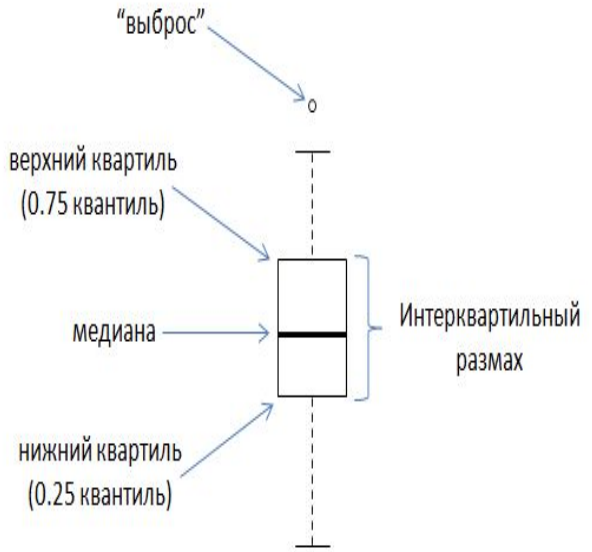
Графические возможности и визуализация данных

Лабораторная работа №3.

Расширенные графические возможности

Диаграммы размахов, или "ящики с усами" (англ. *box-whisker plots*), получили свое название за характерный вид: точку или линию, соответствующую медиане или средней арифметической, окружает прямоугольник ("ящик"), длина которого соответствует одному из показателей разброса или точности оценки генерального параметра. Дополнительно от этого прямоугольника отходят "усы", также соответствующие по длине одному из показателей разброса или точности. Графики этого типа очень популярны, поскольку позволяют дать очень полную статистическую характеристику анализируемой совокупности. Кроме того, диаграммы размаха можно использовать для визуальной экспресс-оценки разницы между двумя и более группами (например, между датами отбора проб, экспериментальными группами, участками пространства, и т.п.).

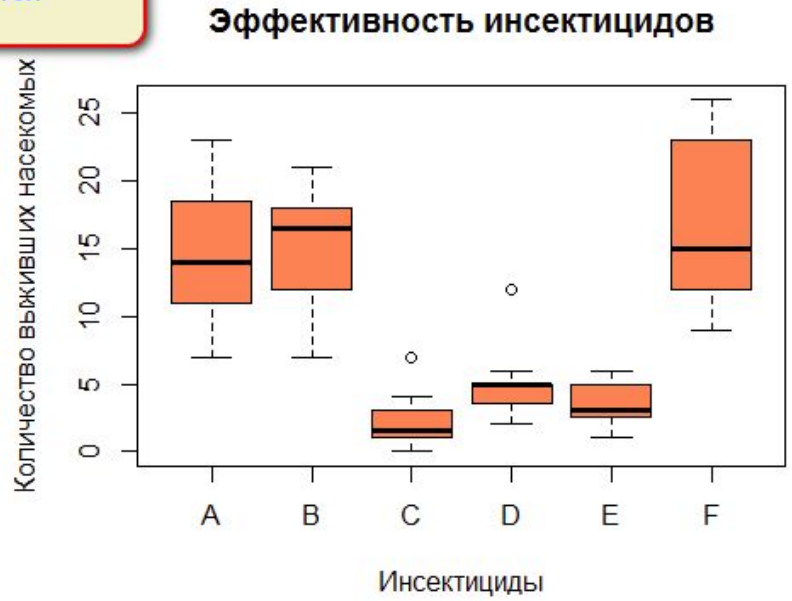
В R для построения диаграмм размахов служит функция `boxplot()`. Строение получаемых при помощи этой функции "ящиков с усами" представлено ниже:



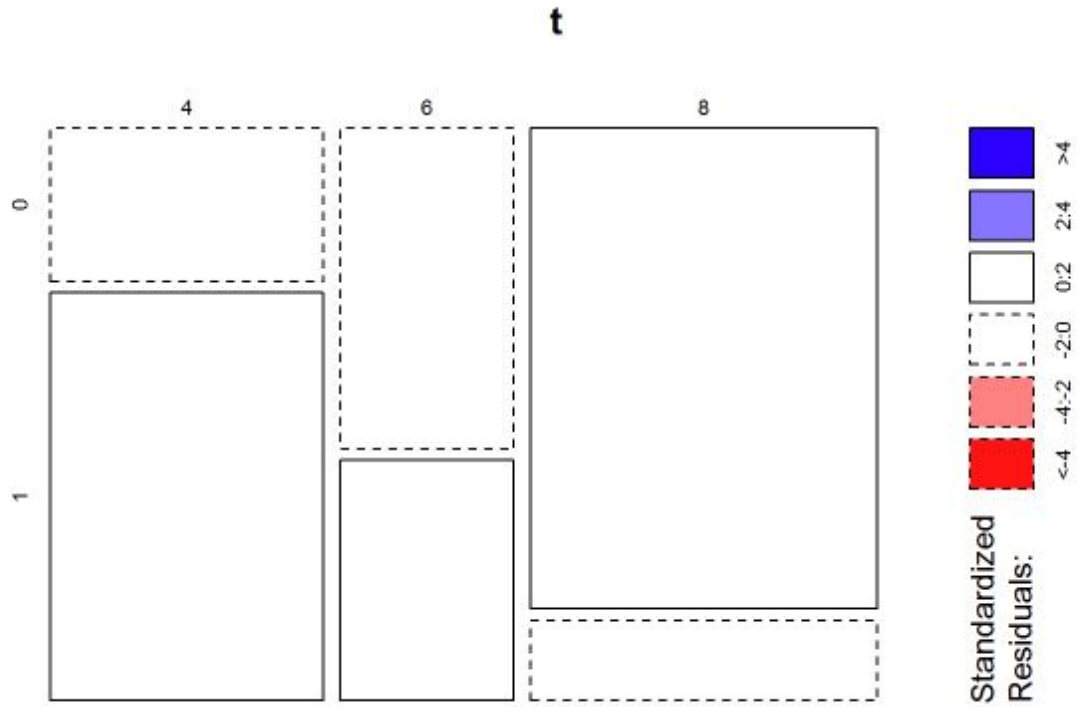
```r
boxplot(count ~ spray,
        xlab = "Инсектициды",
        ylab = "Количество выживших насекомых",
        main = "Эффективность инсектицидов",
        col = "coral", data = InsectSprays)
```

Параметр, по которому сортируем (в данному случае вид химиката A,B,C,D,E,F)

Параметр, переменная (по оси У), которая подсчитывается



```r
# построение мозаичного графика
t <- table(mtcars$cyl, mtcars$am)
mosaicplot(t, shade = T)
```



Пакет ggplot2

Пакет ggplot2 позволяет использовать более расширенные графические возможности языка R.

Если на вашем рабочем месте не установлен данный пакет, для его скачивания и установки наберите следующую команду : install.packages(“ggplot2”).

Затем подключите данную библиотеку library(“ggplot2”).

Основная функция данного графического пакета ggplot().

Пример ее использования изображен на рисунке ниже. Первым аргументом указывается имя датафрейма df, откуда вы берете данные (перед началом работы все используемые данные необходимо поместить в один датафрейм). Затем внутри функции aes() перечисляются переменные, откуда функция берет данные. На рсиунке ниже данные берутся только из 1 столбца - mpg (разгон).

На следующей строке после оператора + указыцвается тип графика, которный вы строите. Например гистограмма, как на примере. Стои заменить , что все названия графиков начинаются с фразы geom_****.

В ggplot2 график является результатом взаимодействия ряда элементов:

- Массив данных (**data**)
- Схема соответствия переменных из массива визуальным средствам (**aes**thetic)
- Геометрический объект (**geom**)
- Статистическое преобразование (**stat**)
- Координатная система (**coord**)
- Ориентиры (**guide**)
- Панели (**facet**)
- Художественное оформление (**theme**)

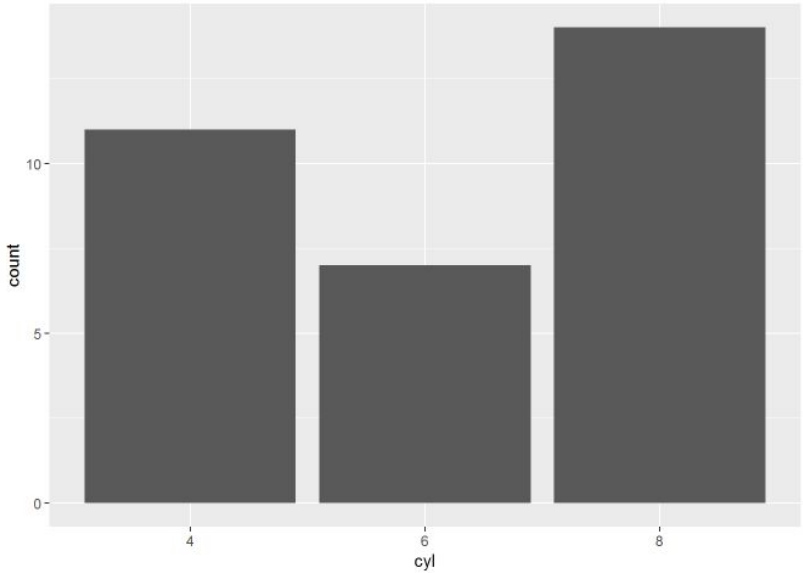
Например, пользователь сообщает компьютеру, что хочет использовать массив данных про автомобили, переменная “скорость” будет выражена через положение по горизонтали, переменная “тормозной путь” через положение по вертикали, всё это нужно нарисовать с помощью геометрических объектов типа “точка”.

Вы спросите, а почему не заданы статистическое преобразование, координатная система, ориентиры и прочее. Все явно не указанные пользователем элементы графика берутся из значений по умолчанию. Например, если пользователь в качестве графического объекта указал тип “точки”, то по умолчанию статистических преобразований производиться не будет. А если он укажет тип “столбик”, то наблюдения в исходной переменной будут сгруппированы, а результатом применения статистики станет количество наблюдений в каждой группе.

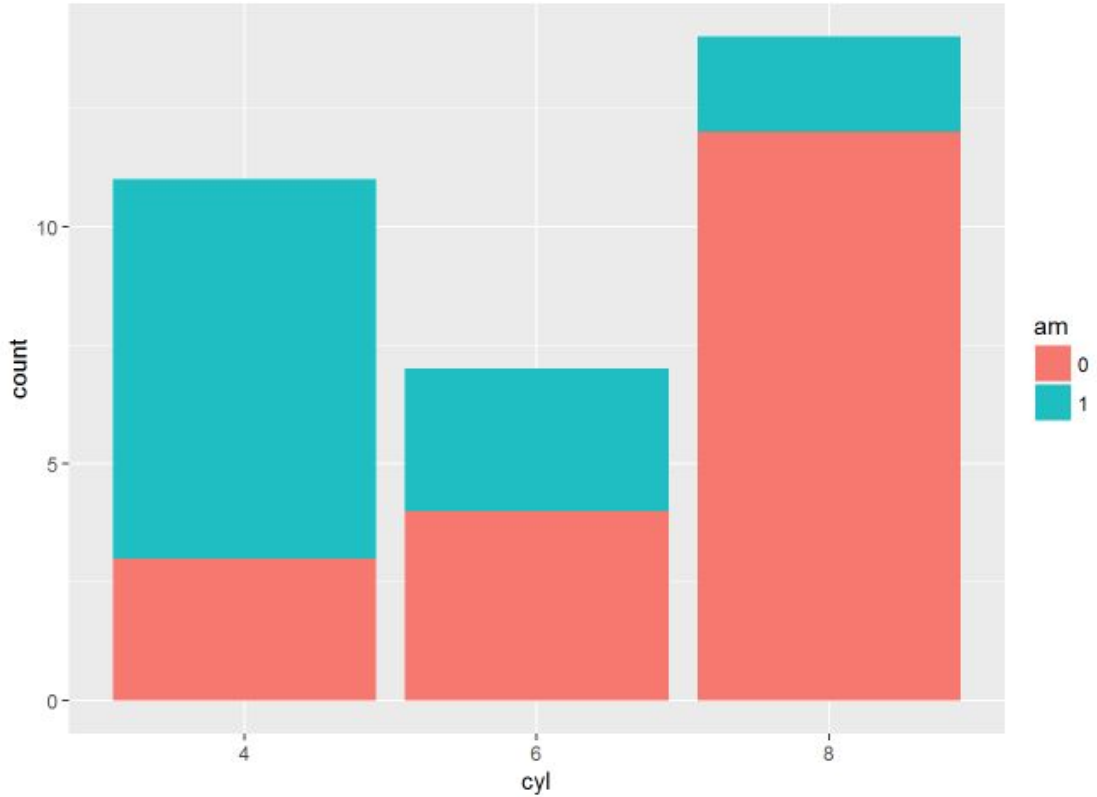
Визуализация номинативных переменных

Рассмотрим несколько примеров графикоф для анализа номинативных данных.

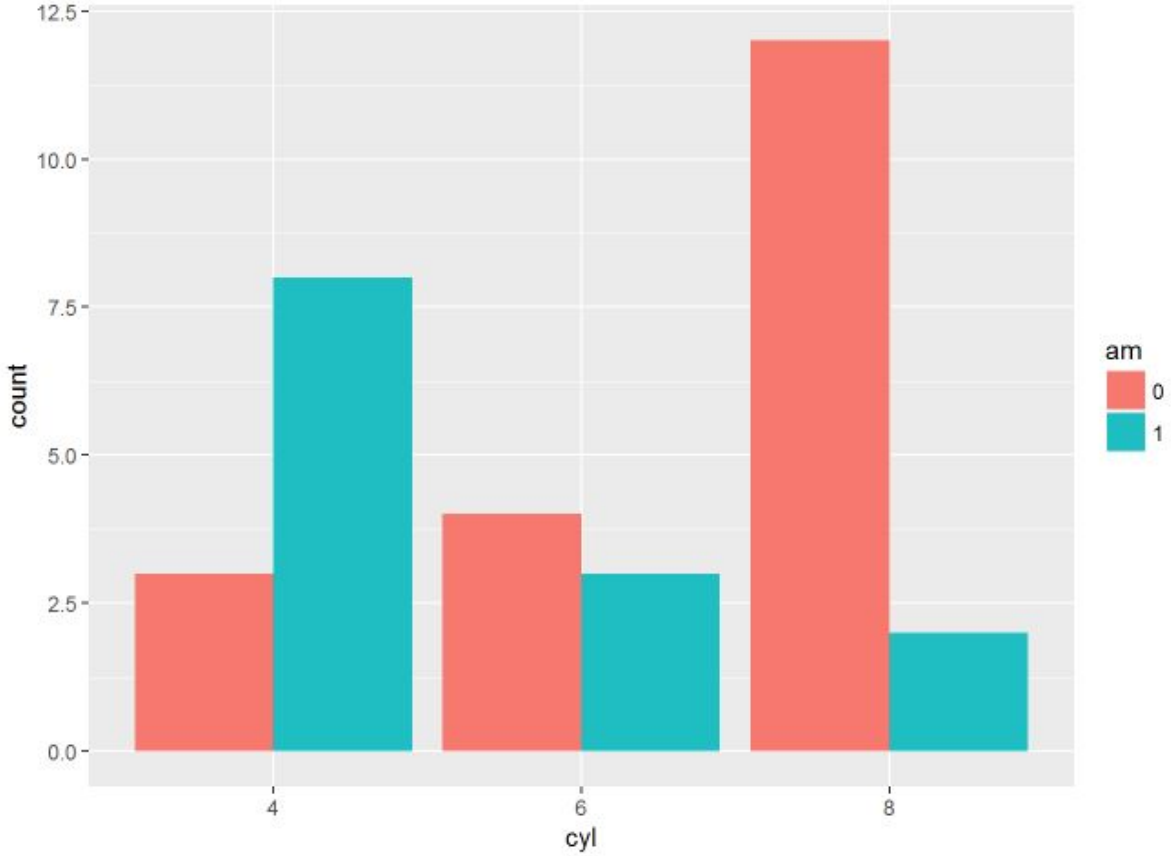
```
# сначала переведем наши номинативные переменные в фактор
mtcars$am <- factor(mtcars$am)
mtcars$vs <- factor(mtcars$vs)
mtcars$cyl <- factor(mtcars$cyl)


# будем использовать библиотеку ggplot для построения графиков

#install.packages("ggplot2")
library(ggplot2)

# построим простую гистограмму частот
ggplot(mtcars, aes(x = cyl)) +
    geom_bar()
```



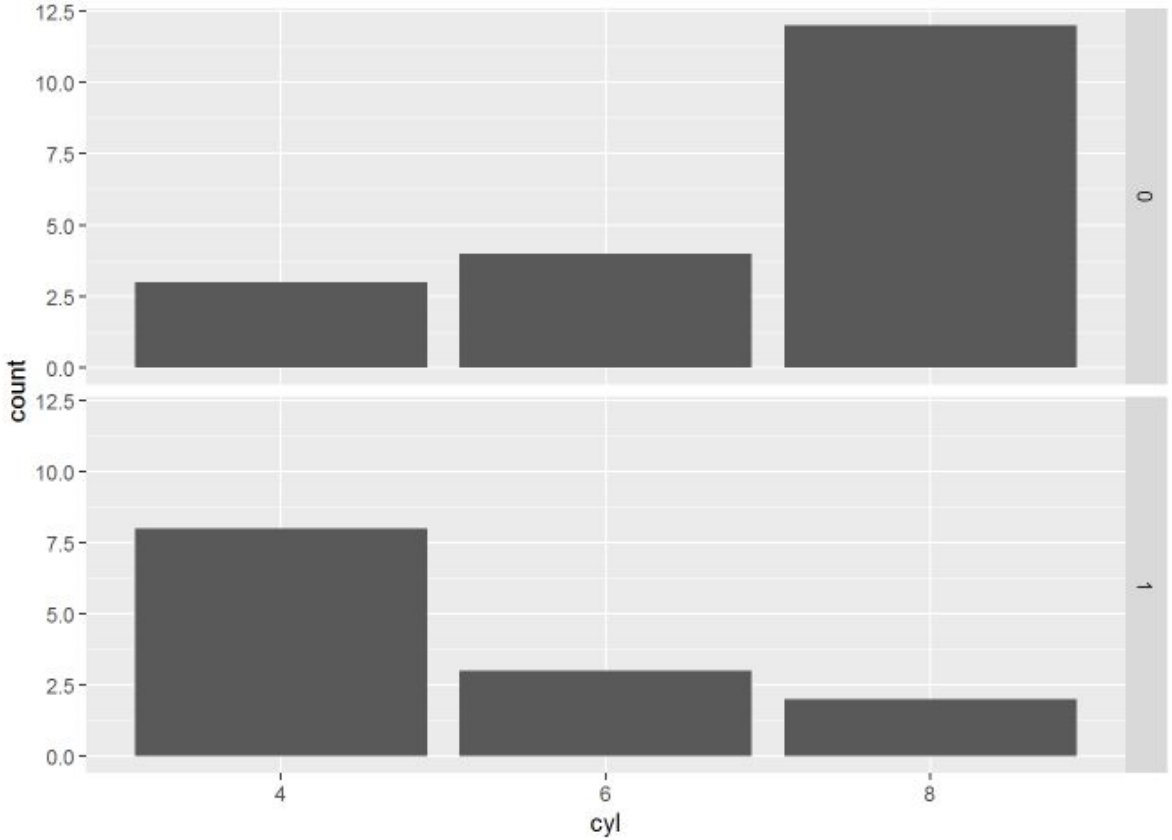
```r
# добавим другие переменные на график
# три разных варианта
ggplot(mtcars, aes(x = cyl, fill = am)) +
    geom_bar()
```



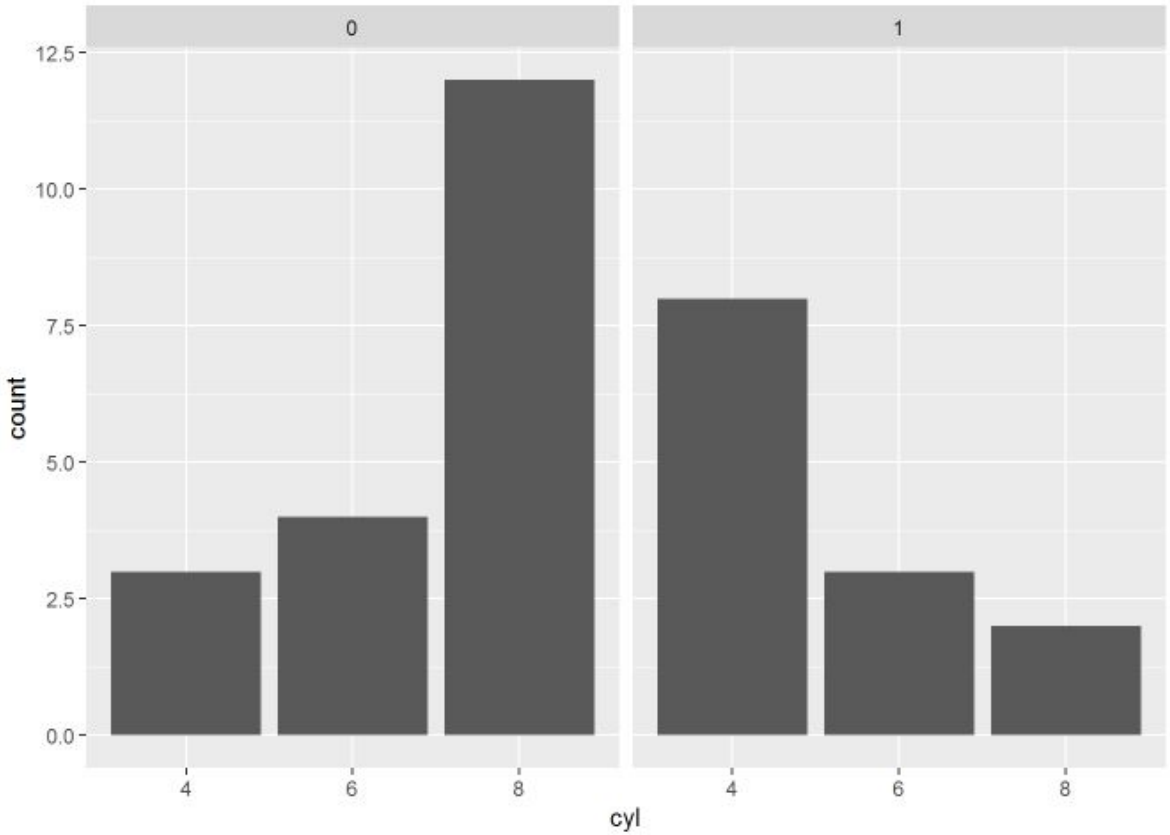
```
ggplot(mtcars, aes(x = cyl, fill = am)) +
    geom_bar(position = 'dodge')
```



```
ggplot(mtcars, aes(x = cyl)) +
    geom_bar() +
    facet_grid(am ~ .)
```



```
ggplot(mtcars, aes(x = cyl)) +
    geom_bar() +
    facet_grid(. ~ am)
```



```r
df  <- mtcars
df$vs  <- factor(df$vs  , labels = c("V", "S"))
df$am  <- factor(df$am  , labels = c("Auto", "Manual"))

library(ggplot2)

ggplot(df, aes(x = mpg))+
  geom_histogram()
```

8:18 (Top Level) R Script

Global Environment

Data

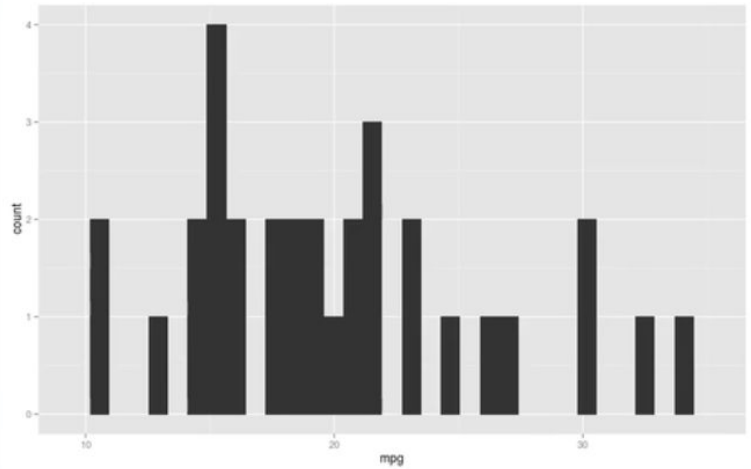
df 32 obs. of 11 variables

Console ~/R/RCourse/

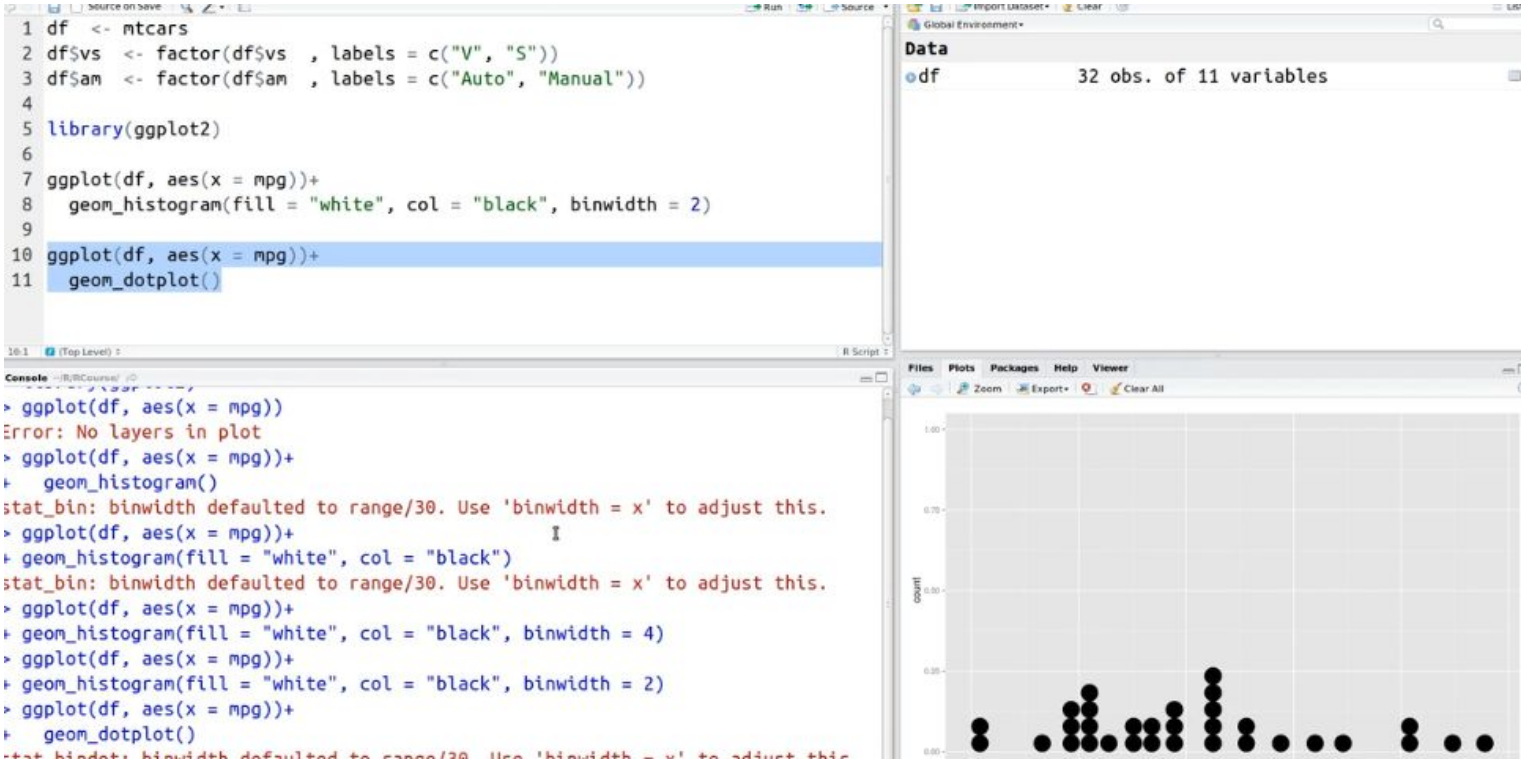
```
> library(ggplot2)
> ggplot(df, aes(x = mpg))
Error: No layers in plot
> ggplot(df, aes(x = mpg))+
+   geom_histogram()
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
>
```

Files Plots Packages Help Viewer

Zoom Export Clear All



Те же самые данные можно изобразить в виде dotplot графика, где каждая точка отображает каждый объект данных:



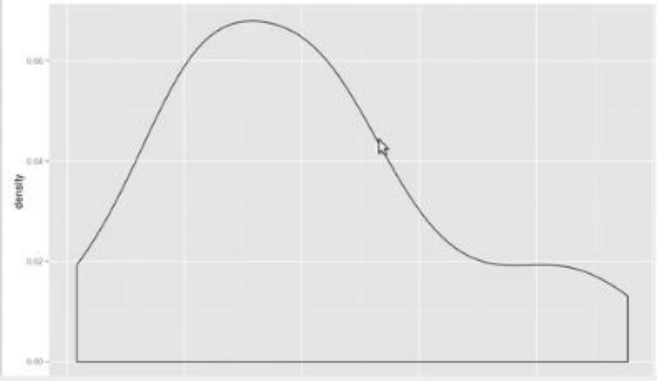
Те же самые данные в виде графика плотности вероятности

```
df$vs  <- factor(df$vs  , labels = c("V", "S"))
df$am  <- factor(df$am  , labels = c("Auto", "Manual"))

library(ggplot2)

ggplot(df, aes(x = mpg))+
  geom_histogram(fill = "white", col = "black", binwidth = 2)

ggplot(df, aes(x = mpg))+
  geom_dotplot()

ggplot(df, aes(x = mpg))+
  geom_density()
```

```
> ggplot(df, aes(x = mpg))+
+   geom_histogram()
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+ geom_histogram(fill = "white", col = "black")
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+ geom_histogram(fill = "white", col = "black", binwidth = 4)
> ggplot(df, aes(x = mpg))+
+ geom_histogram(fill = "white", col = "black", binwidth = 2)
> ggplot(df, aes(x = mpg))+
+   geom_dotplot()
stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+   geom_density()
```

Data

df 32 obs. of 11 variables

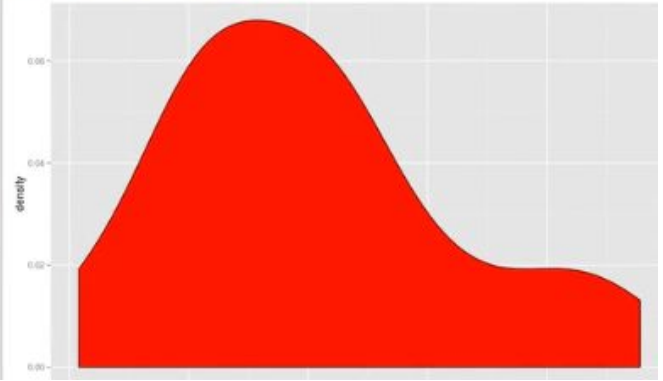


Параметр fill отвечает за цвет наполнения фигур.

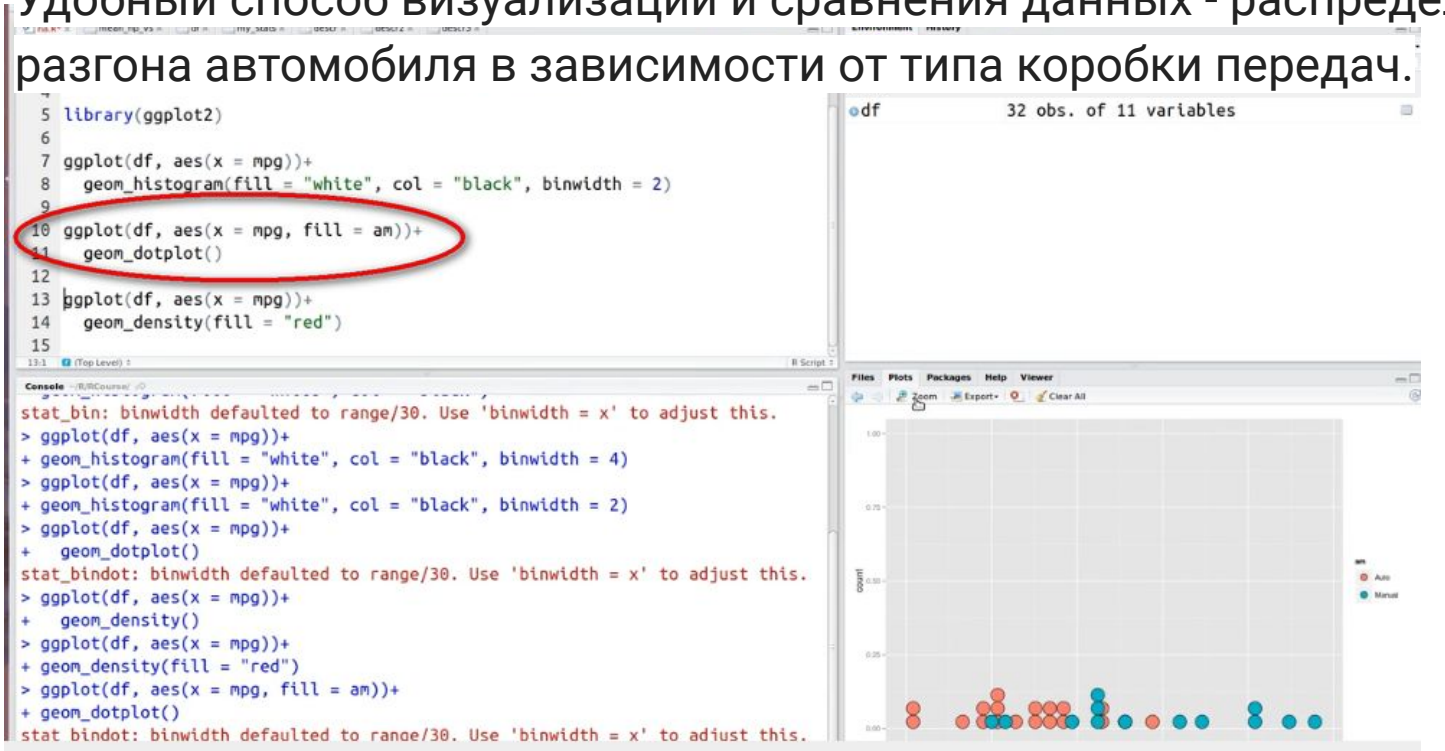
```
df$am  <- factor(df$am  , labels = c("Auto", "Manual"))

library(ggplot2)

ggplot(df, aes(x = mpg))+
  geom_histogram(fill = "white", col = "black", binwidth = 2)

ggplot(df, aes(x = mpg))+
  geom_dotplot()

ggplot(df, aes(x = mpg))+
  geom_density(fill = "red")
```

```
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+ geom_histogram(fill = "white", col = "black")
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+ geom_histogram(fill = "white", col = "black", binwidth = 4)
> ggplot(df, aes(x = mpg))+
+ geom_histogram(fill = "white", col = "black", binwidth = 2)
> ggplot(df, aes(x = mpg))+
+   geom_dotplot()
stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+   geom_density()
> ggplot(df, aes(x = mpg))+
+ geom_density(fill = "red")
```

```
Data
df          32 obs. of 11 variables
```



Если в параметр fill поместить другую переменную (тип коробки передач - автомат или механика), функция aes() автоматически сформирует 2 набора данных для 2-х коробок передач, а ggplot() раскрасит их в 2 разных цвета. Удобный способ визуализации и сравнения данных - распределение времени разгона автомобиля в зависимости от типа коробки передач.



```r
library(ggplot2)

ggplot(df, aes(x = mpg))+
  geom_histogram(fill = "white", col = "black", binwidth = 2)

ggplot(df, aes(x = mpg, fill = am))+
  geom_dotplot()

ggplot(df, aes(x = mpg))+
  geom_density(fill = "red")

ggplot(df, aes(x = mpg, fill = am))+
  geom_density()
```

```
+ geom_histogram(fill = "white", col = "black", binwidth = 4)
> ggplot(df, aes(x = mpg))+
+ geom_histogram(fill = "white", col = "black", binwidth = 2)
> ggplot(df, aes(x = mpg))+
+   geom_dotplot()
stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+   geom_density()
> ggplot(df, aes(x = mpg))+
+ geom_density(fill = "red")
> ggplot(df, aes(x = mpg, fill = am))+
+ geom_dotplot()
stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg, fill = am))+
+   geom_density()
```

Environment History

Import Dataset · Clear · List

Global Environment

Data

df 32 obs. of 11 variables

Files Plots Packages Help Viewer

Zoom · Export · Clear All

Изменим прозрачность слоев:

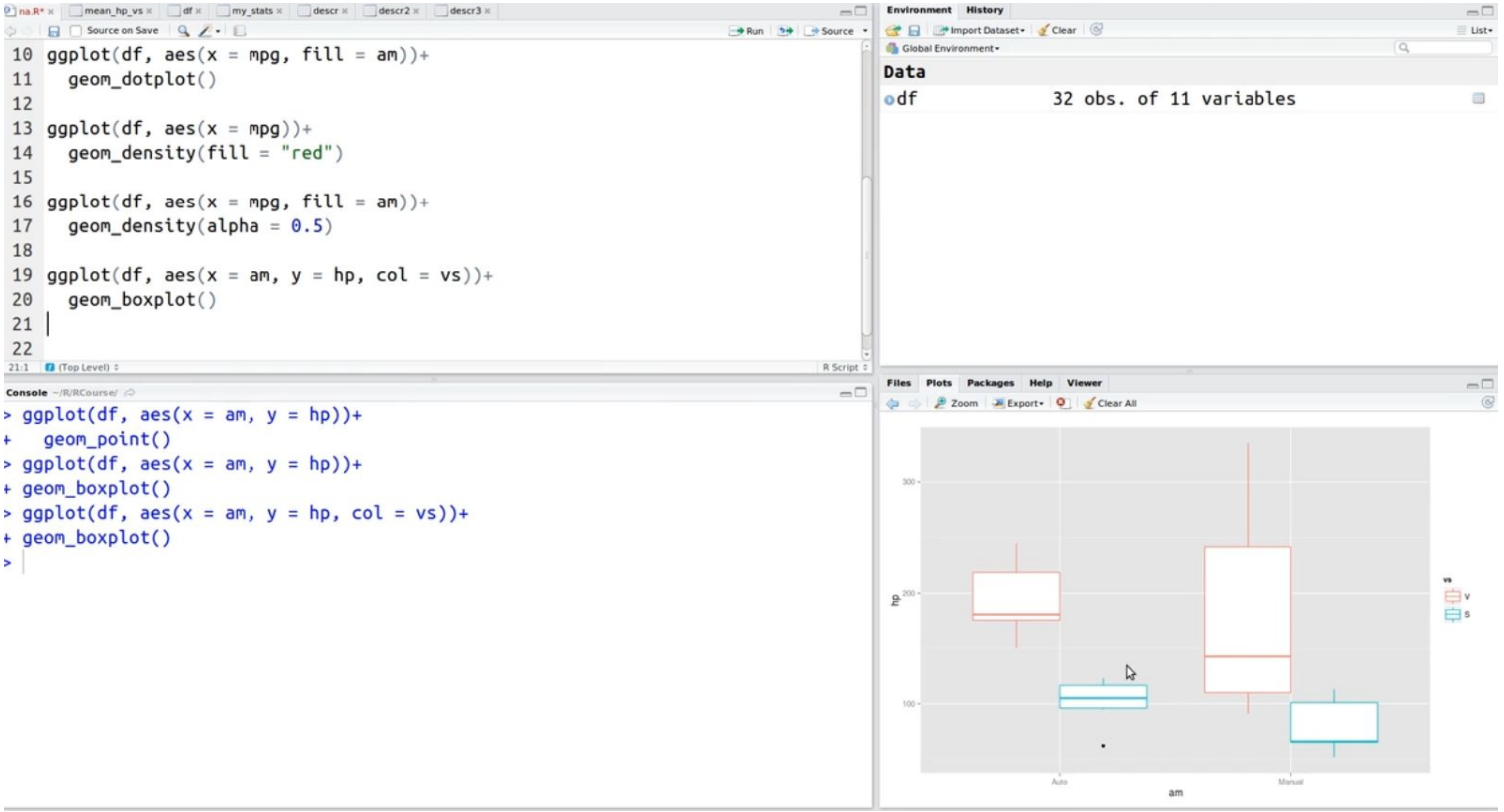
```
library(ggplot2)

ggplot(df, aes(x = mpg))+
  geom_histogram(fill = "white", col = "black", binwidth = 2)

ggplot(df, aes(x = mpg, fill = am))+
  geom_dotplot()

ggplot(df, aes(x = mpg))+
  geom_density(fill = "red")

ggplot(df, aes(x = mpg, fill = am))+
  geom_density(alpha = 0.5)
```

```
+   geom_dotplot()
stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg))+
+   geom_density()
> ggplot(df, aes(x = mpg))+
+ geom_density(fill = "red")
> ggplot(df, aes(x = mpg, fill = am))+
+ geom_dotplot()
stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
> ggplot(df, aes(x = mpg, fill = am))+
+   geom_density()
> ggplot(df, aes(x = mpg, fill = am))+
+ geom_density(alpha = 0.2)
> ggplot(df, aes(x = mpg, fill = am))+
+   geom_density(alpha = 0.5)
```

Data

df 32 obs. of 11 variables

Или вот так: график boxplot изменения мощности hp в зависимости от типа коробки передач am и градация по цвету по параметру тип двигателя vs.

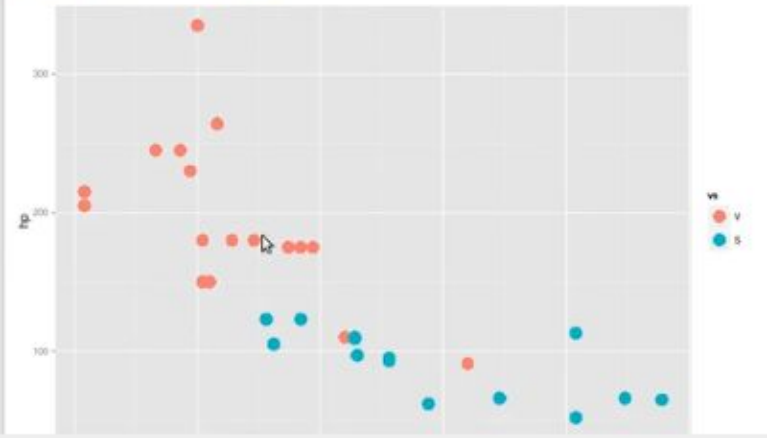


```r

ggplot(df, aes(x = mpg))+
  geom_density(fill = "red")

ggplot(df, aes(x = mpg, fill = am))+
  geom_density(alpha = 0.5)

ggplot(df, aes(x = am, y = hp, col = vs))+
  geom_boxplot()

ggplot(df, aes(x = mpg, y = hp, col = vs))+
  geom_point(size = 6)

```

Data

df 32 obs. of 11 variables

```
> ggplot(df, aes(x = am, y = hp))+
+   geom_point()
> ggplot(df, aes(x = am, y = hp))+
+ geom_boxplot()
> ggplot(df, aes(x = am, y = hp, col = vs))+
+ geom_boxplot()
> ggplot(df, aes(x = mpg, y = hp))+
+   geom_point()
> ggplot(df, aes(x = mpg, y = hp))+
+   geom_point(size = 6)
> ggplot(df, aes(x = mpg, y = hp, col = vs))+
+ geom_point(size = 6)
>
```



```r
ggplot(df, aes(x = mpg, fill = am))+
  geom_density(alpha = 0.5)

ggplot(df, aes(x = am, y = hp, col = vs))+
  geom_boxplot()

ggplot(df, aes(x = mpg, y = hp, col = vs, size = qsec))+
  geom_point()

my_plot  <- ggplot(df, aes(x = mpg, y = hp, col = vs, size = qsec))+
  geom_point()
```

```
> ggplot(df, aes(x = am, y = hp))+
+ geom_boxplot()
> ggplot(df, aes(x = am, y = hp, col = vs))+
+ geom_boxplot()
> ggplot(df, aes(x = mpg, y = hp))+
+   geom_point()
> ggplot(df, aes(x = mpg, y = hp))+
+   geom_point(size = 6)
> ggplot(df, aes(x = mpg, y = hp, col = vs))+
+ geom_point(size = 6)
> ggplot(df, aes(x = mpg, y = hp, col = vs, size = qsec))+
+   geom_point()
> my_plot  <- ggplot(df, aes(x = mpg, y = hp, col = vs, size = qsec))+
+ geom_point()
> my_plot
>
```

Environment | History

Import Dataset | Clear

Global Environment

Data

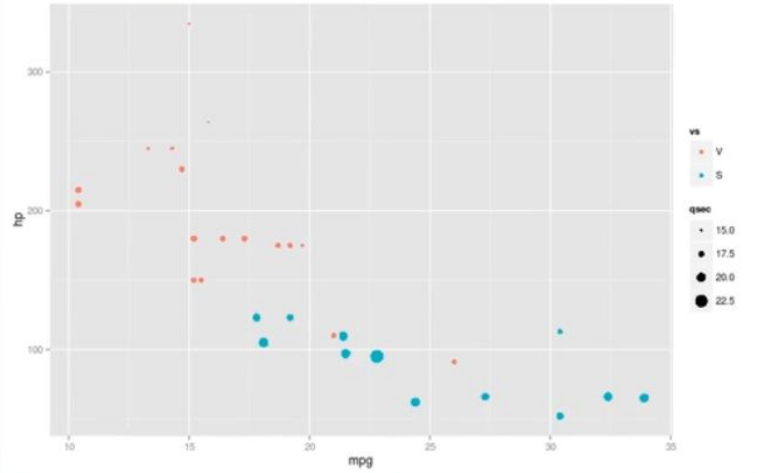
df	32 obs. of 11 variables

Values

my_plot	List of 9

Files | Plots | Packages | Help | Viewer

Zoom | Export | Clear All



Задание 1

Создайте график boxplot времени разгона автомобиля в зависимости от количества цилиндров во встроенном наборе данных mtcars. В графике подпишите названия осей.

Задание 2

При помощи функции ggplot() или boxplot() постройте график boxplot, используя встроенные в R данные airquality. По оси x отложите номер месяца, по оси y — значения переменной Ozone.

На графике boxplot отдельными точками отображаются наблюдения, отклоняющиеся от 1 или 3 квартиля больше чем на полтора межквартильных размаха. Сколько таких наблюдений присутствует в сентябре (месяц №9)?

Обратите внимание, что для корректного отображения графика ggplot ожидает факторную переменную по оси x.

Задание 3

Используем знакомые нам данные mtcars.

Нужно построить scatterplot с помощью ggplot из ggplot2, по оси x которого будет mpg, по оси y - disp, а цветом отобразить переменную (hp).

Полученный график нужно сохранить в переменную plot1. Таким образом в ответе должен быть скрипт:

```
plot1 <- ggplot(data, aes())+
  geom_****()
```