

Организация рабочего процесса: скрипты

В предыдущих главах вы выполняли код в окне консоли. Для начала это было удобно, но как только вы приступите к работе с более сложными диаграммами **ggplot2** и каналами **dplyr**, то вскоре убедитесь, что это доставляет большие неудобства. Использование окна редактора скриптов (сценариев) позволяет значительно расширить размеры доступного рабочего пространства. Чтобы открыть редактор скриптов, щелкните на пункте меню File (Файл), а затем последовательно выберите пункты New File (Создать файл) и R script (R-скрипт), либо воспользуйтесь комбинацией клавиш <Ctrl+Shift+N>. Теперь в вашем распоряжении будут четыре панели (рис. 4.1).

Редактор скриптов предоставляет великолепные возможности для работы с интересующим вас кодом. Продолжайте использовать консоль для того, чтобы экспериментировать с кодом, но как только вы получите вариант кода, который делает то, что вам нужно, переместите его в редактор скриптов. RStudio автоматически сохраняет его содержимое при завершении рабочего сеанса и автоматически загружает его при повторном открытии RStudio. Тем не менее будет неплохо, если вы выработаете привычку регулярно сохранять свои скрипты и создавать их резервные копии.

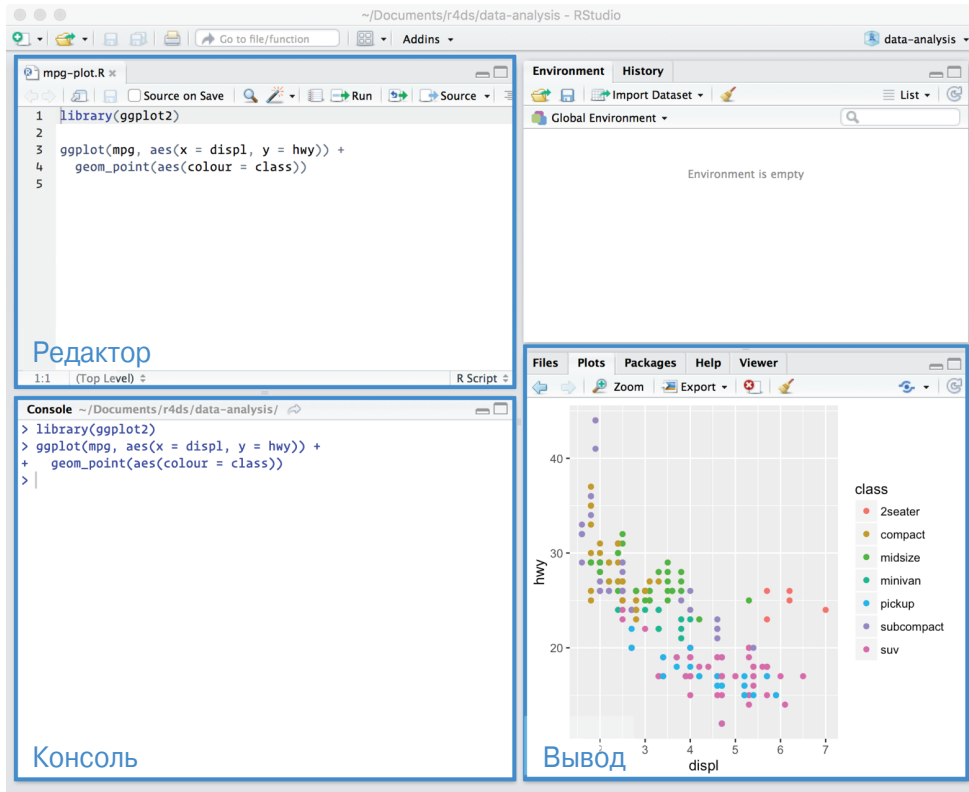


Рис. 4.1

Выполнение кода

Редактор скриптов отлично подходит для создания сложных диаграмм пакета **ggplot2** или длинных последовательностей команд пакета **dplyr**. Для эффективного использования редактора скриптов вам нужно запомнить одну из наиболее важных комбинаций клавиш: **<Ctrl+Enter>**. Она обеспечивает выполнение в окне консоли текущего R-выражения. Возьмем, например, приведенный ниже код. Если курсор находится в позиции █, нажатие клавиш **<Ctrl+Enter>** приведет к запуску полной команды **not_cancelled**. При этом курсор переместится к следующей инструкции (начинающейся с выражения **not_cancelled %>%**). Эта особенность упрощает выполнение всего сценария путем повторного нажатия комбинации клавиш **<Ctrl+Enter>**.

```
library(dplyr)
library(nycflights13)
```

```
not_cancelled <- flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay))

not_cancelled %>%
  group_by(year, month, day) %>%
  summarize(mean = mean(dep_delay))
```

Вместо того чтобы последовательно выполнять одну инструкцию за другой, можно выполнить весь сценарий за один раз. Для этого следует нажать комбинацию клавиш <Ctrl+Shift+S>. Это отличный способ контроля того, что в сценарий включены все важные компоненты вашего кода.

Мы рекомендуем всегда начинать свой сценарий с указания необходимых пакетов. Благодаря этому люди, с которыми вы поделитесь своим кодом, сразу же увидят, какие пакеты им необходимо установить. Однако никогда не включайте в код совместно используемого скрипта инструкции наподобие `install.packages()` или `setwd()`. Принудительное изменение настроек на чужом компьютере недопустимо!

Я настоятельно рекомендую, чтобы работу с кодом, приведенным в последующих главах, вы всегда начинали в окне редактора скриптов и практиковались в использовании клавиатурных сокращений. Со временем отправка кода на консоль описанным способом станет для вас настолько естественной, что вы даже не будете об этом задумываться.

Диагностические средства RStudio

Редактор скриптов выделяет синтаксические ошибки волнистой красной линией и помечает соответствующие строки крестиком слева (рис. 4.2).




Рис. 4.2

Наведя указатель мыши на этот крестик, вы получите подсказку относительно возможных причин возникновения ошибки (рис. 4.3).

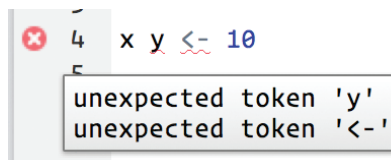


Рис. 4.3

RStudio также информирует вас о наличии потенциальных проблем (рис. 4.4).

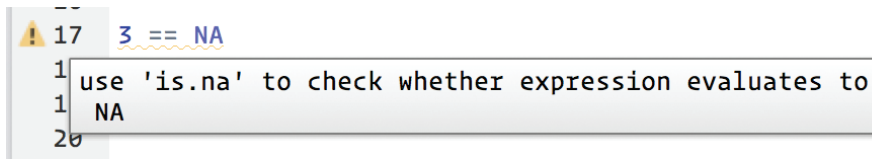


Рис. 4.4

Упражнения

1. Посетите микроблог RStudio Tips в Твиттере ([@rstudiotips](https://twitter.com/rstudiotips)), содержащий советы по использованию RStudio, и выберите тот из них, который вас больше всего заинтересовал. Попрактикуйтесь в его использовании!
2. О каких других распространенных видах ошибок информирует RStudio в своих диагностических сообщениях? Чтобы выяснить это, перейдите по адресу <http://bit.ly/RStudiocodediag>.