ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ УПРАВЛЕНИЯ

Основан в 1919 году

Обработка данных в среде офисных приложений: введение в R

Светлана Андреевна Суязова (Аксюк)
sa_aksyuk@guu.ru

18 сентября 2021

Лекция 2

Инструменты предварительного анализа данных и построения линейных моделей

- графические системы в R: base, lattice и ggplot2

- очистка данных
- корреляционный анализ и линейные регрессионные модели

Базовая графика в R

- все средства находятся в базовой сборке (`base`);
- можно собрать любой статический график с нуля
- результат сложно сохранить как объект
- вывод на графическое устройство: экран, файл
- Чтобы сделать красиво, нужно очень много кода

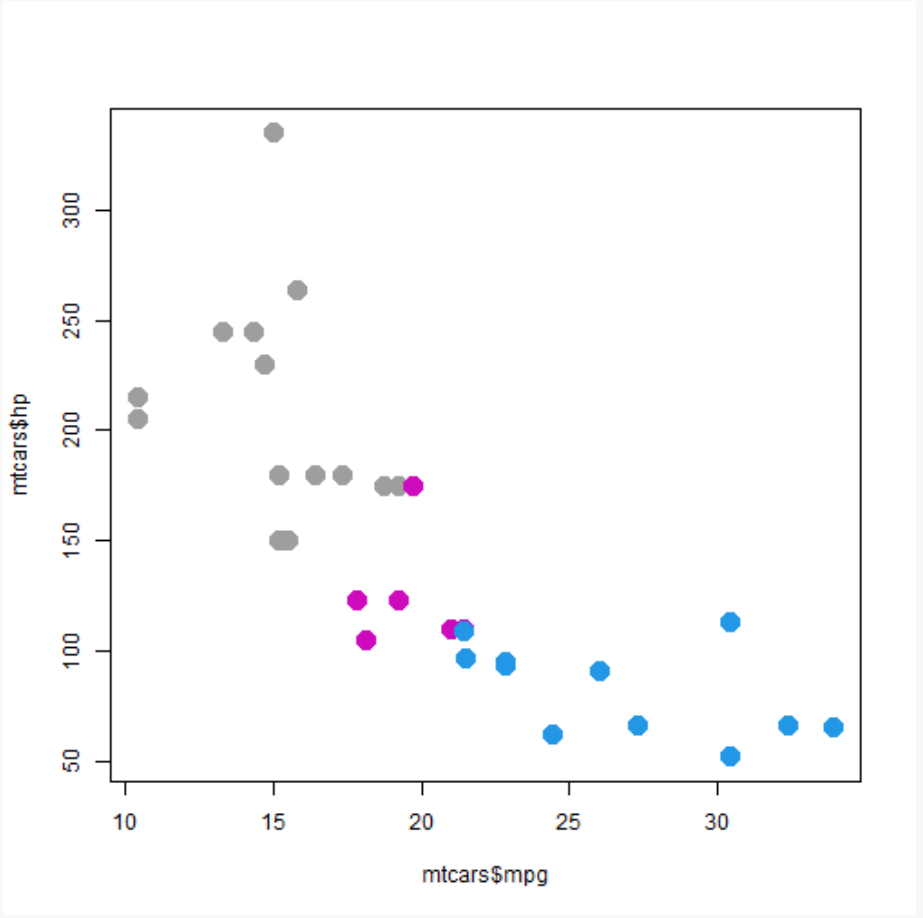
Базовая графика в R

- График собирается “слоями”, предыдущие слои нельзя отменить
- Начинается с функции высокого уровня: `plot()`, `curve()`, `boxplot()`, `hist()` и др.
- Элементы добавляются на активный график функциями низкого уровня: `points()`, `abline()`, `axis()`, `mtext()`, `text()` и др.

Пример простого графика base

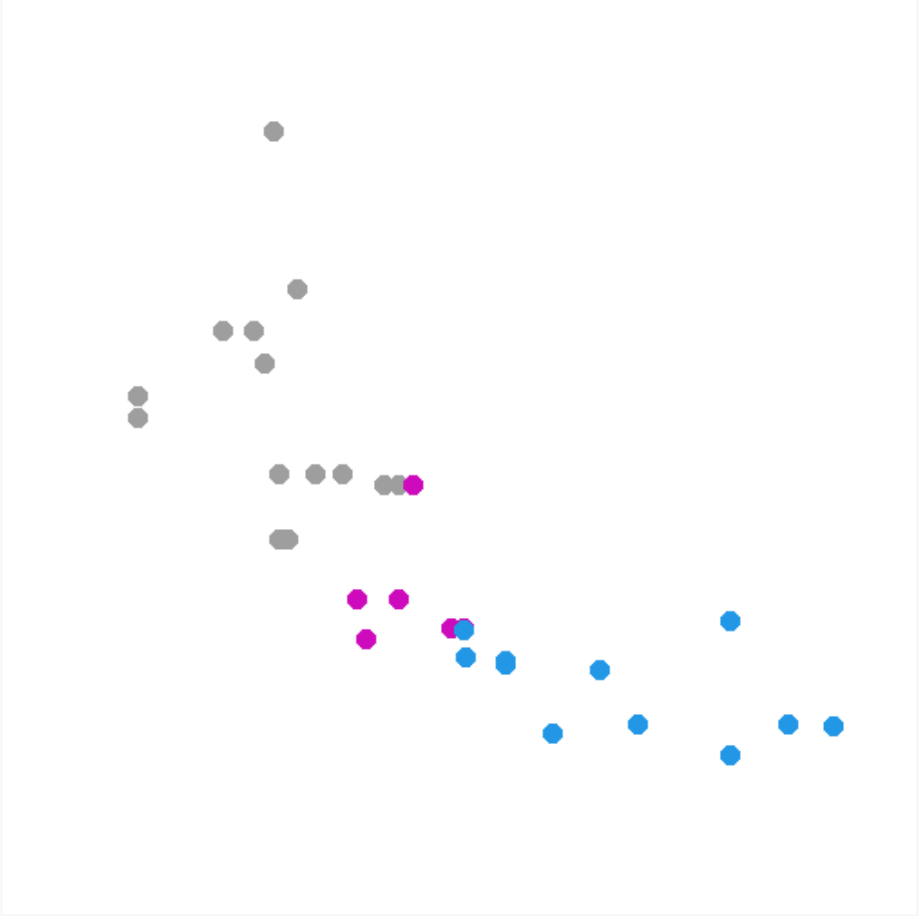
```
plot(mtcars$mpg,
    mtcars$hp,
    pch = 21,
    col = mtcars$cyl,
    bg = mtcars$cyl,
    cex = 2)
```

- оси пересекаются между делениями
- непонятные подписи осей
- мелкий текст



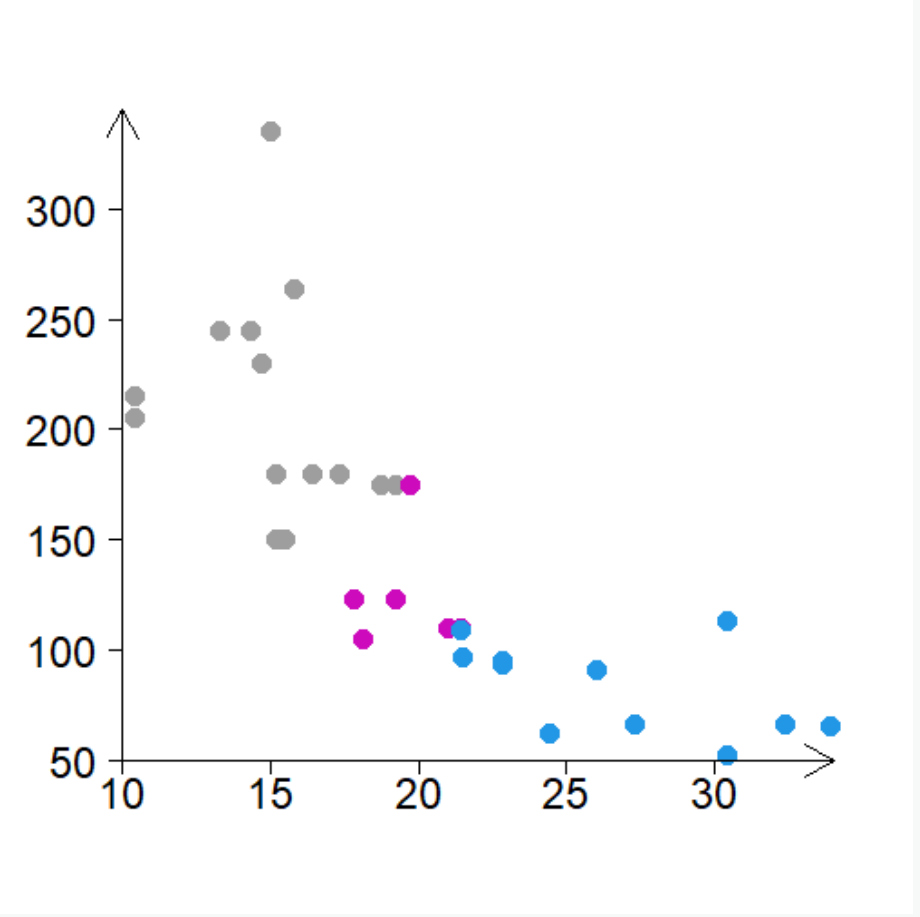
Отключаем оси...

```
plot(mtcars$mpg,
     mtcars$hp,
     pch = 21,
     col = mtcars$cyl,
     bg = mtcars$cyl,
     cex = 2,
     # пустые подписи осей
     xlab = '', ylab = '',
     # скрыть сами оси
     axes = F)
```



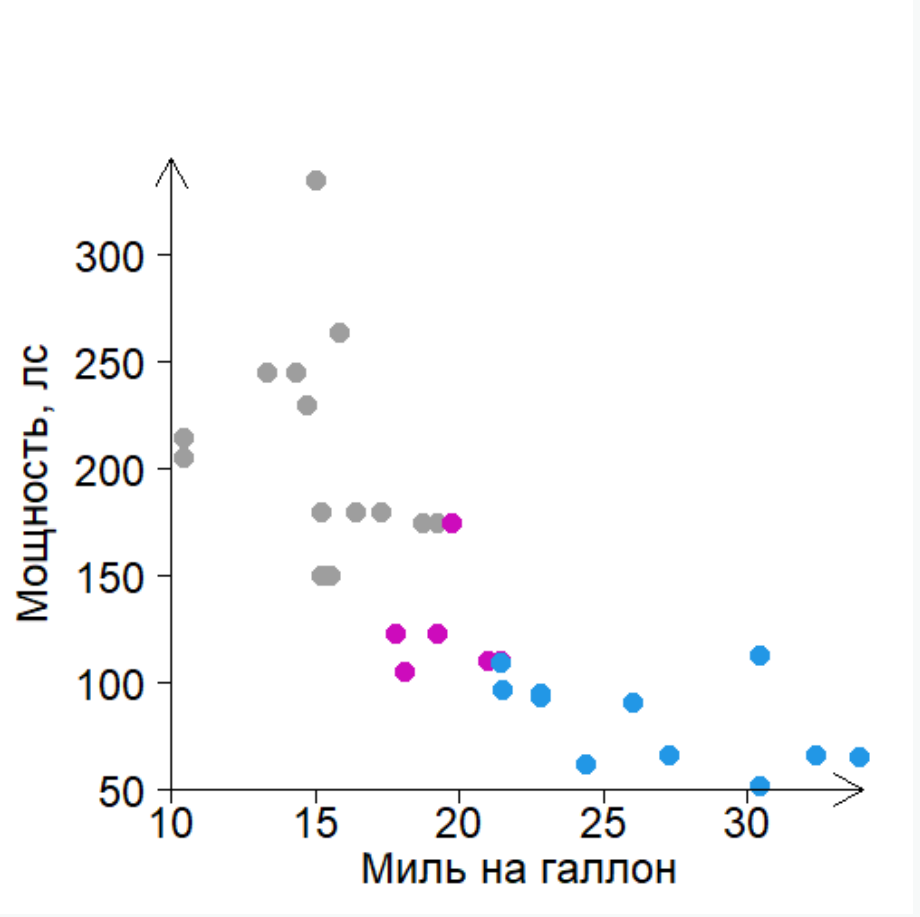
...перерисовываем оси...

```
# строим график без осей
plot(...)

# оси с настройками
axis(1, pos = 50,
     cex.axis = 2)
axis(2, pos = 10, las = 2,
     cex.axis = 2)

# добавляем стрелки
arrows(x0 = c(30, 10),
       y0 = c(50, 300),
       x1 = c(34, 10),
       y1 = c(50, 345))
```



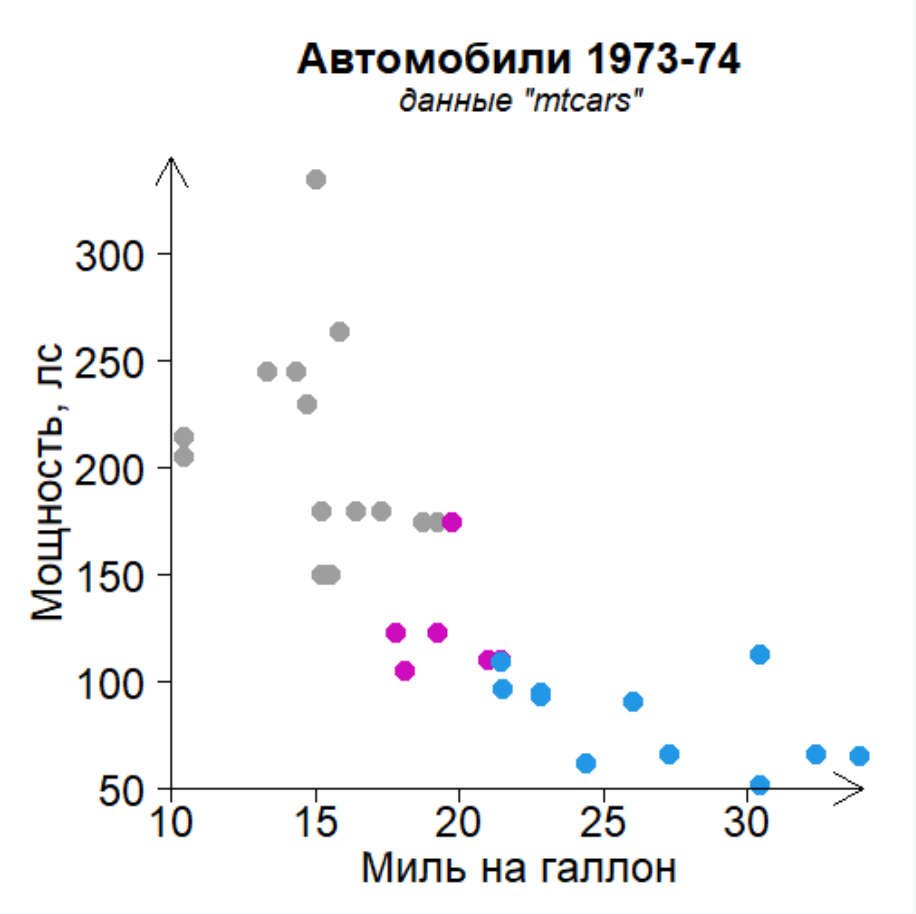
...добавляем подписи осей...

```
# ширина полей графика
par(mar = c(4, 6, 6, 1))

plot(...)     # график
axis(...)     # оси
arrows(...)   # стрелки

# подписи осей
mtext("Миль на галлон",
      side = 1, line = 2,
      cex = 2)
mtext("Мощность, лс",
      side = 2, line = 4,
      cex = 2)
```



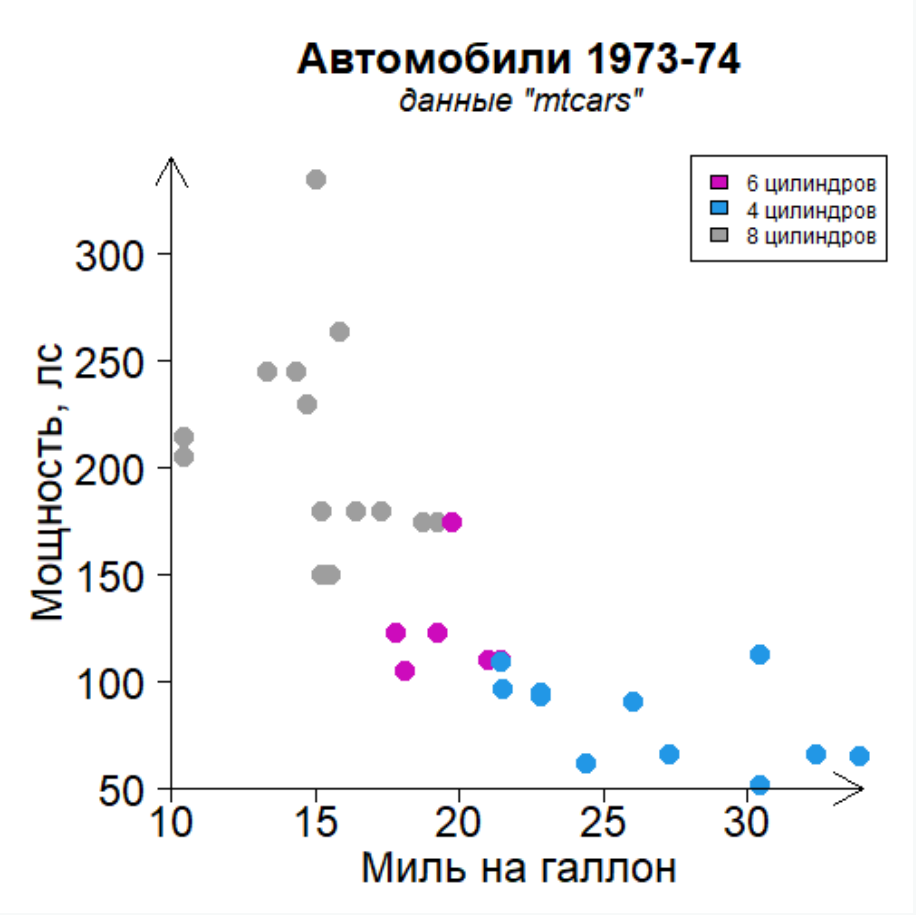
...добавляем заголовок...

```
par(...)      # поля
plot(...)     # график
axis(...)     # оси
arrows(...)   # стрелки
mtext(...)    # подписи осей

# заголовок
mtext("Автомобили 1973-74",
      side = 3, line = 3,
      cex = 2, font = 2)
# подзаголовок
mtext('данные "mtcars"',
      side = 3, line = 1.5,
      cex = 1.5, font = 3)
```



...добавляем легенду

```
par(...)     # поля
plot(...)    # график
axis(...)    # оси
arrows(...) # стрелки
mtext(...)   # подписи осей
mtext(...)   # заголовки

# легенда
mark <- unique(mtcars$cyl)
legend('topright',
  legend = paste(mark,
                 'цилиндров'),
  fill = mark)
```



Итого десять вызовов функций

```
par(mar = c(4, 6, 6, 1))       # поля графика

# сам график без осей
plot(mtcars$mpg, mtcars$hp, pch = 21, col = mtcars$cyl,
     bg = mtcars$cyl, cex = 2, xlab = '', ylab = '', axes = F)
# оси
axis(1, pos = 50, cex.axis = 2)
axis(2, pos = 10, cex.axis = 2, las = 2)
# стрелки на концах осей
arrows(x0 = c(30, 10), y0 = c(50, 300), x1 = c(34, 10), y1 = c(50, 345))
# подписи осей
mtext("Миль на галлон", side = 1, line = 2, cex = 2)
mtext("Мощность, лс", side = 2, line = 3.5, cex = 2)

# заголовок и подзаголовок графика
mtext("Автомобили 1973-74", side = 3, line = 3, cex = 2, font = 2)
mtext('данные "mtcars"', side = 3, line = 1.5, cex = 1.5, font = 3)

# легенда
legend('topright', legend = paste(unique(mtcars$cyl), 'цилиндров'),
       fill = unique(mtcars$cyl))
```

Основные функции верхнего уровня

`plot()` – подстраивается под данные: график разброса, временного ряда, графики для объектов специальных типов: дендрограммы, график осыпи, остатки регрессии, и т.д.

`boxplot()` – коробчатые диаграммы;

`hist()` – гистограммы;

`pie()` – круговая диаграмма.

Основные функции нижнего уровня

`abline()` – добавляет прямую на график;

`points()` – добавляет точки наблюдений;

`lines()` – добавляет линию по точкам;

`curve()` – строит кривую по заданой функции.

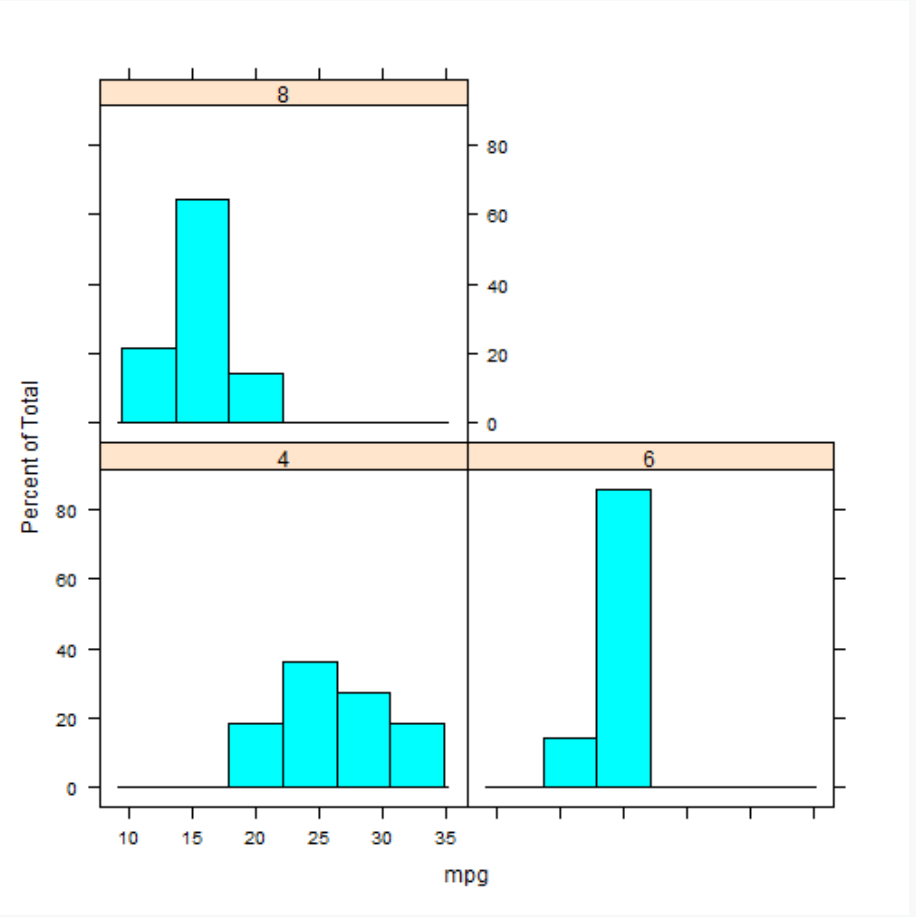
Графическая система `lattice`

- пакет **`lattice`**
- функции оптимизированы для представления кросс-секционных данных с большим количеством признаков (multivariate data)
- упрощена разбивка данных по факторам (цвет, фасетки)
- принцип: одна функция – один график: после построения на график ничего нельзя добавить
- настройка отображения элементов графика затруднена

Пример графика lattice

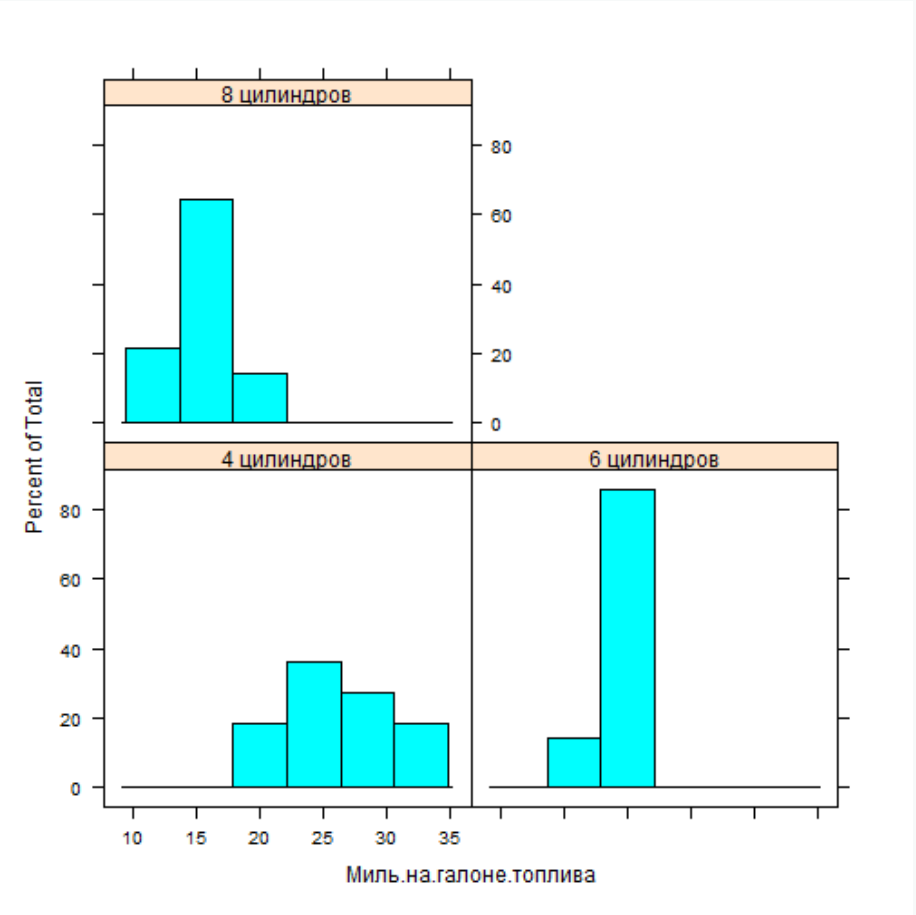
```
library('lattice')

histogram(~ mpg |
            as.factor(cyl),
          data = mtcars)
```

- автоматическое разбиение на фасетки по категориям
- нельзя редактировать подписи фасеток и осей



Чтобы изменить график, нужно менять набор данных

```
# готовим данные
df.plot <-
  mtcars[, c('mpg', 'cyl')]
colnames(df.plot)[1] <-
  'Миль.на.галоне.топлива'
df.plot$cyl <-
  as.factor(paste(df.plot$cyl,
                  'цилиндров'))

# строим график
histogram(~ Миль.на.галоне.топлива |
            cyl, data = df.plot)
```



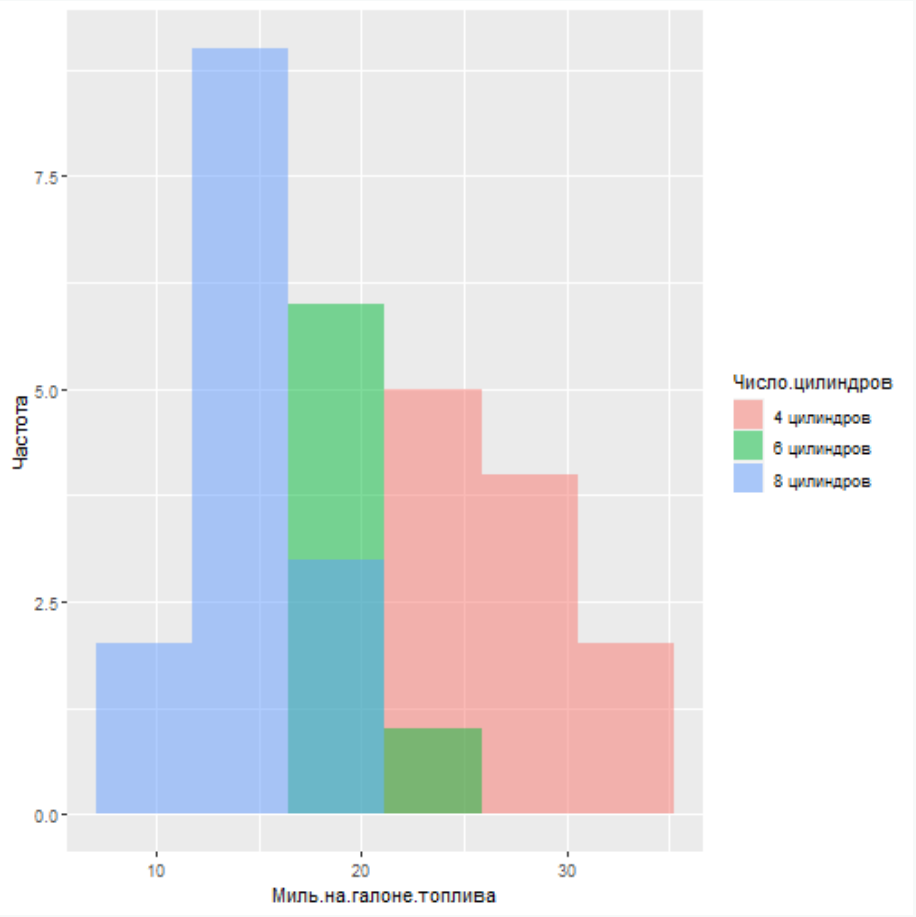
Графическая система `ggplot2`

- пакет **`ggplot2`**
- реализует грамматику графиков Леланда Уилкинсона
- график-предложение составляется из функции-подлежащего (`ggplot()`), функции сказуемого (`geom_lines()`, `geom_points()`, `geom_boxplot()` и др.) и функций-дополнений для настройки отдельных элементов графика
- график строится послойно и содержит графические настройки по умолчанию

Пример графика ggplot2

```
library('ggplot2')

colnames(df.plot)[2] <-
  'Число.цилиндров'

ggplot(data = df.plot,
       aes(x = Миль.на.галоне.топлива,
           fill = Число.цилиндров)) +
 geom_histogram(bins = 6,
                alpha = 0.5,
                position = 'identity')+
 ylab('Частота')
```

- есть настройки отображения графика по умолчанию
- чтобы менять эти настройки отображения, нужны дополнительные параметры функции `theme()`



Резюме по графическим системам в R

- `base` – чтобы быстро посмотреть на данные или построить график нестандартного типа или с нестандартными элементами
- `lattice` – для лаконичного вызова нескольких графиков с разбиением по факторам
- `ggplot2` – чтобы построить график со встроенным оформлением и с дополнительными возможностями (сглаживание, доверительные интервалы) или картограмму

Функции различных графических систем друг с другом не сочетаются

Инструменты предварительного анализа данных и построения линейных моделей

- графические системы в R: `base`, `lattice` и `ggplot2`
- очистка данных
- корреляционный анализ и линейные регрессионные модели

Как не потратить вечность на очистку сырых данных?

- стремимся к опрятным (tidy) данным
- пакет `dplyr` для манипуляций с данными, tibble-таблицы
- пакет `data.table` и специальные выражения в операторе `[]`
- очистка текстовых значений с помощью `gsub()`

Опрятные (tidy) данные

1. Каждая переменная формирует столбец.
2. Каждое наблюдение формирует строку.
3. Каждый тип единицы наблюдения формирует таблицу.
4. Компактные (и, желательно, понятные) названия столбцов.
5. Наличие справочника к данным.

Пакет dplyr

- реализует грамматику обработки данных
- таблицы – подлежащие
- функции-сказуемые: `filter()`, `select()`, `mutate()`, `summarize()` и др.
- каналы `%>%` уменьшают объём кода

Пример использования функций dplyr

```
library('dplyr')

# отфильтровать таблицу по автоматической коробке передач
filter(mtcars, am == 1) %>%
  # выбрать только нужные столбцы
  select(hp, mpg, cyl) %>%
  # сгруппировать строки по показателю cyl (число цилиндров)
  group_by(cyl) %>%
  # посчитать среднюю мощность и число миль на галоне топлива
  summarise(hp = mean(hp), mpg = mean(mpg)) -> df.01
df.01
```

```
## # A tibble: 3 x 3
##     cyl    hp   mpg
##   <dbl> <dbl> <dbl>
## 1     4  81.9  28.1
## 2     6 132.   20.6
## 3     8 300.   15.4
```

Пример использования объектов data.table

```
library('data.table')
# создаём таблицу данных из фрейма mtcars
dt.02 <- data.table(mtcars)
# создаём столбец на основе существующего
dt.02[, Число.цилиндров := cyl]
# убираем исходный столбец
dt.02[, cyl := NULL]
# проверяем результат
colnames(dt.02)
```

```
##  [1] "mpg"              "disp"               "hp"
##  [4] "drat"             "wt"                 "qsec"
##  [7] "vs"               "am"                 "gear"
## [10] "carb"             "Число.цилиндров"
```

Пример использования объектов data.table

```
# посчитать средние по трём количественным столбцам,
#   предварительно разделив на группы
#   по типу коробки передач
# сохраняем имена нужных столбцов в вектор
cols <- c('am', 'mpg', 'disp', 'hp')
# отбираем столбцы и применяем функцию
#   расчёта среднего по группам
dt.02[, ..cols][, lapply(.SD, mean), by = am]
```

```
##    am      mpg     disp       hp
## 1:  1 24.39231 143.5308 126.8462
## 2:  0 17.14737 290.3789 160.2632
```

Поиск и замена подстрок в символьных векторах

- `grep(<что_ищем>', <где_ищем>')` – функция просматривает символьный вектор *<где ищем>* и возвращает номера тех элементов, в которых встречается подстрока *<что ищем>*.

- `gsub(<что_ищем>', <на**что**заменяем>', <где_ищем>')` – ищет и заменяет все вхождения подстроки в векторе.

В шаблоне поиска и замены можно использовать регулярные выражения.

Пример использования grep()

```
# ищем в заголовках строк таблицы mtcars
#   названия моделей автомобилей,
#   которые содержат шаблон "Merc"
# сами названия
grep('Merc', rownames(mtcars), value = T)

# позиции в векторе
grep('Merc', rownames(mtcars))
```

```
## [1] "Merc 240D"   "Merc 230"    "Merc 280"    "Merc 280C"
## [5] "Merc 450SE"  "Merc 450SL"  "Merc 450SLC"

## [1]  8  9 10 11 12 13 14
```

Пример использования gsub()

```
# заменяем "Merc" на "Мерседес"
gsub('Merc', 'Мерседес', rownames(mtcars))
```

```
##  [1] "Mazda RX4"           "Mazda RX4 Wag"
##  [3] "Datsun 710"          "Hornet 4 Drive"
##  [5] "Hornet Sportabout"   "Valiant"
##  [7] "Duster 360"          "Мерседес 240D"
##  [9] "Мерседес 230"        "Мерседес 280"
## [11] "Мерседес 280C"       "Мерседес 450SE"
## [13] "Мерседес 450SL"      "Мерседес 450SLC"
## [15] "Cadillac Fleetwood"  "Lincoln Continental"
## [17] "Chrysler Imperial"   "Fiat 128"
## [19] "Honda Civic"         "Toyota Corolla"
## [21] "Toyota Corona"       "Dodge Challenger"
## [23] "AMC Javelin"         "Camaro Z28"
## [25] "Pontiac Firebird"    "Fiat X1-9"
## [27] "Porsche 914-2"       "Lotus Europa"
## [29] "Ford Pantera L"      "Ferrari Dino"
## [31] "Maserati Bora"       "Volvo 142E"
```

Пример использования gsub()

```
# разделяем название производителя и модели
#   заменить всё после первого пробела на пустую строку (т.е. удалить)
gsub(' .*', '', rownames(mtcars))
```

```
##  [1] "Mazda"    "Mazda"    "Datsun"   "Hornet"   "Hornet"   "Valiant"
##  [7] "Duster"   "Merc"     "Merc"     "Merc"     "Merc"     "Merc"
## [13] "Merc"     "Merc"     "Cadillac" "Lincoln"  "Chrysler" "Fiat"
## [19] "Honda"    "Toyota"   "Toyota"   "Dodge"    "AMC"      "Camaro"
## [25] "Pontiac"  "Fiat"     "Porsche"  "Lotus"    "Ford"     "Ferrari"
## [31] "Maserati" "Volvo"
```

```
#   удалить всё, что идёт до первого пробела
gsub('.+? (.*)', '\\1', rownames(mtcars))
```

```
##  [1] "RX4"          "RX4 Wag"     "710"         "4 Drive"     "Sportabout"
##  [6] "Valiant"      "360"         "240D"        "230"         "280"
## [11] "280C"         "450SE"       "450SL"       "450SLC"      "Fleetwood"
## [16] "Continental" "Imperial"    "128"         "Civic"       "Corolla"
## [21] "Corona"       "Challenger"  "Javelin"     "Z28"         "Firebird"
## [26] "X1-9"         "914-2"       "Europa"      "Pantera L"   "Dino"
## [31] "Bora"         "142E"
```

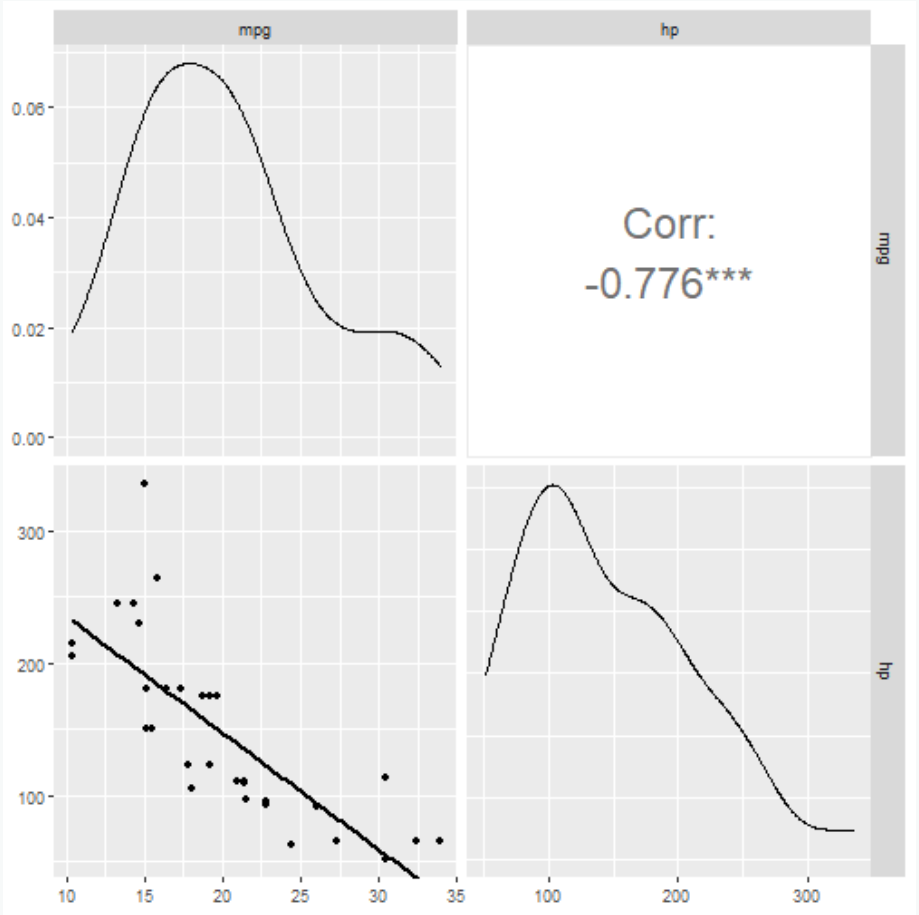
Инструменты предварительного анализа данных и построения линейных моделей

- графические системы в R: `base`, `lattice` и `ggplot2`
- очистка данных

- корреляционный анализ и линейные регрессионные модели

Пример корреляционного анализа

```
library('GGally')
# делаем набор данных
#  из таблицы mtcars
cols <- c('mpg', 'hp')
dt.03 <- data.table(mtcars)[, ..cols]
# графики разброса +
#  коэффициенты корреляции
ggpairs(dt.03,
 lower = list(continuous = 'smooth')))
```



Пример модели регрессии

```
fit <- lm(hp ~ mpg, data = dt.03)
summary(fit)
```

```
##
## Call:
## lm(formula = hp ~ mpg, data = dt.03)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -59.26 -28.93 -13.45  25.65 143.36
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   324.08      27.43  11.813 8.25e-13 ***
## mpg            -8.83       1.31  -6.742 1.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.95 on 30 degrees of freedom
## Multiple R-squared:  0.6024,    Adjusted R-squared:  0.5892
## F-statistic: 45.46 on 1 and 30 DF,  p-value: 1.788e-07
```

Лабораторная работа №2

Подробнее о базовой графике в R:

- Глава 3 учебного пособия "Введение в язык статистической обработки данных R", практические примеры.

Лабораторная работа №2:

- Пример №1: описательные статистики и графики на данных по импорту товаров, связанных с железнодорожным транспортом, в Уругвай, в 2019 году
- Пример №2: предварительный анализ и построение линейных регрессий на показателях, связанных с рейтингом лёгкости ведения бизнеса в странах с высоким и средне-высоким доходом в 2019 году