Урок 6. Основы графики с ggplot2

В этом уроке мы поговорим про более продвинутую визуализацию — построение графиков с помощью библиотеки `ggplot2`. Эта библиотека часто используется на практике, когда в задачи входит построение красивых графиков со сложной структурой: графики по группам наблюдений, наложение графиков друг на друга, графики со сложной легендой и так далее. В качестве примеров и источников вдохновения посмотрим на некоторые графики из галерии `ggplot2`. Как мы уже обсуждали, библиотека `ggplot2` является составной частью библиотеки `tidyverse`, которую мы устанавливали ранее (хотя, конечно, она может использоваться и самостоятельно вне `tidyverse`). Подгрузим библиотеку:

```
library(tidyverse)
```

У библиотеки `ggplot2` есть своя философия, поняв которую, строить графики гораздо легче. Во-первых, графики `ggplot` многослойные, то есть строятся они поэтапно, по слоям. Сначала указывается датафрейм, с которой мы работаем, и интересующие нас показатели (первый слой), затем указывается тип графика (второй слой), затем настройки для подписей, легенды и прочее (остальные слои). Все слои добавляются через `+`.

Начнем разбираться с `ggplot` на примере гистограмм.

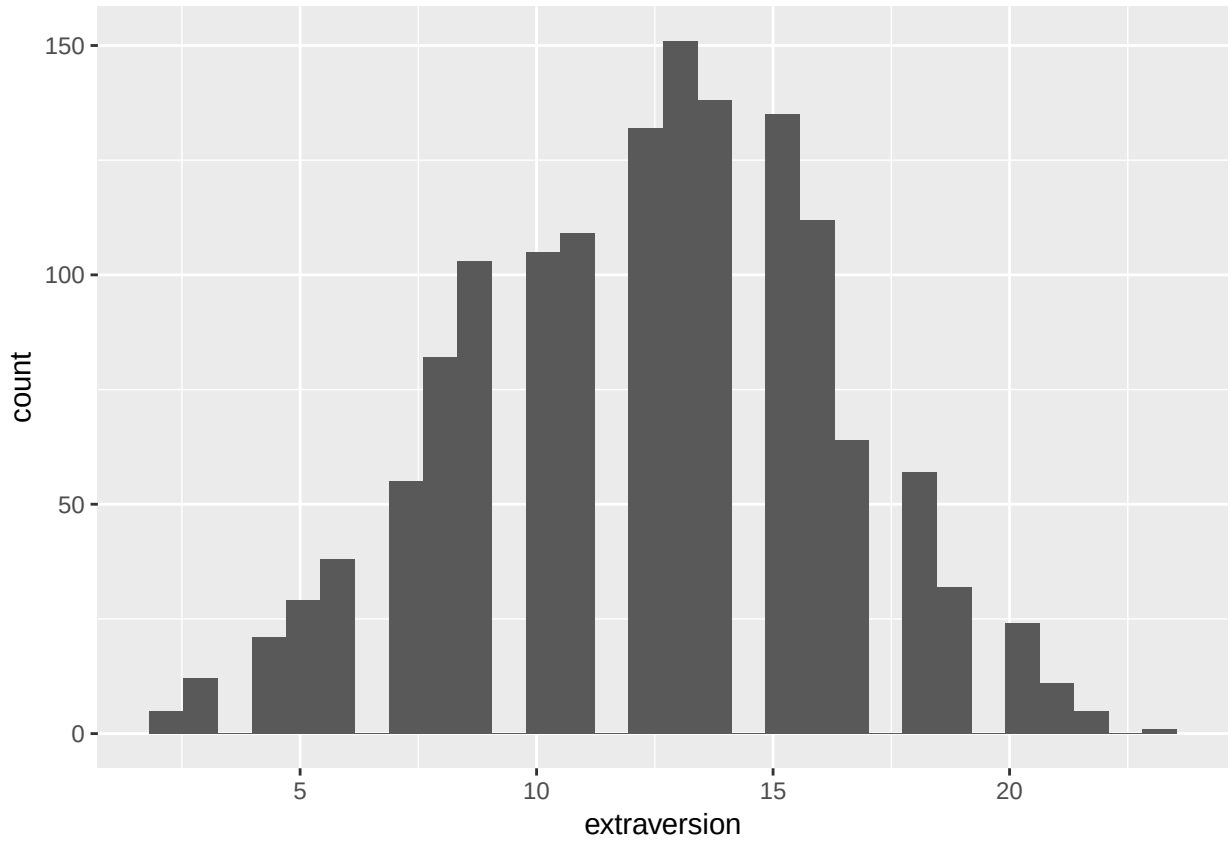
В этом уроке мы поработаем с файлом, в котором содержатся результаты исследования (Cowles, 1994), в котором респонденты фиксировали свой пол и участие/не участие в волонтёрской деятельности, а также заполняли анкету, по которой определялся уровень экстраверсии-интроверсии и уровень тревожности. Вопрос отчасти был в том, правда ли, что более открытые люди, экстраверты, чаще становятся волонтерами. Сейчас мы это проверять не будем, а перейдем к визуализации. Загрузим датафрейм.

```
dat <- read.csv("Cowles.csv")
```

Построим гистограмму для индекса экстраверсии (`extraversion`). Укажем датафрейм `dat`, сам показатель, значения которого идут по оси x, запишем внутри `aes()`, функции, которая отвечает за эстетику (`aes` - *aesthetics*), то есть за наполнение графика. А потом допишем слой `geom_histogram()`, который отвечает за тип графика.

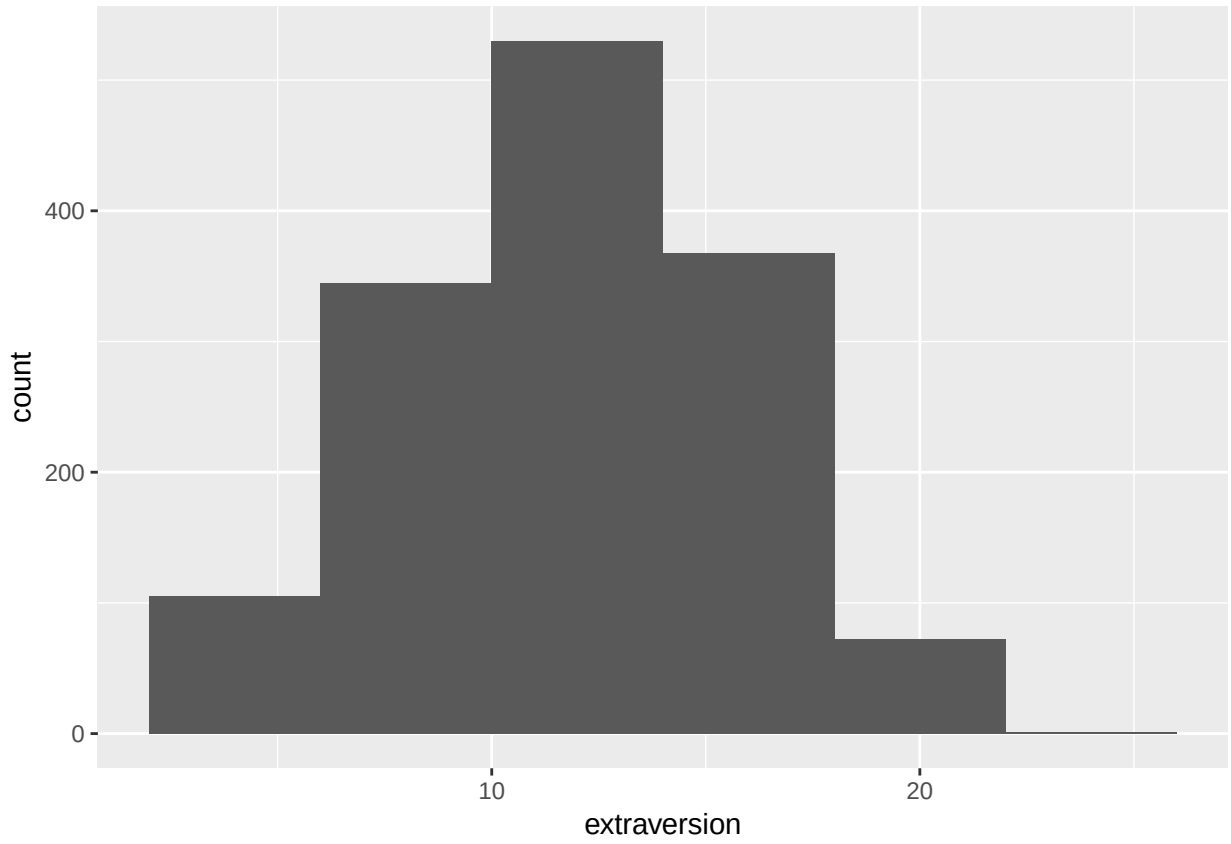
```
ggplot(data = dat, aes(x = extraversion)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



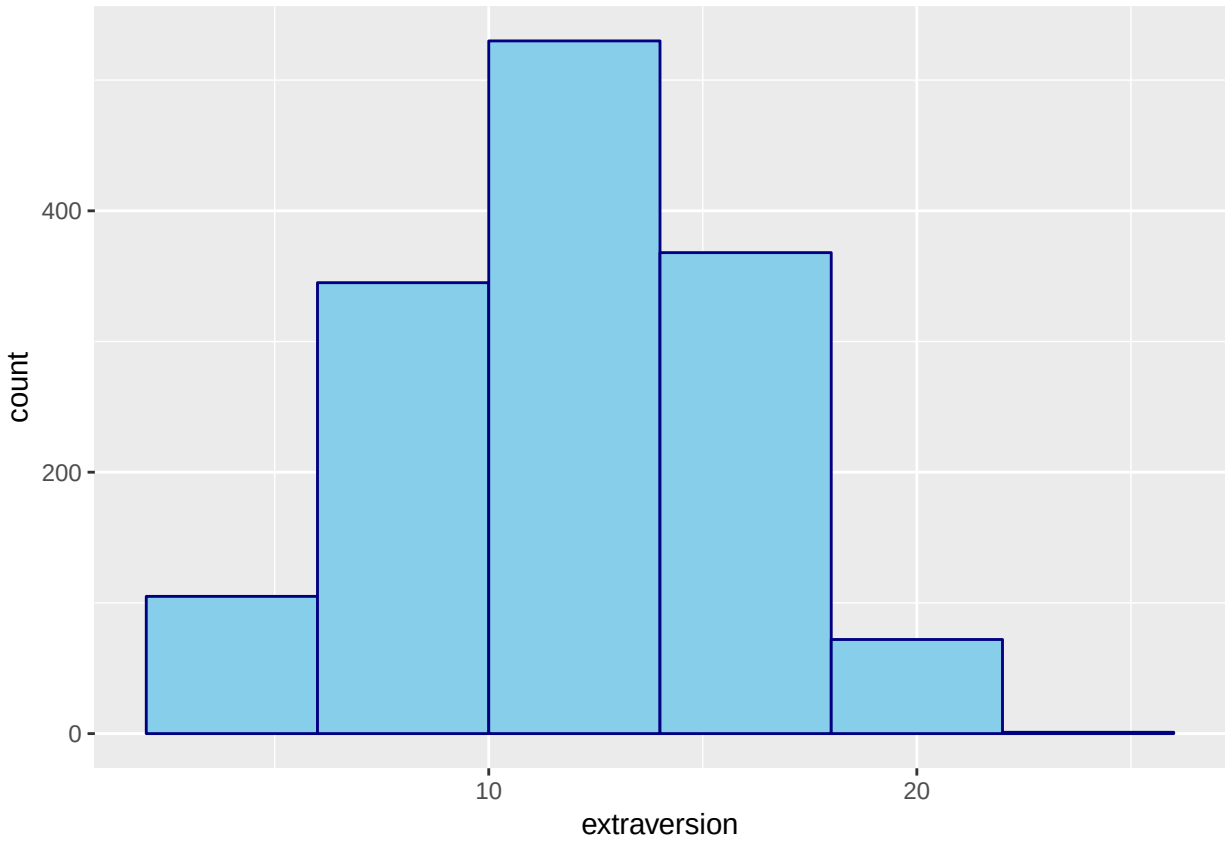
Гистограмма получилась не очень красивой, и не только из-за тёмного цвета. Она получилась какой-то «дырявой». Почему? По умолчанию `ggplot` всегда строит гистограмму с 30 столбцами, поэтому в нашем случае столбцы получились слишком узкими. Дробление на интервалы для столбцов гистограммы получилось слишком детальным, поэтому некоторые интервалы оказались пустыми – в них не попало ни одно значение, из-за чего и появились «дырки». Собственно, поэтому R нам и выдал предупреждение `Pick better value with binwidth`. Поменяем шаг у гистограммы (ширину столбца) вручную, добавив `binwidth` внутри `geom_histogram()`:

```
# выставим шаг равен 4
ggplot(data = dat, aes(x = extraversion)) + geom_histogram(binwidth = 4)
```



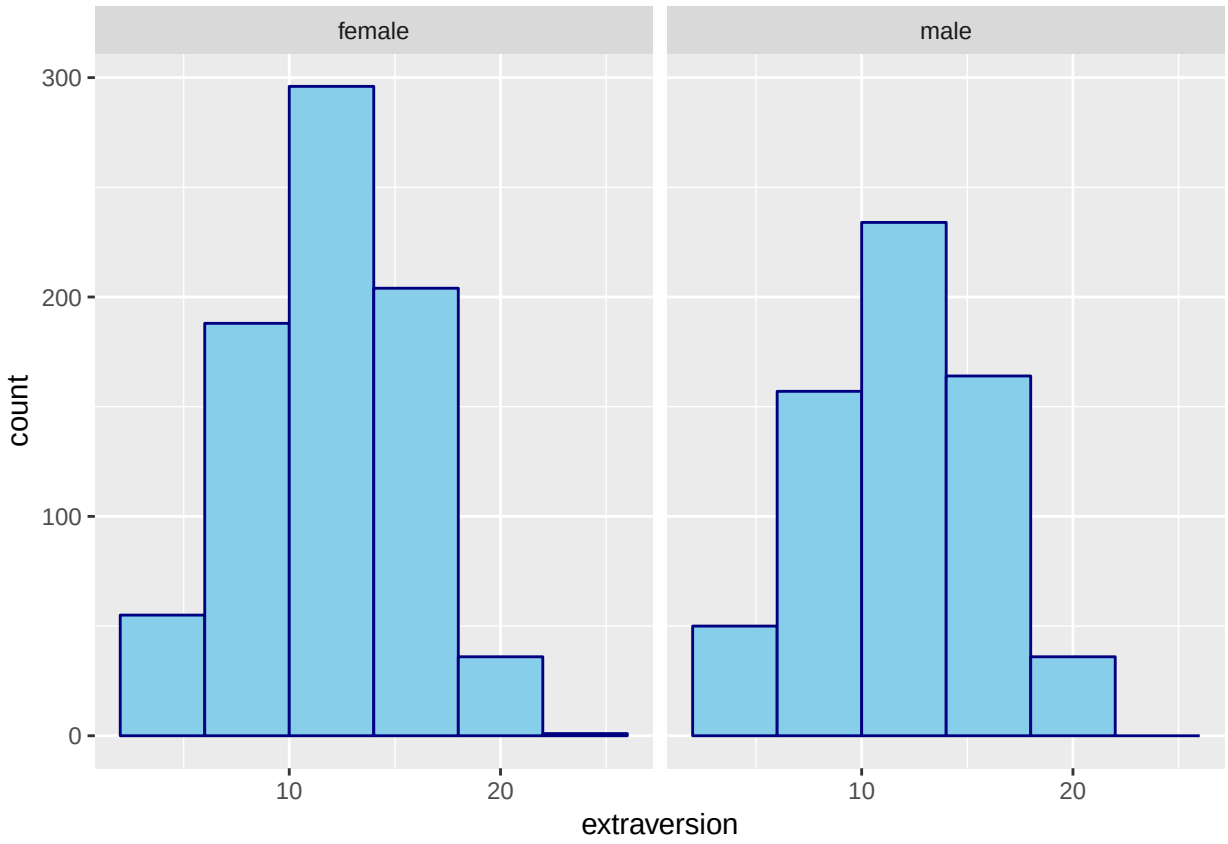
Теперь сделаем гистограмму более красивой — добавим цвет. При изменении цвета «заполненных» (состоящих не из отдельных линий и точек) графиков нужно помнить, что есть два параметра: `color` и `fill`. Параметр `color` отвечает за цвет контура графика, а за не цвет их заливки. А уже `fill` – как раз за заливку.

```
# гистограмма небесно-голубого цвета,
# столбцы которой очерчены синей линией
ggplot(data = dat, aes(x = extraversion)) +
geom_histogram(binwidth = 4, fill = "skyblue", color = "navy")
```



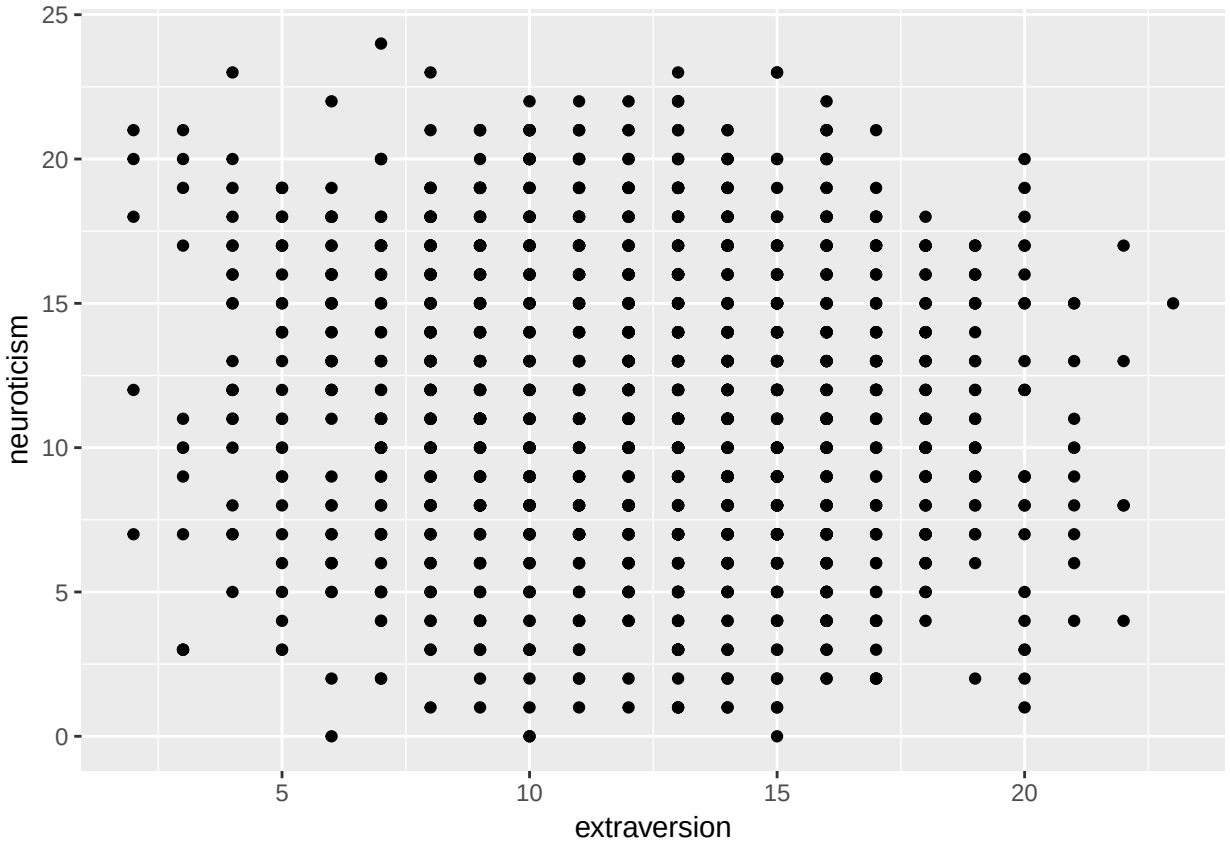
А теперь попробуем построить гистограммы для `extraversion` по группам, отдельно для респондентов мужского пола, отдельно — для женского. С помощью `ggplot2` это сделать гораздо проще, чем с помощью обычной функции `hist()`: не нужно сохранять строки в отдельные датафреймы и после строить гистограммы, можно просто добавить ещё один слой для группировки. Слой называется `facet_wrap` («фасетки»), так как графики для каждой группы отображаются в отдельных окнах-фасетках. Переменная группировки, в нашем случае `sex`, указывается после «тильды» (`~`).

```
ggplot(data = dat, aes(x = extraversion)) +
geom_histogram(binwidth = 4, fill = "skyblue", color = "navy") +
facet_wrap(~sex)
```



Теперь построим диаграмму рассеяния (*scatterplot*) — график, который иллюстрирует связь между двумя числовыми показателями. Построим диаграмму рассеяния для иллюстрации связи между индексом экстраверсии и индексом тревожности. Для этого просто выберем слой `geom_point()`, который используется для графиков с точками, и внутри `aes()` укажем, какой показатель должен идти по оси x, а какой — по оси y:

```
ggplot(data=dat, aes(x = extraversion, y = neuroticism)) +  geom_point()
```



По такой диаграмме, к сожалению, видно немного: облако точек слишком рассеяно, и наклон этого облака кажется нулевым. То есть, можно заключить, что, скорее всего, связи между показателями экстраверсии и тревожности нет. Но нас сейчас больше волнует техническая часть, нежели содержательная, поэтому в заключение давайте сделаем следующее: добавим заголовок, изменим подписи к осям и сделаем так, чтобы цвет точек зависел от того, является ли человек волонтером или нет.

В философии `ggplot` есть одно удобное разделение. Если настройки графика (цвет точек, заливка столбцов, размер точек) не зависит от других показателей в данных, то мы указываем необходимые параметры вне слоя `ggplot` и вне функции `aes()`, если зависят – то внутри `aes()`. Чтобы было более понятно, посмотрим на примеры.

Пример 1. Строим диаграмму рассеяния для роста и веса человека, хотим, чтобы все точки на диаграмме рассеяния были зелёными. Цвет точек никак не зависит от других показателей, это наше чисто «дизайнерское» решение, считаем, что так будет красиво.

Код будет выглядеть так (просто код для условного датафрейма `DAT`, без графика, график построим уже для наших данных Cowles, чуть позже):

```
# по оси x - рост, по оси y - вес, цвет указан отдельно, в слое для точек

ggplot(data = DAT, aes(x = height, y = weight)) +
geom_point(color = "green"")
```

Пример 2. Строим диаграмму рассеяния для роста и веса человека, хотим, чтобы точки на диаграмме рассеяния, соответствующие женщинам, были красными, а мужчинам — синими. Различия в цветах точек здесь сделаны не просто для красоты, они несут смысловую нагрузку: цвет точки указывает на пол человека.

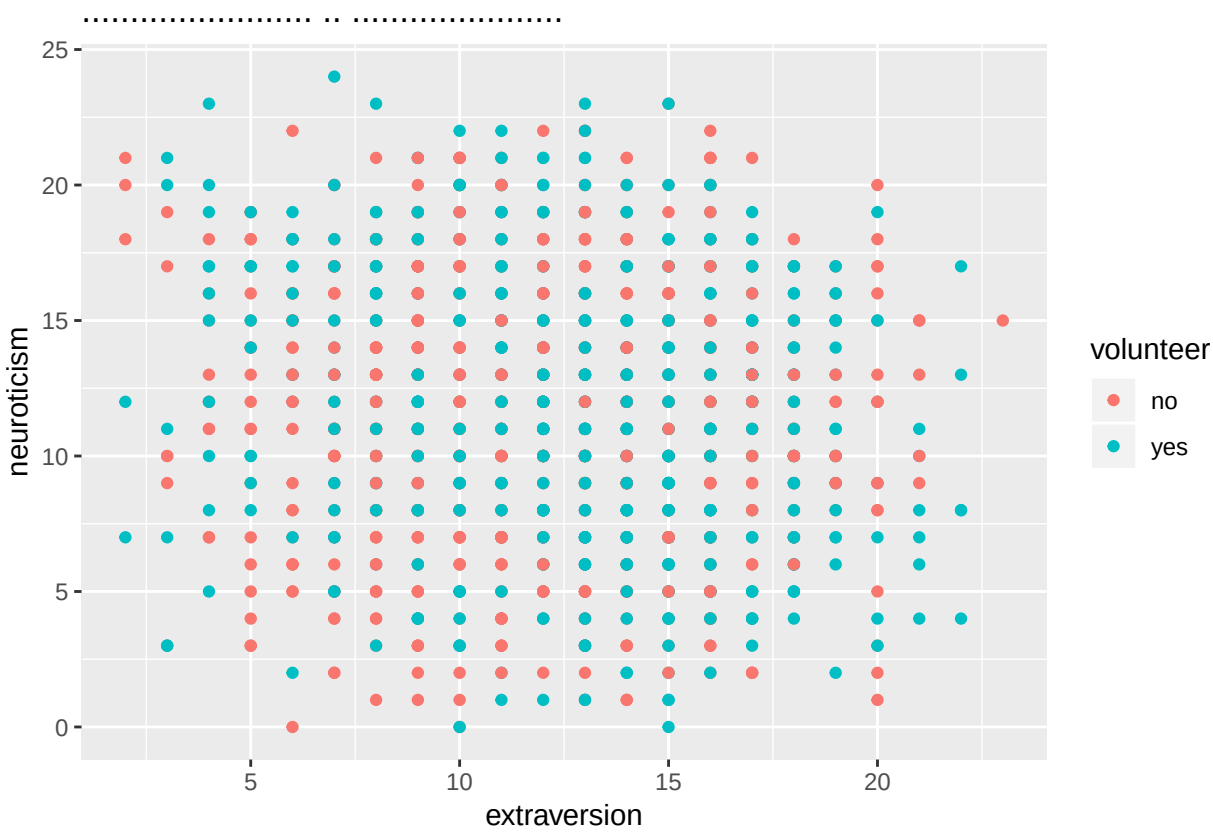
Код для этого примера будет выглядеть так:

```
# по оси x - рост, по оси y - вес
# но теперь указываем, что цвет точек зависит от значений в столбце "пол"
# и указываем это внутри aes
ggplot(data = DAT, aes(x = height, y = weight, color = sex)) +
geom_point()
```

Почему во втором примере указываем цвет именно в `aes()`? Все просто: функция `aes()` внутри слоя `ggplot()` извлекает данные прямо из датафрейма, того, который мы указали в data, а остальные слои так не могут, могут лишь реагировать на параметры, не связанные с рассматриваемым датафреймом – общий цвет точек, размер точек в пунктах, размер и цвет заголовков и прочее.

Теперь применим полученные знания к нашему случаю с волонтерами. Как можно догадаться, цвет мы будем указывать внутри `aes()`, так как мы не просто красим все точки одним цветом, а делаем так, чтобы выбор цвета зависел от показателя `volunteer` в датафрейме:

```
ggplot(data=dat, aes(x = extraversion, y = neuroticism,
color = volunteer)) + geom_point() +
labs(title = "Экстраверсия и тревожность",
xlab = "Экстраверсия",
ylab = "Тревожность")
```



В этом примере мы добавили в `aes()` аргумент color и прописали, что цвет должен зависеть от показателя `volunteer`. По умолчанию используются голубой и розовый цвета, поскольку R изначально активно использовался в биологии, но, конечно, их можно настроить. Но предлагаю поработать с настройками не сейчас, а в домашнем задании.