Основы программирования в R

Визуализация данных в R: часть 1

Алла Тамбовцева, НИУ ВШЭ

Содержание

Загрузим данные из файла `tree` с учетом кодировки и вида ячеек с пропущенными значениями и удалим строки с `NA`:

```
tree <- read.csv("/Users/allat/Desktop/firtree.csv",
                 encoding = "UTF-8", na.strings = "")
tree <- na.omit(tree)
```

Визуализация качественных данных: столбиковая диаграмма

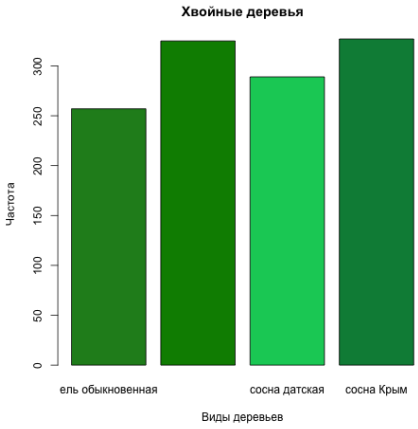
Построим столбиковую диаграмму (*bar plot* или *bar chart*) для столбца `ftype`. Но для начала вспомним, сколько в этом столбце уникальных значений, чтобы представлять, как будет выглядеть график:

```
unique(tree$ftype)
```

```
## [1] пихта Нобилис    сосна Крым       ель обыкновенная сосна датская
## Levels: ель обыкновенная пихта Нобилис сосна датская сосна Крым
```

Воспользуемся функцией `barplot()`. Добавим заголовок к графику (аргумент `main`), подписи к горизонтальной и вертикальной оси (`xlab` и `ylab`) и вектор цветов для столбцов (`col`):

```
barplot(table(tree$ftype),
        main = "Хвойные деревья",
        col = c("forestgreen", "green4", "springgreen3",
                "springgreen4"),
        xlab = "Виды деревьев", ylab = "Частота")
```



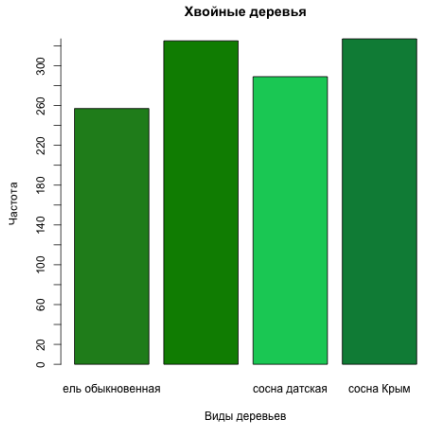
А теперь сделаем ось Oy более детальной – добавим побольше делений. Для ручной корректировки осей есть специальная функция `axis()`, где первым аргументом идёт номер оси (1 – для оси Ox, 2 – для оси Oy), а далее в виде вектора указываются деления на оси. В нашем случае логично сделать деления от 0 до 350 с шагом 20, поэтому нужная строка будет выглядеть так:

```
axis(2, at = seq(from = 0, to = 350, by = 20))
```

Эта функция накладывает на уже построенный график новые оси. Если редактируем оси с помощью `axis()`, важно не забыть убрать оси на исходном графике, иначе новые оси наложатся на старые, и это будет выглядеть странно. Для этого в функции `barplot()` нужно добавить аргумент `axes=FALSE`.

```
barplot(table(tree$ftype),
        main = "Хвойные деревья",
        col = c("forestgreen", "green4", "springgreen3",
                "springgreen4"),
        xlab = "Виды деревьев",
        ylab = "Частота",
        axes = FALSE)
axis(2, at = seq(from = 0, to = 350, by = 20))
```

Внимание: строка с `axis()` должна идти сразу после кода для построения самого графика, иначе R не поймёт, к какому графику она относится, и выдаст ошибку.



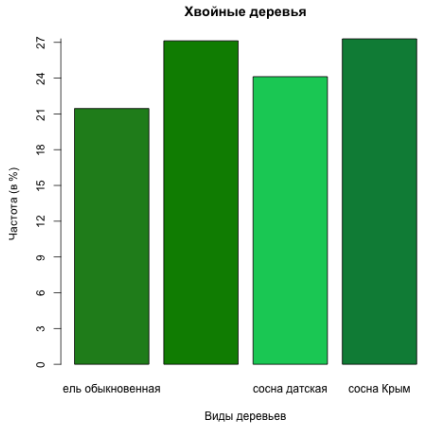
А что если мы захотим получить столбиковую диаграмму, но не с абсолютными частотами (число деревьев), а с относительными частотами в процентах?

Всё просто: чтобы построить такой график, нужно создать таблицу с частотами в процентах – взять таблицу, полученную с помощью `table()`, поделить все частоты в ней на сумму значений и умножить на 100:

```
tab <- table(tree$ftype)/sum(table(tree$ftype)) * 100
```

Проверим:

```
barplot(tab,
        main = "Хвойные деревья",
        col = c("forestgreen", "green4", "springgreen3",
                "springgreen4"),
        xlab = "Виды деревьев",
        ylab = "Частота (в %)",
        axes = FALSE)
axis(2, at = seq(from = 0, to = 33, by = 3))
```



А чтобы было совсем здорово, добавим подписи с процентами на сам график. Для этого сначала создадим вектор с подписями – округлим значения в `tab` и «приклеим» к каждому значению символ `%`.

Округляем:

```
perc <- round(table(tree$ftype)/sum(table(tree$ftype)) * 100, 2)
```

Для «приклеивания» символа `%` воспользуемся функцией `paste()`, она используется для склеивания строк, но умеет склеивать числовые векторы со строками. Посмотрим на отвлечённый пример:

```
paste("group", c(171, 172, 173, 174))
```

```
## [1] "group 171" "group 172" "group 173" "group 174"
```

В примере выше функция `paste()` к слову “group” доклеила номера групп из числового вектора. По умолчанию в качестве разделителя используется пробел.

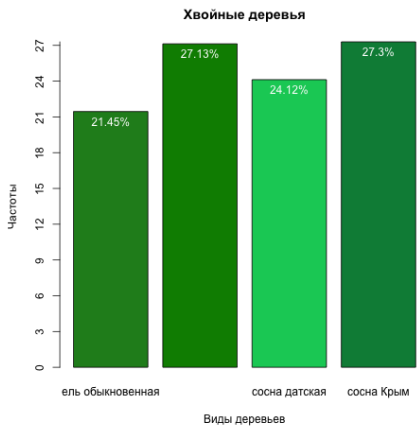
Вернёмся к нашему графику. К каждому элементу вектора `perc` нам нужно приклеить `%`, причём без пробела и иных символов.

```
# sep - separator
perc_labels <- paste(perc, "%", sep="")
```

Теперь нанесем всё на график – добавим нужный нам текст. Это можно сделать с помощью функции `text()`, только сначала нужно сохранить график в переменную. Сохраним его в переменную `p`, а потом укажем её в качестве аргумента в функции `text()`:

```
p <- barplot(tab,
        main = "Хвойные деревья",
        col = c("forestgreen", "green4", "springgreen3",
                "springgreen4"),
        xlab = "Виды деревьев",
        ylab = "Частоты",
        axes = FALSE)
axis(2, at = seq(from = 0, to = 50, by = 3))
text(x = p, y = perc, labels = perc_labels,
     pos = 1, col = "white", face = 2)
```

Здесь в `text()` в качестве координат наносимого текста по оси `x` мы используем значения из самого графика `p`, в качестве координат текста по оси `y` используем значения процентов для каждого столбца из `perc`. Затем указываем сам текст (аргумент `labels`) и сообщаем R, что текст должен размещаться над точкой с заданными координатами (`pos=1`) и что цвет текста должен быть белым. Плюс, добавив

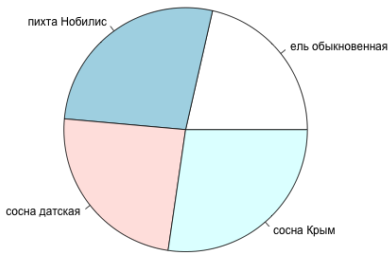


аргумент `face=2`, мы сделали шрифт подписей полужирным.

Визуализация качественных данных: круговая диаграмма

Круговая диаграмма (*pie chart*) строится для относительных частот – для процентов. Таблицу для процентов мы уже построили (`perc`).

```
pie(perc)
```



Пока выглядит грустно. Начинаем исправлять. Создадим и добавим вектор цветов для диаграммы:

```
pie(perc, col = c("forestgreen", "darkolivegreen1", "springgreen3",
                  "lightseagreen"),
    main = "Хвойные деревья")
```

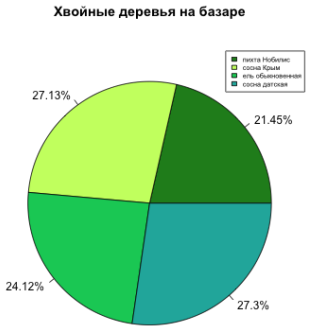
Сильно красивее не стало.

Поэтому сделаем по-другому: на самом графике подпишем проценты (`perc_labels`), а остальное вынесем в легенду графика.

Внимание: как в случаях с `axis()` и `text()` прописываем функцию для легенды `legend()` сразу после построения графика.

```
pie(perc, col = c("forestgreen", "darkolivegreen1", "springgreen3",
                  "lightseagreen"),
    main = "Хвойные деревья",
    labels = perc_labels)

legend(x = 0.5, y = 1, as.character(unique(tree$ftype)),
       fill = c("forestgreen", "darkolivegreen1", "springgreen3",
                  "lightseagreen"),
       cex = 0.6)
```



Так же как и в случае с `text()`, здесь мы сначала указываем координаты для легенды (аргументы `x` и `y`), потом значения, которые должны быть в ней указаны (уникальные значения столбца `ftype`), и цвета, которые им соответствуют на графике (`fill`). Аргумент `cex` отвечает за размер шрифта, который по умолчанию равен 1.

Визуализация количественных данных: гистограмма

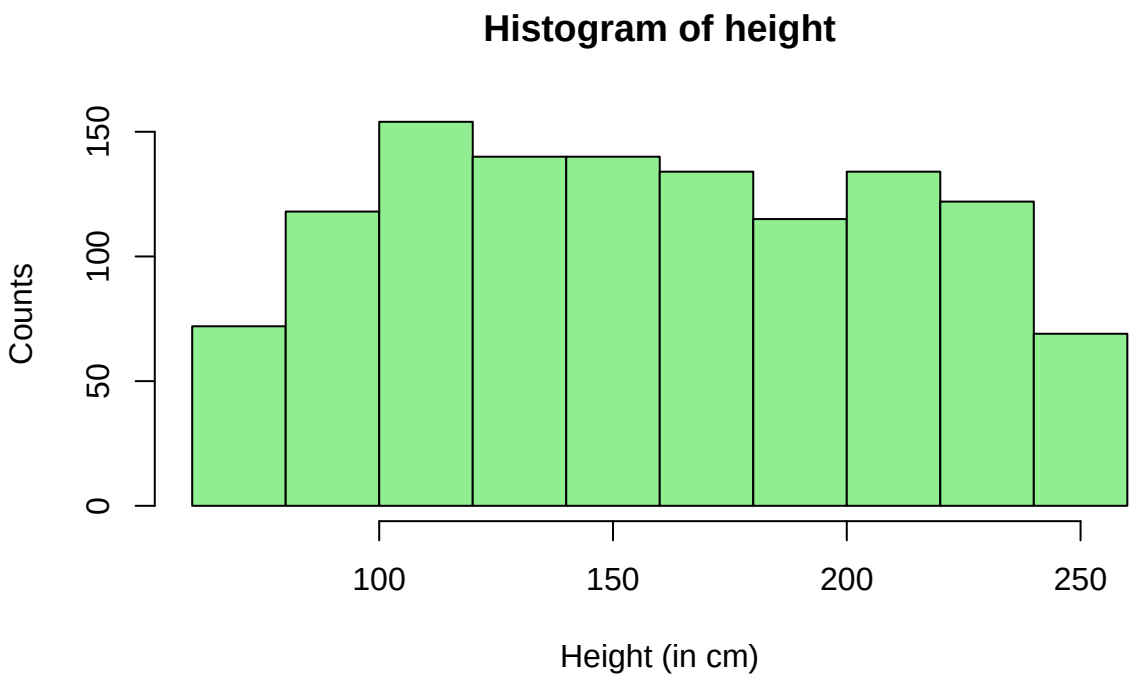
Гистограмма – график, который иллюстрирует соответствие между значениями числового показателя и частотой, с которыми эти значения встречаются в данных.

Для начала построим самую простую гистограмму для высоты деревьев (`height`).

```
hist(tree$height)
```

Теперь сделаем её красивой: добавим заголовок графика, поменяем цвет и подпишем оси. Для разнообразия все подписи сделаем на английском:

```
hist(tree$height,
     col = "lightgreen",
     main = "Histogram of height",
     xlab = "Height (in cm)",
     ylab = "Counts")
```



Что показывает эта гистограмма? Форму распределения высоты деревьев. По гистограмме видно следующее:

- распределение можно считать симметричным относительно среднего значения;
- меньше всего низких деревьев (высота 60-100 см) и высоких (более 240 см);
- если не рассматривать низкие и высокие деревья, распределение похоже на равномерное — шансы встретить деревья высотой 120-140 см примерно такие же, что и встретить деревья высотой 140-160 см, 160-180 см и так далее.

При построении гистограмм можно задействовать важный параметр – `breaks`, число «разбивок», то есть «перегородок», которые нужно взять, чтобы поделить упорядоченную выборку на равные интервалы. Нетрудно догадаться, что число самих равных интервалов всегда будет на 1 меньше, чем число «перегородок».

Пример:

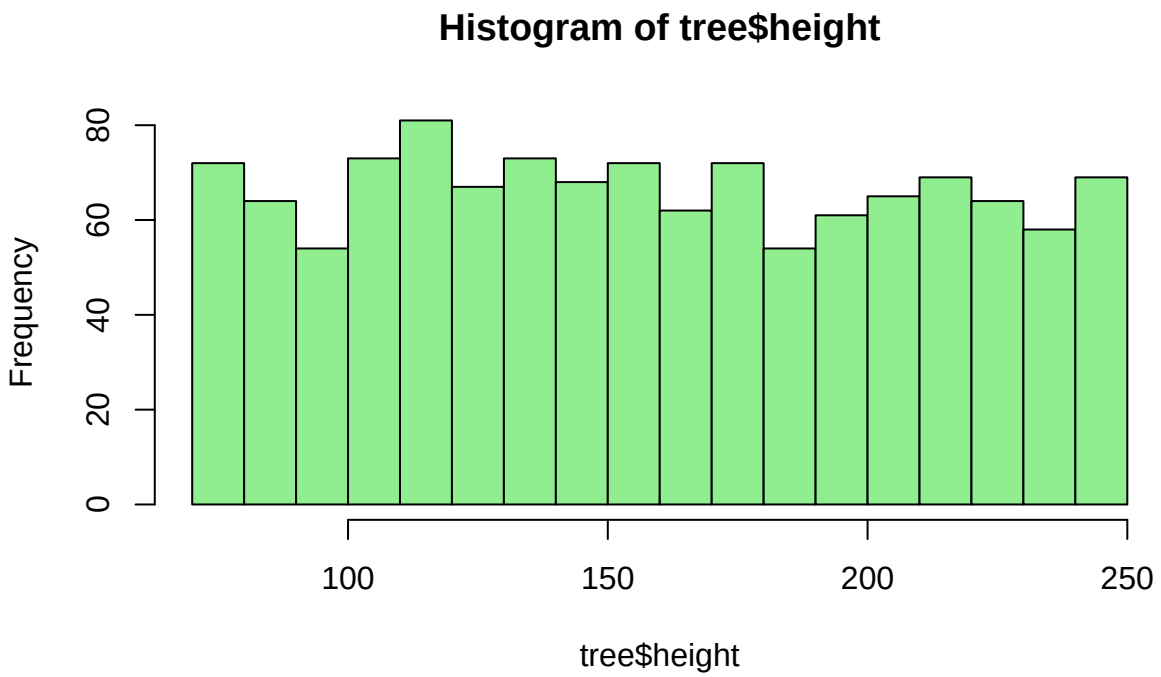
|2 2 | 3 5 | 6 8 | 9 10|

breaks = 5

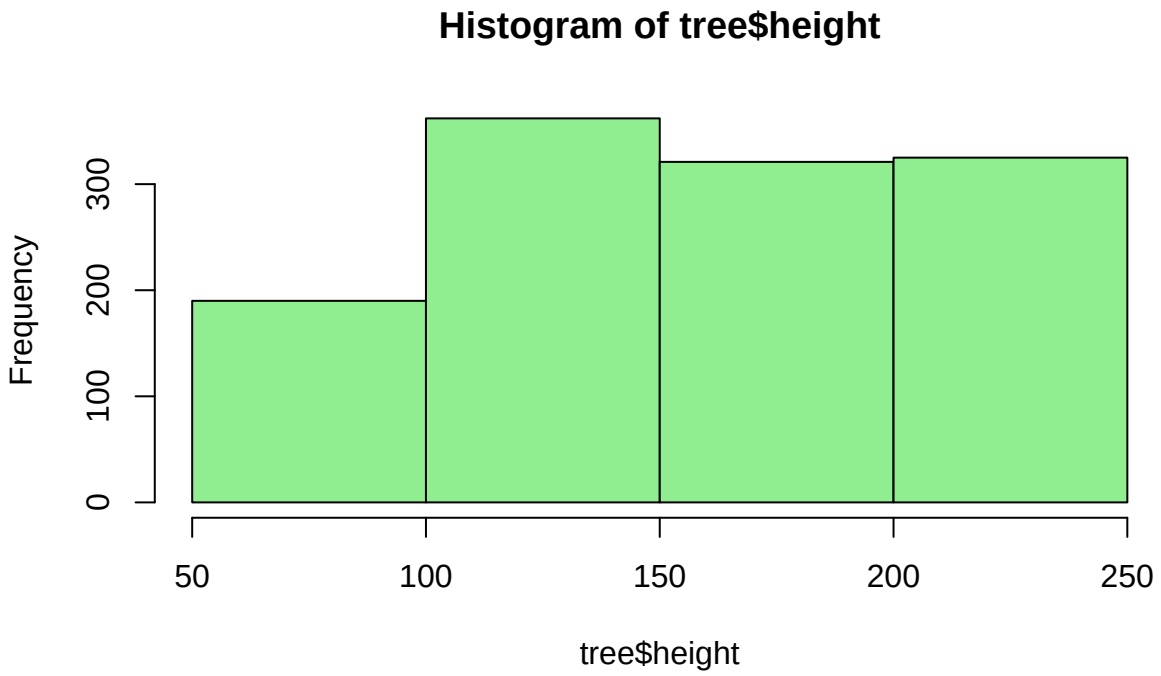
число интервалов = 4 (число столбцов в гистограмме)

Сравним две гистограммы:

```
hist(tree$height,
     col = "lightgreen",
     breaks = 20)
```




```
hist(tree$height,
     col = "lightgreen",
     breaks = 5)
```

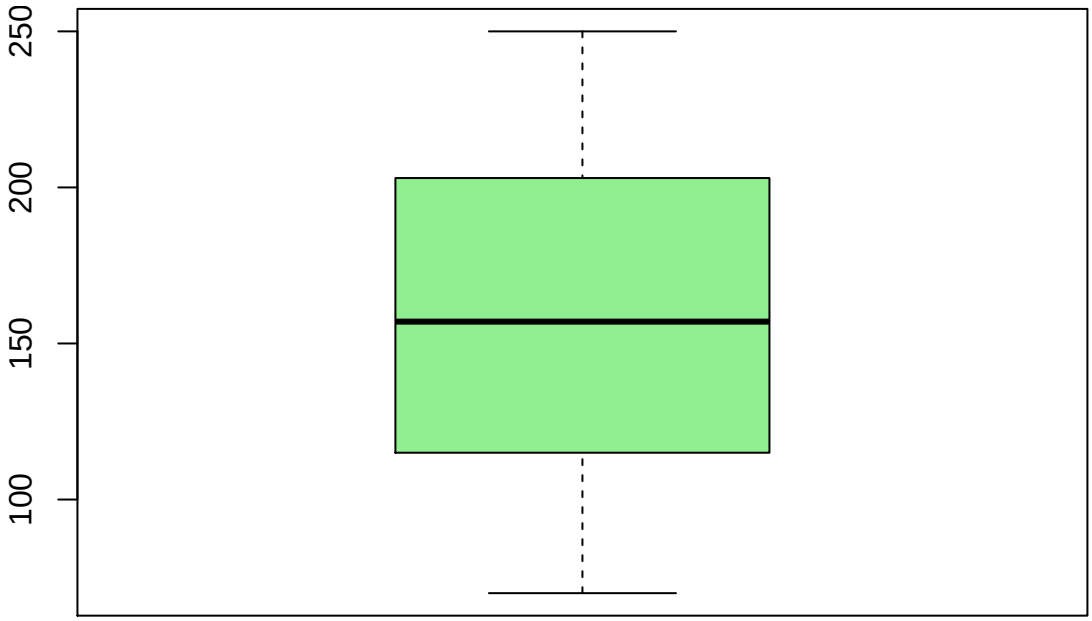


Гистограммы внешне отличаются друг от друга. Но, на самом деле, это одна и та же гистограмма, просто с разными интервалами разбивки. На первой гистограмме ширина столбца равна 10, а на второй – 50. Как правило, вопрос о том, какое число столбцов гистограммы взять, не возникает, потому что R автоматически подбирает подходящее число столбцов. (По умолчанию используется алгоритм Стёржеса, кому интересно почитать про различные алгоритмы, см. здесь, *Number of bins and width*). Да, это Википедия, но там есть ссылки на оригинальные статьи про способы разбивки.

Визуализация количественных данных: ящик с усами

С помощью ящика с усами также можно представить распределение количественного показателя. В отличие от гистограммы, ящик с усами позволяет явно увидеть описательные статистики, посчитанные по выборке. Подробнее см. здесь.

```
boxplot(tree$height, col = "lightgreen")
```



Ящик с усами позволяет также определить, есть ли в выборке нетипичные значения (выбросы, *outliers*), то есть значения, которые сильно отличаются от остальных и располагаются за пределами «усов» графика. В нашем случае таких наблюдений нет. К сожалению, с помощью стандартной функции `boxplot()` мы не сможем отметить на графике, что это за регионы. Но к более «продвинутым» ящикам с усами мы еще вернёмся, когда будем работать с библиотекой ggplot2.

Визуализация количественных данных: скрипичная диаграмма

Еще один тип графика, который используется для визуализаии распределения количественных данных, это скрипичная диаграмма (*violin plot* или *bean plot* или *vase plot*). Чтобы понять, почему у графика такие специфические названия, давайте его построим.

Установим сначала библиотеку `vioplot`.

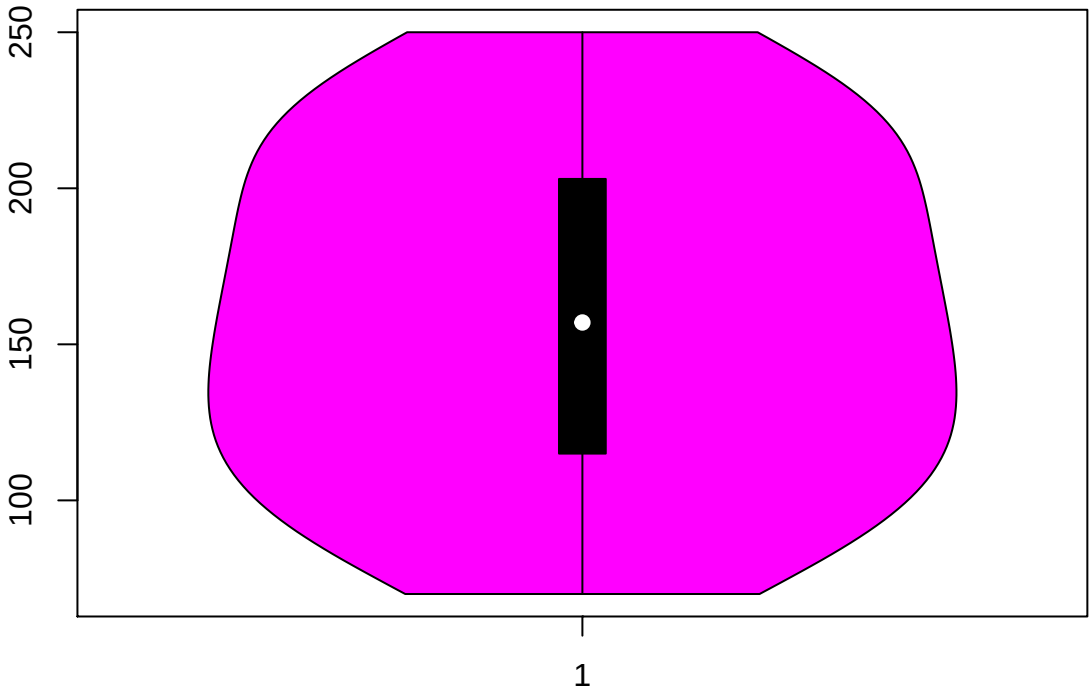
```
install.packages("vioplot")
```

И обратимся к ней.

```
library(vioplot)
```

Построим диаграмму для `height`:

```
vioplot(tree$height)
```



Получился питон, который проглотил ящик с усами! На самом деле, скрипичная диаграмма объединяет два графика: ящик с усами и график плотности распределения (сглаженный вариант гистограммы – если повернем график на 90 градусов, увидим, что он состоит из двух одинаковых половинок, похожих на обведенную по контуру гистограмму). Несмотря на то, что график кажется необычным, он часто бывает достаточно полезен, особенно, когда нам важно показать форму распределения. По обычному ящику с усами мы бы поняли только, что распределение симметрично и что выбросов нет, а по скрипичной диаграмме мы видим, что распределение похоже на равномерное — перепадов частот не наблюдается, они примерно одинаковы для всех значений.