MRO)*. Он полностью совместим с дистрибутивом, распространяемым через центральную систему хранения пакетов CRAN, но при этом существенно улучшен в отношении определенных типов вычислений, а также снабжен дополнительными полезными инструментами. Выполните следующие шаги, чтобы загрузить дистрибутив *MRO* в виртуальную машину.

1. Узнайте версию R, используемую в Power BI. Соответствующую информацию можно найти на сайте Microsoft по адресу <https://docs.microsoft.com/en-us/power-bi/visuals/service-r-visuals>.
2. Откройте браузер в виртуальной машине и перейдите по следующей ссылке: <https://mran.microsoft.com/open>. В виртуальной машине по умолчанию установлены два браузера: *Microsoft Edge* и *Firefox*.
3. Нажмите на кнопку **Download** справа, и вы будете перенаправлены на страницу загрузок. Здесь щелкните по ссылке **Past Releases** справа, которая откроет страницу со всеми предыдущими версиями *Microsoft R Open*. Выберите версию, которую использует Power BI.
4. Нажмите на кнопку **Download** напротив версии для *Windows*.
5. Выполните установку загруженного дистрибутива.
6. Откройте R Studio на виртуальной машине.
7. Откройте меню **Tools => Global Options** и убедитесь, что выбрана версия MRO, которую вы только что установили. Если это не так, нажмите на кнопку **Change...** и выберите нужный дистрибутив из списка, после чего нажмите на кнопку **OK**.

## Настройка Python на виртуальной машине

Одним из преимуществ использования виртуальной машины является то, что вы получаете предустановленный дистрибутив Python, идеально под-

ходящий для анализа данных. Этот дистрибутив называется *Anaconda*. Он поставляется с более чем 1500 библиотек, популярных в среде анализа данных. Также вместе с ним идет *диспетчер пакетов* (package manager) и *система управления окружением* (environment management system) под названием *conda*. Инсталлировать пакеты предпочтительно именно посредством *conda* по причине установки правильных зависимостей между ними. Система управления окружением *conda* значительно облегчает задачу создания изолированной копии Python с предустановленными библиотеками нужных версий.

Давайте для примеров из этой книги создадим отдельное окружение Python с именем *pbi*. Для этого необходимо выполнить следующие действия.

1. Подключитесь к виртуальной машине.
2. Откройте командную строку, нажав на значок поиска рядом с иконкой *Windows* и введя команду *cmd*.
3. Введите следующую команду для создания окружения *conda* с именем *pbi* на базе Python 3.7:

```
conda create -n pbi python=3.7
```

Решение использовать Python версии 3.7 основывается на информации, полученной из инструкции от Microsoft по адресу <https://docs.microsoft.com/en-us/business-applications-release-notes/october18/intelligence-platform/power-bi-service/pervasive-artificial-intelligence-bi/python-service>. Согласно документации, службы Power BI совместимы с Python 3.x, так что текущая версия 3.x должна подойти.

Теперь у нас есть окружение Python, которое мы можем использовать для примеров из данной книги.

## Настройка SQL Server Machine Learning Services на виртуальной машине

В нескольких примерах из этой книги вам потребуется наличие служб машинного обучения SQL Server (*SQL Server Machine Learning Services – SSMLS*). *SSMLS* предоставляет вам инструменты, позволяющие проводить углубленную аналитику в базах данных с использованием R и Python, а также средства, облегчающие работу с большими данными. Еще в составе *SSMLS* есть несколько предварительно обученных моделей от Microsoft, которыми вы можете пользоваться в процессе учебы и работы. В виртуальной машине по умолчанию запущены службы *SSMLS*. Если вы не используете виртуальную машину, вам необходимо будет вручную запустить эти службы, воспользовавшись подробной инструкцией по адресу <https://docs.microsoft.com/en-us/sql/machine-learning/install/sql-machine-learning-services-windows-install?view=sql-server-ver15>. Предварительно обученные модели, которые мы будем использовать в этой книге, могут быть добавлены в ваш экземпляр SQL Server поверх базовой установки. Перейдите по следующей ссылке и следуйте инструкциям: <https://docs.microsoft.com/en-us/sql/machine-learning/install/sql-pretrained-models-install?view=sql-server-ver15>.

## Установка пакетов R

Некоторые скрипты на языке R из этой книги могут ссылаться на пакеты, которые у вас изначально могут быть не установлены. Исправить это очень просто. Следующая инструкция в консоли R позволит установить популярный пакет с названием *data.table*:

```
install.packages("data.table")
```

Бывает, что вам требуется установить сразу несколько пакетов за раз. Например, вы хотите одновременно установить пакеты *data.table* и *dplyr*. Для этого достаточно объединить названия этих пакетов в вектор и присвоить результат переменной *pkgs*. Затем можно передать эту переменную на вход функции *install.packages()*, как показано ниже:

```
pkgs <- c("data.table", "dplyr")
install.packages(pkgs)
```

**Примечание.** Символьный вектор представляет собой тип данных в языке R для хранения текстовой информации в виде одномерного массива. Вы узнаете больше об этом и других типах данных в R в процессе чтения книги.

При создании визуальных элементов в Power BI при помощи R вам необходимо знать, какую версию пакета использует служба Power BI. Вы можете получить список всех доступных пакетов R в Power BI вместе с их версиями по ссылке <https://docs.microsoft.com/en-us/power-bi/service-r-packages-support>.

При помощи функции *install.packages()* можно установить последнюю версию пакета из репозитория, который вы используете, если у вас дистрибутив R из CRAN. При использовании дистрибутива *Microsoft R Open* будет установлена последняя версия пакета, основываясь на *дате снимка* (snapshot date). В обоих случаях может получиться так, что будет установлен пакет не той версии, с которой работает служба. Чтобы устранить это неудобство, необходимо сначала узнать требуемую версию пакета по ссылке выше, после чего загрузить ее при помощи пакета *devtools*. Ниже приведен пример использования пакета *devtools* для установки пакета *ggplot2* версии 0.9.1 из CRAN:

```
library(devtools)
install_version(
  "ggplot2",
  version = "0.9.1",
  repos = "http://cran.us.r-project.org")
```

## Установка библиотек Python

Установить библиотеки Python можно разными способами. В данной книге мы будем использовать два метода: при помощи *conda* и при помощи *pip*.

Стоит отметить, что установка библиотек в Python выполняется не так просто, как в R. В книге мы будем в основном использовать командную строку *conda* для установки библиотек Python. Для этого необходимо выполнить следующие действия.

1. Откройте строку поиска, иконка которой расположена рядом со значком *Windows*.
2. Введите слово *Anaconda*, после чего в окне выбора появится вариант *Anaconda Prompt*. Щелкните по нему.
3. Активируйте окружение, которое используете для Power BI, введя следующую команду в командную строку:

```
conda activate "<environment name>"
```

Рекомендуется использовать окружение для разработки на Python, ассоциированное с этой книгой.

4. Установите пакет при помощи *conda*, используя следующий шаблон:

```
conda install <"package name">
```

Например, если вы устанавливаете пакет *pandas*, команда должна иметь следующий вид:

```
conda install pandas
```

Если вам необходимо установить пакет *pandas* версии 1.0.4, используйте следующую команду:

```
conda install pandas=1.0.4
```

Не все пакеты доступны для установки при помощи *conda*. Обратитесь к следующей ссылке для получения списка пакетов, которые могут быть установлены с использованием *conda* в Python 3.6: [https://docs.anaconda.com/anaconda/packages/py3.6\\_win-64](https://docs.anaconda.com/anaconda/packages/py3.6_win-64). Один из пакетов, который мы будем использовать в этой книге, недоступен в *conda*, но может быть установлен при помощи *PyPI*. Имя этого пакета *CensusData*. Вам необходимо использовать систему управления пакетами *pip* для установки библиотеки *CensusData*, как показано ниже:

```
pip install CensusData
```

## Настройка Power BI на виртуальной машине

В *Power BI Desktop* необходимо выполнить ряд изменений в настройках, чтобы можно было работать с R и Python. Подробно эти изменения описаны в репозитории кода книги на *GitHub*. Здесь вы также найдете инструкции по созданию и использованию окружения *conda* в Python. Ссылка на репозиторий кода: <https://github.com/Apress/adv-analytics-in-power-bi-w-r-and-python>.



## АЛЬТЕРНАТИВНАЯ НАСТРОЙКА

Использование виртуальной машины – предпочтительная, но вовсе не обязательная опция для работы с примерами из этой книги. Если вы хотите пойти другим путем, вам придется устанавливать все программное обеспечение вручную. Приводим ссылки на все необходимые установки:

- Power BI: <http://www.microsoft.com/en-us/download/details.aspx?id=58494>;
- R Studio: <https://rstudio.com/products/rstudio/download>;
- Microsoft R Open: <https://mran.microsoft.com/download>;
- Anaconda: <http://www.anaconda.com/products/individual>;
- VS Code: <https://code.visualstudio.com/download>;
- SQL Server 2019 Developer: <http://www.microsoft.com/en-us/sql-server/sql-server-downloads>;
- SQL Server Machine Learning Services: <https://docs.microsoft.com/en-us/sql/machine-learning/install/sql-machine-learning-services-windows-install?view=sql-server-ver15>.

Если вы остановите выбор на этом варианте, я очень рекомендую использовать виртуальную машину на базе *Windows Server 2016* или выше. Предлагаю ссылку YouTube на пошаговую инструкцию по установке *Windows Server 2019* на *VirtualBox*: <http://www.youtube.com/watch?v=ZjQSuyuN0nA&t=8s>.

## ЗАГРУЗКА ПАКЕТОВ R В SSMLS

В главе 10 вы научитесь работать с моделями машинного обучения посредством служб *SQL Server Machine Learning Services 2019* с использованием языка R. И для этого вам понадобится, чтобы необходимые пакеты были загружены в *SSMLS 2019*. Ниже представлен скрипт на языке T-SQL, который вы можете использовать для вывода информации о том, какие пакеты в данный момент загружены в ваш экземпляр *SSMLS 2019*:

```
EXECUTE sp_execute_external_script
    @language=N'R',
    @script = N'
packagematrix <- installed.packages();
Name <- packagematrix[,1];
Version <- packagematrix[,3];
OutputDataSet <- data.frame(Name, Version);'

WITH RESULT SETS ((PackageName nvarchar(250), PackageVersion nvarchar(max) ))
```

Если пакета, который вам нужен, нет в списке, вам необходимо будет загрузить его вручную. Для этого нужно выполнить следующую пошаговую инструкцию.

## Шаг 1. Загрузите пакет *sqlmlutils* в папку *Documents*

Пакет *sqlmlutils* можно загрузить по адресу <https://github.com/Microsoft/sqlmlutils/tree/master/R/dist>.

Скачайте файл zip с репозитория GitHub и сохраните в папке *Documents*.

## Шаг 2. Запустите следующий код из командной строки

Откройте командную строку под администратором и запустите следующий код на выполнение:

```
R -e "install.packages('RODBCext', repos='https://cran.microsoft.com')"  
R CMD INSTALL %UserProfile%\Documents\sqlmlutils_0.7.1.zip
```

Этот код сработает, если вы предварительно положили архив *sqlmlutils* в папку *Documents* под вашим профилем. Если файл располагается в другом месте, вам необходимо откорректировать путь.

## Шаг 3. Загрузите необходимые пакеты

После выполнения второго шага инструкции вы будете готовы к загрузке пакетов в *SSMLS 2019* из скрипта R в R Studio. Ниже приведем фрагмент кода для загрузки пакета *dplyr* в *SSMLS 2019*:

```
library(sqlmlutils)  
connection <- connectionInfo(  
  server = "server",  
  database = "database",  
  uid = "username",  
  pwd = "password")  
  
sql_install.packages(connectionString = connection,  
  pkgs = "dplyr", verbose = TRUE, scope = "PUBLIC")
```

Вы можете загружать несколько пакетов одновременно. Скажем, вам необходимо загрузить пакеты *dplyr* и *data.table*. Это можно сделать путем создания символьного вектора, содержащего оба пакета, и передачи его параметру *pkgs*, как показано ниже:

```
library(sqlmlutils)  
connection <- connectionInfo(  
  server = "<server>",  
  database = "<database>",  
  uid = "<username>",  
  pwd = "<password>")  
  
pkgList <- c("dplyr", "data.table")  
sql_install.packages(connectionString = connection,  
  pkgs = pkgList, verbose = TRUE, scope = "PUBLIC")
```

## ЗАГРУЗКА НЕОБХОДИМЫХ БИБЛИОТЕК PYTHON В SSMLS

Как и в случае с загрузкой пакетов R в SQL Server Machine Learning Services 2019, вы должны знать, как установить необходимые пакеты Python при помощи *sqlmlutils* для чтения заключительных глав этой книги. Для этого вам нужно выполнить следующие шаги.

### **Шаг 1. Скачайте пакет *sqlmlutils* на свой компьютер в папку *Documents***

Загрузить пакет *sqlmlutils* можно по ссылке <https://github.com/Microsoft/sqlmlutils/tree/master/Python/dist>.

Скачайте архив zip и сохраните его в папке *Documents*.

### **Шаг 2. Откройте командную строку и введите следующие инструкции**

```
pip install "pymssql<3.0"
pip install --upgrade --upgrade-strategy only-if-needed c:\temp\sqlmlutils-0.7.2.zip
```

### **Шаг 3. Загрузите необходимые пакеты**

После выполнения шага 2 вы сможете загружать нужные вам пакеты в *SSMLS 2019* посредством запуска скрипта на Python в *VS Code*. Ниже представлен фрагмент кода, позволяющий загрузить библиотеку *pandas* в *SSMLS 2019*:

```
import sqlmlutils
connection = sqlmlutils.ConnectionInfo(
    server="<имя сервера>", database="<база данных>",
    uid="<имя пользователя>", pwd="<пароль>")
sqlmlutils.SQLPackageManager(connection).install("pandas")
```

## ЛОКАЛЬНЫЙ ШЛЮЗ ДАННЫХ

*Локальный шлюз данных* (on-premises data gateway) представляет собой инструмент для обеспечения безопасной передачи данных между локальными источниками данных и облаком Azure. Локальный шлюз может быть запущен в двух режимах: персональном и стандартном. Если вы захотите развернуть решения, рассматриваемые в главах с третьей по девятую, в службе Power BI и запускать их с определенной периодичностью, вам понадобится установить локальный шлюз данных в персональном режиме. На момент написания книги скрипты на R и Python, используемые в Power Query, могут быть запущены через локальный шлюз только в персональном режиме. Недостатком

использования шлюза в таком режиме является то, что ваше решение нельзя будет назвать корпоративным, поскольку доступ к нему будет, как ясно из названия режима, только у вас.

Но в главе 10 вы узнаете, как можно использовать R и Python в корпоративном решении на базе Power BI через *SQL Server Machine Learning Services (SSMLS)*. При использовании служб *SSMLS 2019* ваш код на R и Python будет заключен в специальную хранимую процедуру на языке T-SQL. А хранимые процедуры допустимо использовать в стандартном режиме локального шлюза. В главе 10 вы также познакомитесь с основами рефакторинга кода на R и Python, описанного в предыдущих главах, под специальные хранимые процедуры, используемые в службах *SSMLS 2019*. Заметьте, что визуальные элементы R не полагаются на локальный шлюз, поскольку они обрабатываются при помощи экземпляра R непосредственно в службе Power BI.

## Источники информации

В данной книге описываются разнообразные технологии, и было бы невозможно досконально рассказать о них всех. Понимая это, я решил предоставить вам наиболее полный список литературы для самостоятельного изучения тем, которые вам пока неизвестны. Также я включил ссылку на репозиторий с исходными кодами, используемыми в книге, и дополнил список ссылками на полезные ресурсы.

## Репозиторий книги

Исходный код для всех упражнений из книги собран в едином репозитории по адресу <https://github.com/Apress/adv-analytics-in-power-bi-w-r-and-python>. Полные скрипты на R и Python сгруппированы в хранилище по главам и темам. В репозитории также представлены актуальные источники данных, используемые в примерах, или информация о получении доступа к ним.

## Ресурсы по R

Книги:

- *R for Data Science*: прекрасная книга, в которой о языке R рассказывает один из самых плодовитых создателей пакетов Хэдли Уикхэм (Hadley Wickham). В ней в том числе описываются полезные пакеты из авторской библиотеки *tidyverse*. Книга доступна бесплатно по адресу <https://r4ds.had.co.nz>;
- *An Introduction to Statistical Learning: With Applications in R*: эта книга описывает базовые принципы статистики, без которых не обойтись при изучении машинного обучения с R. Книге уже несколько лет, но она не утратила популярности в сообществе языка R. Часто материалы

из этой книги используются в качестве учебных пособий для курсов во многих колледжах и университетах.

Веб-сайты:

- *RStudio*: на данном портале собраны разнообразные обучающие ресурсы по языку R. Для получения доступа к ним перейдите в меню на вкладку **Resources**, после чего вам будет предложен выбор бесплатных вебинаров, инструкций и книг. Также на этом сайте вы можете скачать последнюю версию рекомендованной среды для работы с языком R – *R Studio*. Адрес сайта: <https://rstudio.com>;
- *The R Graph Gallery*: на этом сайте представлены визуальные элементы, созданные при помощи R, с полными исходными кодами. Вы можете использовать эти наработки для собственных идей. Адрес сайта: [www.r-graph-gallery.com](http://www.r-graph-gallery.com);
- *R Bloggers*: прекрасный сайт для общения в сообществе по языку R. Адрес сайта: [www.r-bloggers.com](http://www.r-bloggers.com).

Обучение:

- *dplyr tutorial Part 1*: первая из двух частей обучающего видео по пакету *dplyr* от его автора Хэдли Уикхэма. Материал был записан в 2014 году, но до сих пор не утратил своей актуальности. Ссылка на первую часть видео: [www.youtube.com/watch?v=8SGif63VW6E](http://www.youtube.com/watch?v=8SGif63VW6E);
- *dplyr tutorial Part 2*: вторая часть обучающего видео от Хэдли Уикхэма: [www.youtube.com/watch?v=Ue08LVuk790](http://www.youtube.com/watch?v=Ue08LVuk790).

## Ресурсы по Python

Книги:

- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*: превосходная книга, освещающая применение принципов машинного обучения и искусственного интеллекта к вашим данным с помощью инструментов Python. Книга доступна для покупки на большинстве ресурсов;
- *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*: эта книга написана автором библиотеки *pandas* Уэсом Маккинни (Wes McKinney). *Pandas* по сей день является наиболее популярной библиотекой для работы с данными в Python. Книга доступна для покупки на большинстве ресурсов.

Видеоблоги, подкасты и курсы:

- *Data School*: это канал в YouTube от Кевина Маркхэма (Kevin Markham), на котором публикуется масса обучающих видео на тему анализа данных в Python при помощи библиотек *pandas*, *matplotlib*, *scikit-learn* и др. Кевин прекрасно справляется с задачей донесения до слушателей сложных тем довольно простым и понятным языком. Адрес канала: [www.youtube.com/channel/UCnVzApLJE2ljPZSeQylSEyg](http://www.youtube.com/channel/UCnVzApLJE2ljPZSeQylSEyg);
- *Google's Python Class*: это достаточно старый, но все еще актуальный ресурс введения в Python от Google. В данном обучающем материале прекрасно представлено введение в структуры данных языка и другие

базовые концепции, лежащие в основе любого проекта. Также на сайте присутствует весьма информативный раздел, посвященный регулярным выражениям. Адрес двухдневного курса по Python: <https://developers.google.com/edu/python>;

- *Talk Python to Me*: очень информативный подкаст, в котором обсуждаются разные области применительно к Python, в том числе и анализ данных. Адрес подкаста: <https://talkpython.fm>.

Веб-сайты:

- *PEP 8*: одним из главных преимуществ Python в сравнении с другими языками программирования является его доступность и легкость чтения исходного кода. В отличие от большинства языков, Python жестко регламентирует стиль написания кода для облегчения его восприятия в дальнейшем. И именно стилю программирования на Python посвящен этот великолепный сайт, находящийся по адресу <https://pep8.org>.

## Ресурсы по Power BI

Видеоблоги:

- *Guy in a Cube*: однозначно лучший видеоблог по Power BI. На этом канале YouTube освещаются все важнейшие аспекты Power BI, включая администрирование, моделирование данных и визуализацию. Ссылка на канал: [www.youtube.com/channel/UCFp1vaKzpfvoGai0vE5VJ0w](http://www.youtube.com/channel/UCFp1vaKzpfvoGai0vE5VJ0w).

Веб-сайты:

- *SQLBI*: создатели этого сайта (Марко Руссо и Альберто Феррари) являются очень авторитетными авторами книг по языку запросов DAX применительно к Power BI. На их сайте содержится очень много статей, видеоуроков и обучающих материалов по DAX, а также полезные инструменты вроде *DAX Studio*;
- *Tabular Editor*: Tabular Editor – это инструмент с открытым кодом, который должен быть в арсенале любого серьезного разработчика Power BI. В скором времени он должен быть интегрирован в Power BI. Чтобы лучше изучить этот полезный инструмент, перейдите по ссылке, ведущей на его страницу: <https://tabulareditor.com>.

Книги:

- *The Definitive Guide to DAX* («Подробное руководство по DAX»): эта книга – настоящая библия для тех, кто хочет освоить язык запросов DAX, используемый в Power BI. Книга доступна для покупки в большинстве магазинов (<https://dmkpress.com/catalog/computer/data/978-5-97060-859-3/>);
- *M Is for Data Monkeys*: в этой книге дано введение в функциональный язык программирования M, используемый в инструменте Power Query. Авторы проделали хорошую работу, рассказав обо всех основных принципах и возможностях преобразования данных доступным языком. Эта книга может быть использована как основа для перехода к более сложным ресурсам по языку M. Книга доступна для покупки в большинстве магазинов.

Подкасты:

- *BIFOCAL*: прекрасный подкаст, позволяющий быть в курсе всего, что происходит в мире Power BI. Найти его можно на популярных платформах подкастинга.

## Общие ресурсы

Книги:

- *Data Science for Business*: книга от Фостера Провоста (Foster Provost) и Тома Фосетта (Tom Fawcett), в которой представлены основные принципы науки о данных (data science) строгим языком программиста. Это очень популярная книга в сообществе бизнес-аналитики и может заинтересовать тех, кто ищет способы реализации методов науки о данных в области бизнес-приложений. Книга доступна для покупки во многих магазинах.

Веб-сайты:

- *Data Science Central*: это один из самых популярных сайтов, посвященных науке о данных. Адрес сайта: [www.datasciencecentral.com](http://www.datasciencecentral.com);
- *Kaggle*: сайт изначально задумывался как место для соревнований в области программирования, но сейчас перерос в нечто большее. На этом сайте можно найти большое количество наборов данных, которые удобно использовать при работе с примерами по машинному обучению и искусственному интеллекту. Адрес сайта: [www.kaggle.com](http://www.kaggle.com);
- *ExcelTV*: Microsoft Excel был и в обозримом будущем останется одним из основных инструментов в аналитике данных. Авторы сайта ExcelTV подготовили большое количество обучающих материалов, которые помогут вам стать настоящим профессионалом в Excel. Адрес сайта: <https://excel.tv>.

Видеоблоги и обучающие сайты:

- *Excel on Fire*: Oz – создатель и ведущий этого видеоблога, – пожалуй, самый яркий и необычный преподаватель Excel и Power Query. За последние несколько лет он записал большое количество полезных видео по искусственному преобразованию данных при помощи инструмента Power Query. При этом сами ролики записаны на высшем уровне, а способность ведущего доносить сложные вещи простым и понятным языком просто поражает. Ссылка на канал: [www.youtube.com/user/WalrusCandy/featured](http://www.youtube.com/user/WalrusCandy/featured);
- *Regular Expression Tutorial* от Кори Шафера (Corey Schafer): отличный канал с вводными и продвинутыми уроками по использованию регулярных выражений. Поверьте, после просмотра нескольких видео на этом канале вы не только поймете, что означают все эти мудреные символы в регулярных выражениях, но и поразитесь, насколько полезными они могут быть при решении самых разных задач. Ссылка на видеоблог: [www.youtube.com/watch?v=sa-TUpSx1JA](http://www.youtube.com/watch?v=sa-TUpSx1JA).

Подкасты:

- *Analytics on Fire podcast*: это универсальный источник еженедельных мастер-классов по бизнес-аналитике от ведущих специалистов в дан-

ной области. Настраивайтесь на волну каждую неделю, и вы будете получать наслаждение от этого микса из образования и развлечения. Найти подкаст можно на популярных платформах подкастинга;

- *Data Skeptic*: в этом подкасте вопросы из области науки о данных повышенной сложности преподносятся в простой и легкой для усвоения форме с очень увлекательными примерами. Найти подкаст можно на популярных платформах подкастинга;
- *Freakonomics*: изучить технические аспекты науки о данных – это лишь полдела. Очень важно также развить в себе аналитические способности. И этот подкаст, безусловно, поможет вам в этом. Найти подкаст можно на популярных платформах подкастинга;
- *SQL Data Partners*: это очень познавательный и нескучный подкаст, освещающий вопросы, касающиеся *платформы данных Microsoft* (Microsoft Data Platform). Ведущие подкаста – признанные специалисты в этой области, но при этом они много шутят и смеются. Возможно, они будущие комики. :) Их подкаст можно найти на популярных платформах подкастинга;
- *Storytelling with Data*: этот прекрасный подкаст от Коул Нафлик (Cole Knaflic) посвящен различным техникам визуализации данных. Коул также является автором одноименной книги. Найти ее подкаст можно на популярных платформах подкастинга.

## ОПИСАНИЕ ГЛАВ

- Глава 1 «Грамматика графиков». Вероятно, одним из главных преимуществ языка R над Python является его богатая оснащенность средствами визуализации данных. Ведущим пакетом R в области визуализации является *ggplot2*, базирующийся на концепции, известной как *грамматика графиков* (grammar of graphics). В данной главе мы изучим основы графического пакета *ggplot2* и рассмотрим его применение на практике совместно с Power BI.
- Глава 2 «Создание пользовательских визуализаций на R в Power BI при помощи *ggplot2*». Одно из явных преимуществ пакета *ggplot2* состоит в выразительности, с которой вы можете создавать свои собственные визуализации. В данной главе мы на нескольких примерах продемонстрируем идею выбора типа визуализации на языке R из всех доступных в Power BI при помощи пакета *ggplot2*.
- Глава 3 «Чтение файлов CSV». В этой главе мы рассмотрим концепции применительно к языкам R и Python, позволяющие динамически комбинировать файлы CSV, что с использованием инструмента Power Query было бы достаточно затруднительно.
- Глава 4 «Чтение данных из Microsoft Excel». Здесь мы научимся при помощи R и Python динамически сочетать рабочие листы из нескольких рабочих книг Excel, что в Power Query реализовать бывает непросто.
- Глава 5 «Чтение данных из SQL Server». Из данной главы вы узнаете, как посредством R и Python загружать данные из SQL Server в модель



данных Power BI. Одним из преимуществ такого метода загрузки информации – и мы покажем это в наших примерах – является возможность осуществления логирования.

- Глава 6 «Чтение в модель данных Power BI посредством API». В данной главе вы познакомитесь со способами извлечения данных в Power BI при помощи API с использованием языков R и Python. Посредством Power Query реализовать подобные методы бывает довольно сложно, а иногда просто невозможно.
- Глава 7 «Продвинутые строковые операции и распознавание шаблонов». Из этой главы вы узнаете, как при помощи регулярных выражений в R и Python осуществлять сложные манипуляции со строками. Регулярные выражения – очень мощный инструмент для работы с текстом, и если в R и Python он присутствует по умолчанию, то в Power Query пока нативно не реализован.
- Глава 8 «Вычисляемые столбцы с помощью R и Python». Здесь мы рассмотрим технику создания сложных математических выражений при помощи R и Python. Вы познакомитесь с основами написания математических формул и узнаете, как использовать готовые функции, скрывающие от вас истинную сложность вычислений. В качестве примера мы используем формулу гаверсинуса.
- Глава 9 «Применение методов машинного обучения и искусственного интеллекта к моделям данных Power BI». В девятой главе книги мы обсудим множество тем, включая использование в бизнес-аналитике методов машинного обучения и искусственного интеллекта. Начнем мы с примеров того, как использовать пользовательские модели машинного обучения, реализованные на R и Python, к модели данных Power BI. Затем посмотрим, как можно улучшить модели данных Power BI с применением службы *Microsoft Cognitive Services* без необходимости оформления дорогостоящей подписки на Power BI Premium. При этом мы не будем ограничиваться лишь инструментами от Microsoft и покажем, как можно воспользоваться службой *IBM Watson Natural Language Understanding* для выполнения специфического анализа текста, недоступного в рамках *Microsoft Cognitive Services*.
- Глава 10 «Создание моделей анализа данных и скриптов для обработки информации». В заключительной главе мы посмотрим, как можно воспользоваться языками программирования R и Python в корпоративных решениях, внедренных в Power BI. Показанные методы ориентированы на бесплатные решения, доступные пользователям, у которых уже установлена локальная версия SQL Server версии 2017 и выше.

Теперь, когда все подготовительные мероприятия завершены, вы можете приступить к чтению книги. Инструмент Power BI славится своими богатыми возможностями в области визуализации данных. И начнем мы с того, как можно значительно расширить их при помощи языка R и пакета *ggplot2*.

Часть I

.....

**СОЗДАНИЕ  
ПОЛЬЗОВАТЕЛЬСКОЙ  
ВИЗУАЛИЗАЦИИ  
ПРИ ПОМОЩИ R**

# Глава 1

## Грамматика графиков

Визуализация данных во все времена была одной из важнейших составляющих статистики как науки. Именно посредством визуализаций специалисты в этой области могут делиться с окружающими своими изысканиями и при этом не нагружать их сухими цифрами, а говорить на понятном им языке. С учетом того, что язык программирования R был создан аналитиками и для аналитиков, участники сообщества приложили немало усилий, чтобы разработчики могли создавать богатые визуализации, способные помочь им доходчиво рассказывать свои истории о данных.

Наиболее популярным пакетом визуализации данных в R, безусловно, является *ggplot2*. Он был разработан Хэдли Уикхэмом и базируется на концепции, получившей название *многослойная грамматика графиков* (*layered grammar of graphics*). В рамках этой концепции любой график определяется следующими характеристиками:

- набором данных по умолчанию с привязками переменных к *эстетикам* (*aesthetics*);
- одним или более слоями, состоящими из геометрического объекта, статистической трансформации, настройки позиции, а также в качестве необязательных элементов из набора данных и привязки эстетики;
- одной *шкалой* (*scale*) для каждой привязки эстетики;
- системой координат;
- параметрами фасетирования (*faceting*).

Это определение в полной мере описывает принципы многослойной грамматики графиков. Скоро вы узнаете, как использовать описанную выше концепцию для построения прекрасных визуализаций в Power BI при помощи пакета *ggplot2*.

Несмотря на то что Power BI предоставляет возможности создания визуальных элементов с использованием самых разных библиотек и пакетов Python и R, в данной книге мы главным образом сосредоточимся на применении пакета *ggplot2* и нескольких вспомогательных пакетов. Причина в том, что пакет *ggplot2* находится вне конкуренции в сравнении с другими инструментами из арсенала R и Python. Принципы многослойной грамматики графиков, на которых базируется пакет *ggplot2*, позволяют значительно сократить время от рождения идеи определенной визуализации данных до ее воплощения на практике. Этот пакет обладает богатейшим набором возможностей, так что подавляющее большинство ваших идей в области визуализации может быть легко реализовано при помощи него.

# ПОШАГОВОЕ СОЗДАНИЕ ВИЗУАЛИЗАЦИИ В POWER BI ПРИ ПОМОЩИ R

В данном разделе будет подробно рассказано о процессе создания пользовательского элемента визуализации в Power BI посредством языка R. Следующие действия вам необходимо будет выполнить вне зависимости от того, какой тип визуализации вы собираетесь использовать.

## Шаг 1. Настройте Power BI

Во вводной части книги мы уже говорили о том, как настроить Power BI для работы с R. Убедитесь, что вы выполнили все инструкции и рекомендации по подготовке к работе.

## Шаг 2. Перенесите визуальный элемент R в рабочую область Power BI

На вашей панели выбора элементов визуализации справа есть кнопка с латинской буквой «R», показанная на рис. 1.1, при наведении на которую мышью появляется подсказка **Визуальный элемент скрипта R** (R custom visual). Для создания визуализации при помощи этого элемента необходимо нажать на иконку R, в результате чего будет создана заготовка в рабочей области.

Элемент визуализации R предустановлен в Power BI по умолчанию. В рабочей области вы можете изменить его размер и позицию, как и у любого другого визуального элемента в Power BI.



Рис. 1.1 ❖ Иконка создания визуального элемента R в Power BI

## Шаг 3. Определитесь с набором данных

Как и с любым элементом визуализации, в данном случае вам также необходимо определить источник данных. Вы можете собрать набор данных путем переноса полей из вашей модели данных и необязательных мер на панель **Значения** (Values). Полученный в результате набор данных будет доступен в R в виде объекта с именем *data frame*. Вы можете воспринимать этот объект как обычную таблицу Excel с расширенными возможностями для анализа данных.

Одной из особенностей такого объекта, о которой не стоит забывать при создании визуальных элементов R, является обязательная уникальность

строк. При создании объекта *data frame* R автоматически будет проверять его на уникальность строк, и вам необходимо позаботиться о том, чтобы это условие выполнялось в исходном наборе данных.

## Шаг 4. Спроектируйте визуальный элемент в среде разработки R

Теперь, когда вы определились с источником данных для своей визуализации, можно приступить к ее проектированию в вашей любимой среде разработки. Когда вы настраивали Power BI для работы со скриптами R, на одном из шагов вы, если помните, указывали предпочтительную среду разработки. Это делалось для того, чтобы вы могли создавать свой элемент визуализации в привычном для вас инструменте вроде *R Studio* вместо того, чтобы пользоваться встроенным в Power BI редактором скриптов R (R script editor).

В конечном счете код для вашего визуального элемента должен быть вставлен в редактор скриптов R, показанный на рис. 1.2. Этот редактор открывается в нижней части рабочей области в Power BI при нажатии на кнопку создания визуального элемента скрипта R.

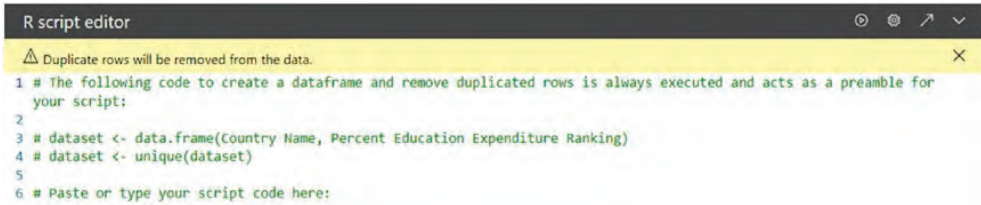


Рис. 1.2 ❖ Редактор скриптов R в Power BI

Как видите, встроенный редактор – не идеальное место для написания скриптов на языке R. Здесь нет *автоматического дополнения ввода* (Intelilicense), нет консоли, а пространство для кода весьма и весьма ограничено. К счастью, Power BI позволяет вам портировать набор данных, который вы определили для своего визуального элемента, в вашу среду разработки. Рекомендованной средой является *R Studio*, и в данной книге мы будем обсуждать работу именно в ней.

Чтобы портировать набор данных для вашего пользовательского визуального элемента в *R Studio*, необходимо выбрать элемент в рабочей области и нажать в заголовке редактора скриптов на стрелку, направленную вверх и направо. Это приведет к запуску оболочки *R Studio* с базовым шаблоном скрипта, включающим в себя код с созданием переменной для *датафрейма* (data frame) с именем *dataset* на основе данных, переданных элементу визуализации в Power BI. Также в оболочку будет перенесен весь код, ассоциированный с вашим визуальным элементом. В общем, в Microsoft решили

не изобретать велосипед, а дать нам возможность разрабатывать свои элементы там, где нам удобно. Лично я рекомендую вам проектировать все свои разработки на языке R именно в *R Studio*, а не пользоваться для этой цели встроенным редактором в Power BI.

## Шаг 5. Используйте следующий шаблон для разработки элемента на R

Представленный ниже шаблон очень удобно использовать при разработке визуального элемента скрипта R в Power BI:

```
if (<"data test">) {  
  #<"Code for R Visual">  
} else {  
  plot.new()  
  title(main = "<predefined message>")  
}
```

В первой строке шаблона выполняется проверка набора данных на соответствие всем требованиям визуализации на языке R. Если все в порядке, визуализация создается успешно, в противном случае будет сгенерирован пустой элемент с заранее определенным текстом. За создание пустого элемента отвечает следующий фрагмент кода:

```
plot.new()  
title(main = " <predefined message> ")
```

В первой строке создается пустая диаграмма, а во второй к ней добавляется заголовок с заранее определенным сообщением, которое будет показано пользователю.

## Шаг 6. Добавьте скрипту функциональности

После написания скрипта для вашего элемента визуализации в сторонней среде разработки перенесите его в редактор скриптов R в Power BI – целиком, за исключением строки с созданием переменной набора данных с именем *dataset*. Этот набор будет автоматически создан в Power BI. Заметьте, что созданный вами элемент визуализации на языке R будет обладать полной интерактивностью и реагировать на изменение фильтров и других визуальных элементов. В следующей главе мы рассмотрим несколько примеров. Единственное, чего вы лишитесь при создании элемента скрипта, – это *двухнаправленной фильтрации* (bidirectional filtering). Таким образом, изменение внешних элементов визуализации в Power BI будет распространять свое действие на ваш элемент R, но фильтровать другие элементы непосредственно из созданного вами не получится.

## РЕКОМЕНДОВАННЫЕ ШАГИ ПО СОЗДАНИЮ ВИЗУАЛЬНОГО ЭЛЕМЕНТА НА R ПРИ ПОМОЩИ GGPLOT2

В предыдущем разделе мы рассмотрели базовый шаблон, с которого стоит начинать разработку собственного элемента визуализации на языке R для Power BI. По сути, он состоит из проверки набора данных на удовлетворение всем необходимым требованиям и перехвата ошибок. В данной секции мы сосредоточимся на фрагменте кода для создания самого элемента, который в представленном листинге обозначен как #<»Code for R Visual»>.

Далее мы рассмотрим, как можно изменить базовый шаблон для построения диаграммы, показанной на рис. 1.3.

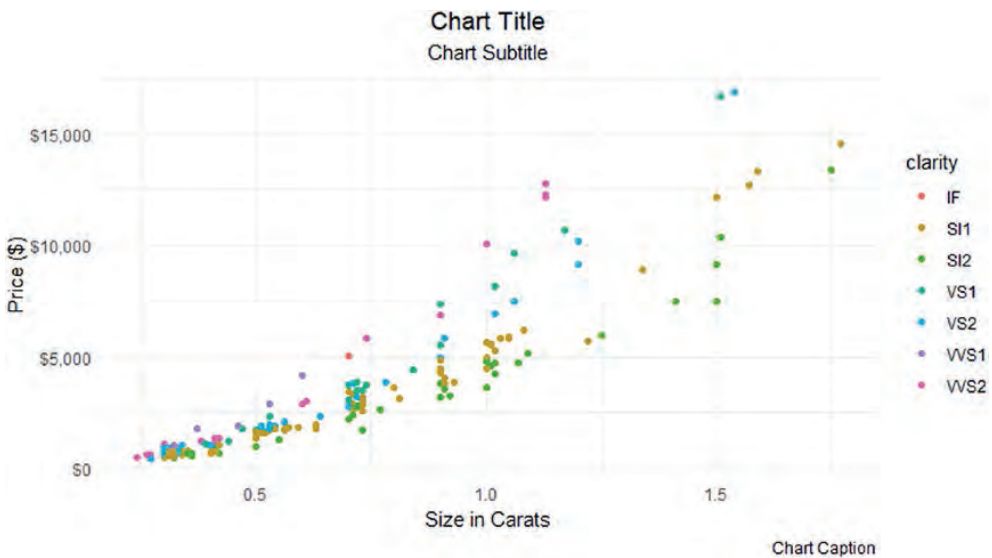


Рис. 1.3 ❖ Визуальный элемент для демонстрации в Power BI

В качестве источника для этой диаграммы используется набор данных об алмазах в части зависимости цены от веса камней в каратах, при этом анализ проводится только по алмазам с цветовой классификацией D. Это довольно подходящий пример для нас, поскольку он позволяет пройти по порядку все шаги при создании элемента визуализации скрипта R. Давайте начнем с самого начала.

### Шаг 1. Импортируйте нужные для скрипта пакеты

Для данного примера мы импортируем два пакета, как показано в представленном ниже листинге:

```
library(tidyverse)
library(scales)
```

Мы будем использовать два пакета *ggplot2* и *dplyr* из набора пакетов *tidyverse*.

**Примечание.** Набор пакетов *tidyverse* объединяет в себе несколько библиотек, значительно облегчающих процесс анализа данных в языке R. Указание на весь набор пакетов (или метапакет) *tidyverse* позволяет загрузить все нужные вам ключевые пакеты одной строкой, вместо того чтобы подгружать каждый пакет по отдельности.

Пакет *ggplot2* мы будем использовать для построения визуализаций, тогда как пакет *dplyr* поможет нам при обработке исходных данных. Загруженный также пакет *scales* поможет нам с форматированием числовых значений.

## Шаг 2. Выполните необходимое преобразование исходных данных

Зачастую, перед тем как «скормить» исходные данные пакету *ggplot2*, требуется привести их в удобный для анализа вид. Один из примеров такой трансформации данных приведен ниже:

```
plot.data <-
  diamonds %>%
  filter(color == "D") %>%
  sample_n(200)
```

Набор данных, переданный на вход элементу визуализации R, помимо полезных показателей содержит информацию, не предназначенную для вывода. Так что мы воспользовались пакетом *dplyr*, чтобы очистить исходные данные от лишнего наполнения. Мы знаем, что наша визуализация должна показывать информацию только по алмазам цветовой классификации D. Таким образом, нам необходимо отфильтровать оригинальный датафрейм *diamonds*, чтобы он включал только нужный нам цветовой диапазон. В листинге выше мы делаем именно это в третьей строке кода.

После фильтрации в наборе данных остается информация только об алмазах нужной нам цветовой классификации. Всего о таких алмазах насчитывается 6775 записей, что довольно много для нашей *диаграммы рассеяния* (scatter plot). Вывод на график чересчур большого количества исследований может привести к тому, что визуализация станет совершенно нечитаемой. Чтобы избежать этого, можно случайным образом выбрать из всей информации какое-то ее подмножество и отобразить только его. Подмножество данных будет обладать тем же распределением, что и исходный набор, что позволит отобразить на графике правильные тенденции и при этом сделать его легким для восприятия. В представленном примере мы воспользовались функцией *sample\_n()* из пакета *dplyr*, оставив в итоговом наборе всего 200 записей, которые вполне приемлемо смотрятся на диаграмме рассеяния.



## Шаг 3. Создайте визуализацию при помощи функции `ggplot()`

Создание диаграммы при использовании пакета `ggplot2` осуществляется путем вызова функции `ggplot()`. На вход функции подаются параметры, которые будут доступны всем без исключения слоям нашей визуализации. В данном случае мы передали в качестве параметра `data` содержимое датафрейма `plot.data`, а значения переменных, хранящих информацию о весе алмазов и их цене, передали аргументам `x` и `y` функции `aes()`, которая также поступила на вход функции `ggplot()`, как показано ниже:

```
ggplot(data = plot.data, aes(x = carat, y = price))
```

Очень важно понимать, что собой представляет функция `aes()` и как ее следует использовать. Функция `aes()` описывает правила привязки ваших данных к визуальным атрибутам, или *эстетикам* (aesthetics). Среди распространенных эстетик можно выделить следующие:

- координаты `x` и `y` на выбранном виде диаграммы;
- цвет выбранных вами геометрических фигур;
- размер геометрических фигур;
- цветовое заполнение фигур на диаграмме.

В нашем простом примере функция `aes()` используется для описания того, какие показатели должны быть отмечены на горизонтальной и вертикальной осях. При этом функция `aes()` может быть объявлена как в функции `ggplot()`, как в нашем случае, так и в функции `geom()`, о которой вы узнаете на следующем шаге. Разница в том, что если функция `aes()` объявлена внутри функции `ggplot()`, ее действие будет распространяться на все слои диаграммы, а если указана в функции `geom()`, действие будет ограничиваться конкретным слоем. В следующей главе мы рассмотрим множество примеров применения функции `aes()` на практике.

## Шаг 4. Добавьте нужные геометрии

Пример добавления *геометрии* (`geom`) к визуализации показан в следующем фрагменте кода:

```
ggplot(plot.data, aes(x = carat, y = price)) +  
geom_point()
```

При инициализации функции `ggplot()` на предыдущем шаге наша визуализация не содержала ни одного слоя, а значит, и визуализировать ничего не могла. Механизм добавления слоев в функции `ggplot()` реализуется через указание соответствующих функций `geom`. Геометрия, которую мы используем для диаграммы рассеяния, добавляется в функцию `ggplot()` посредством вы-

зова функции `geom_point()`. В представленном выше фрагменте кода функция `geom_point()` наследуется от предшествующей ей функции `ggplot()`, поэтому источник данных и координаты  $x$  и  $y$ , обычно обязательные для указания, можно явно не определять. Результирующий график показан на рис. 1.4.

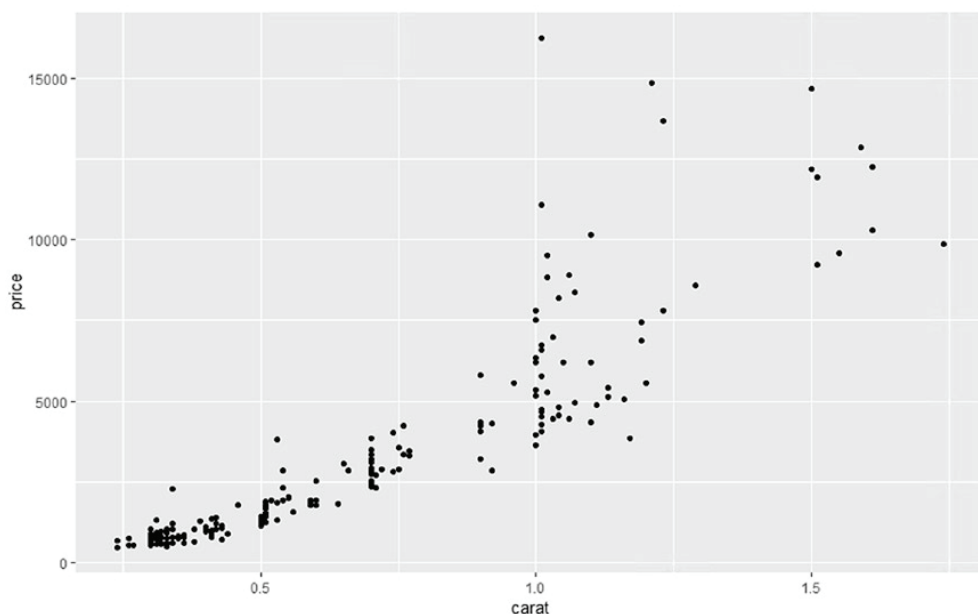


Рис. 1.4 ❖ Базовая диаграмма рассеяния

Теперь нам необходимо добавить еще один слой детализации, а именно мы добавим цветовую дифференциацию по *чистоте* (`clarity`) алмаза. Для этого необходимо присвоить эстетике `color` значение переменной `clarity` в функции `geom_point()`, как показано ниже:

```
ggplot(plot.data, aes(x = carat, y = price)) +
  geom_point(aes(color = clarity))
```

Как видите, мы установили эстетику `color` только в созданной нами геометрии, поскольку она будет использоваться лишь в этом слое.

**Примечание.** Очень важно помнить, что, когда вы устанавливаете источники данных или эстетики в функции `ggplot()`, они будут распространяться на все слои в вашей визуализации, а когда делаете это в конкретной геометрии, их область доступа будет ограничена только ассоциированным с этой геометрией слоем.

Результирующая диаграмма с параметром `clarity` для каждой точки показана на рис. 1.5.

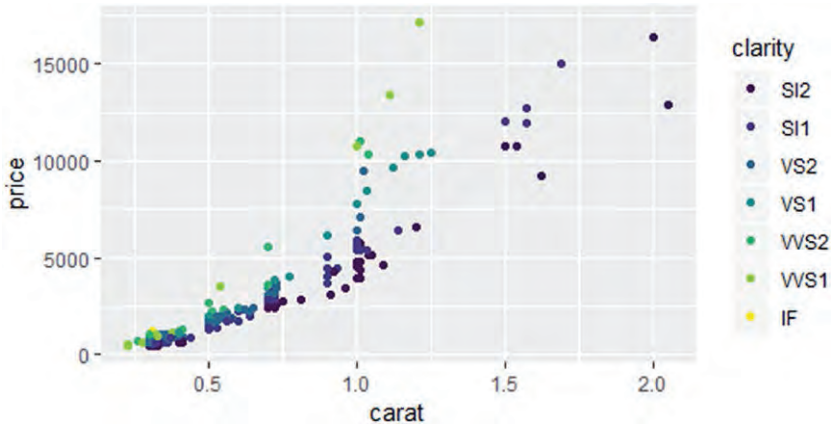


Рис. 1.5 ❖ Базовая диаграмма рассеяния с детализацией

## Шаг 5. Определите заголовки, подзаголовки и подписи

Используя следующий код, можно добавить на визуализацию заголовки:

```
ggplot(plot.data, aes(x = carat, y = price)) +
  geom_point(aes(color = clarity))+
  labs(
    title = "Chart Title",
    subtitle = "Chart Subtitle",
    caption = "Chart Caption"
  )
```

Пакет *ggplot2* позволяет очень легко добавлять на диаграммы заголовки, подзаголовки и подписи при помощи функции *labs()*. Название этой функции сокращено от «labels» (метки). При этом титульные надписи на вашей визуализации могут быть как статическими строками, так и динамическими. В следующей главе мы рассмотрим примеры использования динамических заголовков. На рис. 1.6 показано, как будет выглядеть наша диаграмма с заголовками.

## Шаг 6. Приведите в порядок оси

Зачастую имена, присвоенные осям *x* и *y* по умолчанию при построении диаграммы, нормально подходят и не требуют каких-то изменений. Но если вам необходимо внести правки в названия осей, придется воспользоваться функциями семейства *scale*.

Давайте вспомним весь код, который мы написали на данный момент:

```
library(tidyverse)
library(scales)
```

```
plot.data <-
  diamonds %>%
  filter(color == "D") %>%
  sample_n(200)

ggplot(plot.data, aes(carat, price)) +
  geom_point(aes(color = clarity)) +
  labs(
    title = "Chart Title",
    subtitle = "Chart Subtitle",
    caption = "Chart Caption"
  )
)
```

Запуск этого фрагмента кода приведет к построению диаграммы рассеяния, показанной на рис. 1.6. Как видите, на ней осталось несколько надписей, вставленных по умолчанию, которые нам хотелось бы изменить. Список наших будущих изменений:

- название оси x;
- название оси y;
- способ форматирования чисел на оси y.

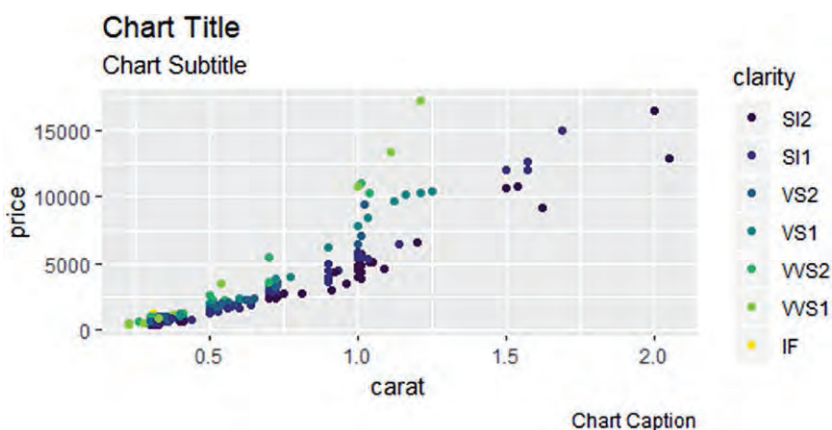


Рис. 1.6 ❖ Визуализация с установленными заголовками и подписями

Эти изменения можно внести, воспользовавшись функциями *scale\_x\_continuous()* и *scale\_y\_continuous()*, как показано ниже:

```
library(tidyverse)
library(scales)

plot.data <-
  diamonds %>%
  filter(color == "D") %>%
  sample_n(200)

ggplot(plot.data, aes(x = carat, y = price)) +
  geom_point(aes(color = clarity)) +
```

```
labs(
  title = "Chart Title",
  subtitle = "Chart Subtitle",
  caption = "Chart Caption"
) +
scale_x_continuous(name = "Size in Carats") +
scale_y_continuous(
  name = "Price ($)",
  labels = dollar_format()
)
```

Получившаяся в результате запуска этого кода диаграмма показана на рис. 1.7.

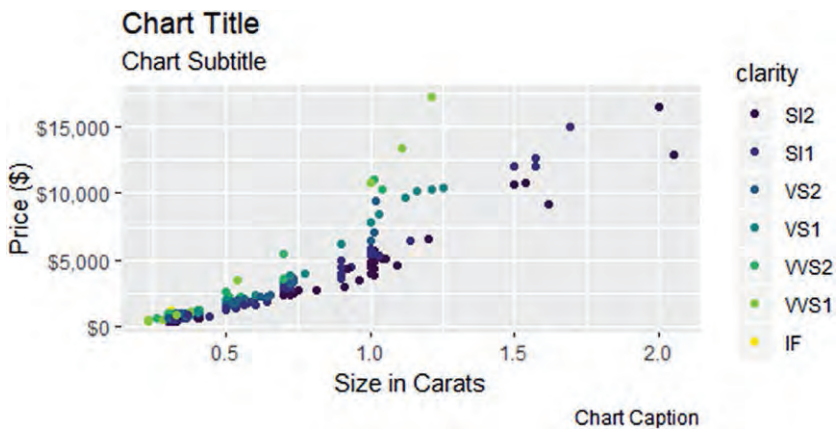


Рис. 1.7 ❖ Диаграмма с настроенными осями

Как видите, при помощи функций `scale_x_continuous()` и `scale_y_continuous()` можно достаточно быстро внести необходимые нам корректировки.

**Примечание.** Заметьте, что вы добавили шкалы на график так же точно, как до этого добавляли слои. Здесь важно понять, что добавленные шкалы заменяют собой шкалы по умолчанию, но не добавляют к вашей диаграмме слои.

Поскольку оси  $x$  и  $y$  базируются на *непрерывных данных* (continuous data), для них будет использоваться именно непрерывная шкала. Единственное, что нам необходимо изменить в отношении оси  $x$ , – это ее подпись, что, как видите, легко делается путем передачи аргумента `name`. Для оси  $y$  мы изменим не только подпись, но и способ форматирования числовых меток на оси. Подпись также меняется при помощи аргумента `name`, а для меток предусмотрен свой аргумент с названием `labels`. Мы передали этому аргументу функцию `dollar_format()` из пакета `scales`, что привело к форматированию меток на оси в виде долларов. Функция `dollar_format()` наследует параметр `u` от функции `ggplot()`, которая была определена в начале кода, так что явно объявлять его не нужно.

## Шаг 7. Примените тему при необходимости

В пакете *ggplot2* есть предустановленные *темы* (themes), отвечающие за оформление диаграмм в той части, где нет данных. Сюда относятся и расположение подписей и заголовков на графике, и цвет фона, и размер шрифтов для заголовков, и многое другое. Применить одну из предустановленных тем к вашему визуальному элементу довольно просто. Темы добавляются так же точно, как устанавливаются слои и модифицируются шкалы. Вы просто ставите знак + (плюс), следом за которым указываете название функции, символизирующей выбранную вами тему, как показано во фрагменте кода ниже:

```
ggplot(plot.data, aes(x = carat, y = price)) +
  geom_point(aes(color = clarity)) +
  labs(
    title = "Chart Title",
    subtitle = "Chart Subtitle",
    caption = "Chart Caption"
  ) +
  scale_x_continuous(name = "Size in Carats") +
  scale_y_continuous(
    name = "Price ($)",
    labels = dollar_format()
  ) +
  theme_minimal()
```

Результат показан на рис. 1.8.

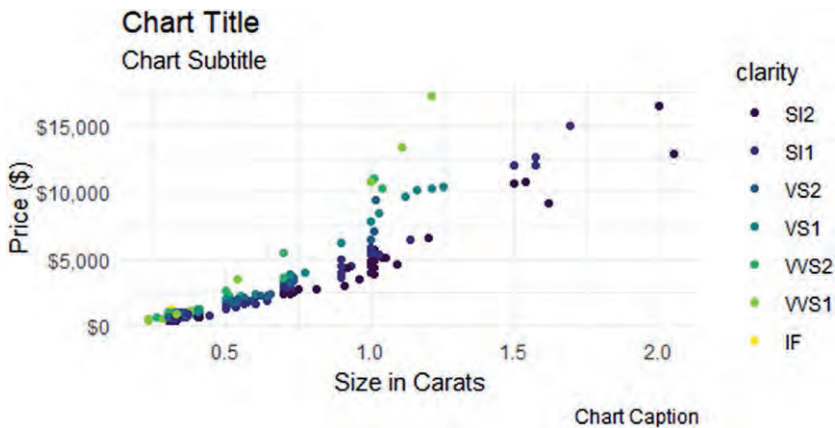


Рис. 1.8 ❖ Установка темы для визуализации

В данном примере мы выбрали предустановленную тему *theme\_minimal()*. Считается хорошим тоном снабжать графики дополнительными данными только в случае особой необходимости. Белый фон под диаграммой также выглядит довольно приятно. Тема *theme\_minimal()*, как ясно из названия, базируется на основных принципах минимализма. Подробно о темах, предустановленных в пакете *ggplot2*, мы поговорим далее в этой главе.

## Шаг 8. Используйте функцию `theme()` для настройки оформления

Если вы остались не полностью удовлетворены примененной к визуализации предустановленной темой, то можете провести ее тонкую настройку при помощи специальной функции `theme()`. В следующем фрагменте кода мы вызовем эту функцию с целью форматирования заголовков:

```
ggplot(plot.data, aes(carat, price)) +
  geom_point(aes(color = clarity)) +
  labs(
    title = "Chart Title",
    subtitle = "Chart Subtitle",
    caption = "Chart Caption"
  ) +
  scale_x_continuous(name = "Size in Carats") +
  scale_y_continuous(
    name = "Price ($)",
    labels = dollar_format()
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    plot.caption = element_text(hjust = 1)
  )
)
```

Результат запуска этого кода показан на рис. 1.9.

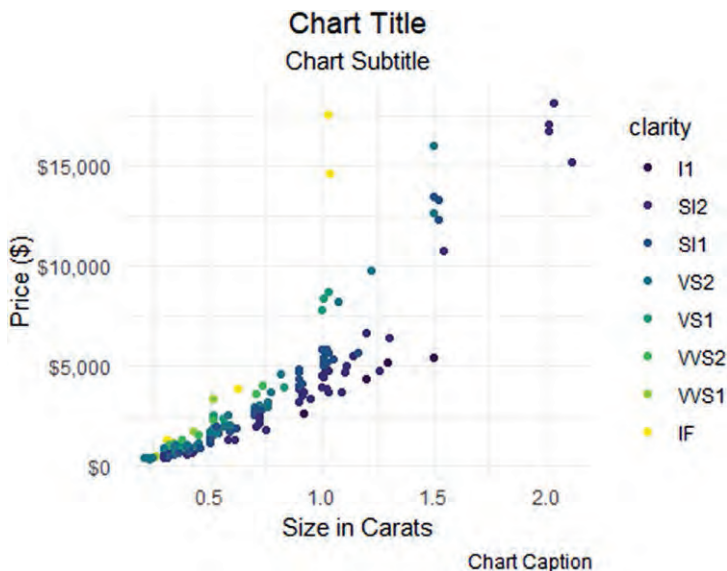


Рис. 1.9 ❖ Смещение заголовков и подписей с помощью функции `theme()`

Функция `theme()` открывает вам доступ к форматированию более чем 80 различных элементов на графике. Для большинства этих элементов существуют функции семейства `element`, предназначенные специально для выполнения подобных изменений.

В представленном выше фрагменте кода мы изменили положение заголовка диаграммы, подзаголовка и подписи. Как видите, элементу, ассоциированному с заголовком графика, соответствует объект `plot.title`, к подзаголовку мы обращаемся как к `plot.subtitle`, а к нижней подписи – как к `plot.caption`. При этом изменение всех трех элементов выполняется при помощи функции `element_text()`. Заголовок и подзаголовок графика были центрированы путем передачи аргументу `hjust` функции `element_text()`, отвечающему за горизонтальное позиционирование элемента, значения 0.5. Этот аргумент может принимать значения в диапазоне от 0 до 1, где 0 означает расположение элемента по левому краю, 0.5 – по центру, а 1 – по правому краю.

Ниже представлены функции семейства `element` со всеми своими аргументами:

```
element_blank()
```

```
element_rect(fill = NULL, colour = NULL, size = NULL, linetype = NULL, color = NULL,
inherit.blank = FALSE)
```

```
element_line(colour = NULL, size = NULL, linetype = NULL, lineend = NULL, color = NULL,
arrow = NULL, inherit.blank = FALSE)
```

```
element_text(family = NULL, face = NULL, colour = NULL, size = NULL, hjust = NULL, vjust =
NULL, angle = NULL, lineheight = NULL, color = NULL, margin = NULL, debug = NULL, inherit.
blank = FALSE)
```

По адресу <https://ggplot2.tidyverse.org/reference/element.html> можно найти полную документацию по функции `theme()`.

## Дополнительный шаг: задайте цвета точек на диаграмме рассеяния

Иногда вам будет необходимо, чтобы все точки на диаграмме рассеяния были одного цвета. Давайте посмотрим, что будет, если присвоить эстетике `color` конкретный цвет, а не привязывать ее к определенному полю в наборе данных. Во фрагменте кода ниже мы присвоим значение `blue` этой эстетике в функции `aes()`:

```
library(tidyverse)
library(scales)
```

```
plot.data <-
  diamonds %>%
  filter(color == "D") %>%
  sample_n(200)
```



```

ggplot(plot.data, aes(carat, price)) +
  geom_point(aes(color = "blue")) +
  labs(
    title = "Chart Title",
    subtitle = "Chart Subtitle",
    caption = "Chart Caption"
  ) +
  scale_x_continuous(name = "Size in Carats") +
  scale_y_continuous(
    name = "Price ($)",
    labels = dollar_format()
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    plot.caption = element_text(hjust = 1)
  )

```

Результирующая диаграмма показана на рис. 1.10.

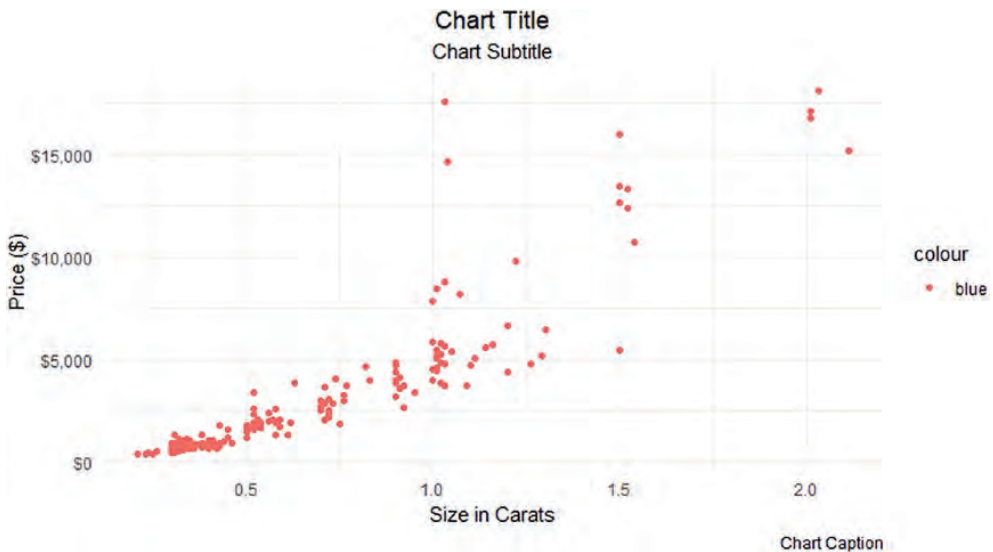


Рис. 1.10 ❖ Установка эстетики *color* в функции *aes()*

Заметьте, что мы не добились желаемого результата. Точки на нашей диаграмме рассеяния получились не синего цвета, как мы планировали, а красного. Причина в том, что мы попытались присвоить цвет точкам внутри функции *aes()*. Если вы хотите присвоить цвету конкретное значение, делать это необходимо за пределами функции *aes()*, как показано в исправленном листинге ниже:

```

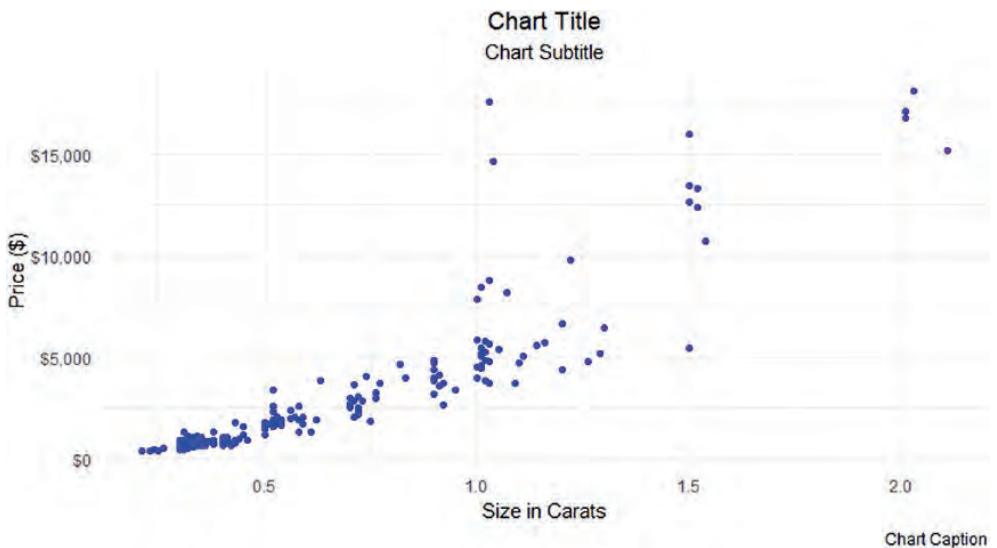
library(tidyverse)
library(scales)

plot.data <-
  diamonds %>%
  filter(color == "D") %>%
  sample_n(200)

ggplot(plot.data, aes(carat, price)) +
  geom_point(color = "blue") +
  labs(
    title = "Chart Title",
    subtitle = "Chart Subtitle",
    caption = "Chart Caption"
  ) +
  scale_x_continuous(name = "Size in Carats") +
  scale_y_continuous(
    name = "Price ($)",
    labels = dollar_format()
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    plot.caption = element_text(hjust = 1)
  )

```

Теперь мы получим ожидаемый результат с синими точками по диаграмме рассеяния, как видно на рис. 1.11.



**Рис. 1.11** ❖ Определение цвета точек за пределами функции *aes()*