

Программирование и статистический анализ данных на языке R

Лекция 4 (Основы статистического
анализа на языке R)



Петровский Михаил (ВМК МГУ), michael@cs.msu.su

Рассматриваемые модели

Предиктор Отклик	Категориальный	Непрерывный	Непрерывный и категориальный
Непрерывный	Дисперсионный анализ (ANOVA)	Регрессия наименьших квадратов (OLS Regression)	Ковариационный анализ (ANCOVA)
Категориальный	Логистическая регрессия	Логистическая регрессия	Логистическая регрессия

Линейная корреляция Пирсона

- Коэф. корреляция Пирсона:

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

- Проверяется гипотеза о равенстве 0 коэф. корр. по распределению стьюдента с $n-2$ степенями свободы:

$$t = (n - 2)^{1/2} \left(\frac{r^2}{1 - r^2} \right)^{1/2}$$

- Недостатки: линейность, неустойчивость к выбросам
- Любая корреляция не есть зависимость и причинно-следственная связь!

Другие важные корреляции

- Ранговая корреляция (Спирмана), где R_i и S_i - ранги наблюдений:

$$\theta = \frac{\sum_i ((R_i - \bar{R})(S_i - \bar{S}))}{\sqrt{\sum_i (R_i - \bar{R})^2} \sqrt{\sum_i (S_i - \bar{S})^2}}$$

- Корреляция Кендалла на основе согласованности пар наблюдений из разных выборок (вместе растут или вместе убывают с учетом «ничьих»):

$$\tau = \frac{\sum_{i < j} (\text{sgn}(x_i - x_j)\text{sgn}(y_i - y_j))}{\sqrt{(T_0 - T_1)(T_0 - T_2)}}$$

где $T_0 = n(n - 1)/2$, $T_1 = \sum_k t_k(t_k - 1)/2$, $T_2 = \sum_l u_l(u_l - 1)/2$.

t_k - число ничьих в k -группе ничьих в первой выборке, u_l -аналогично для второй выборки

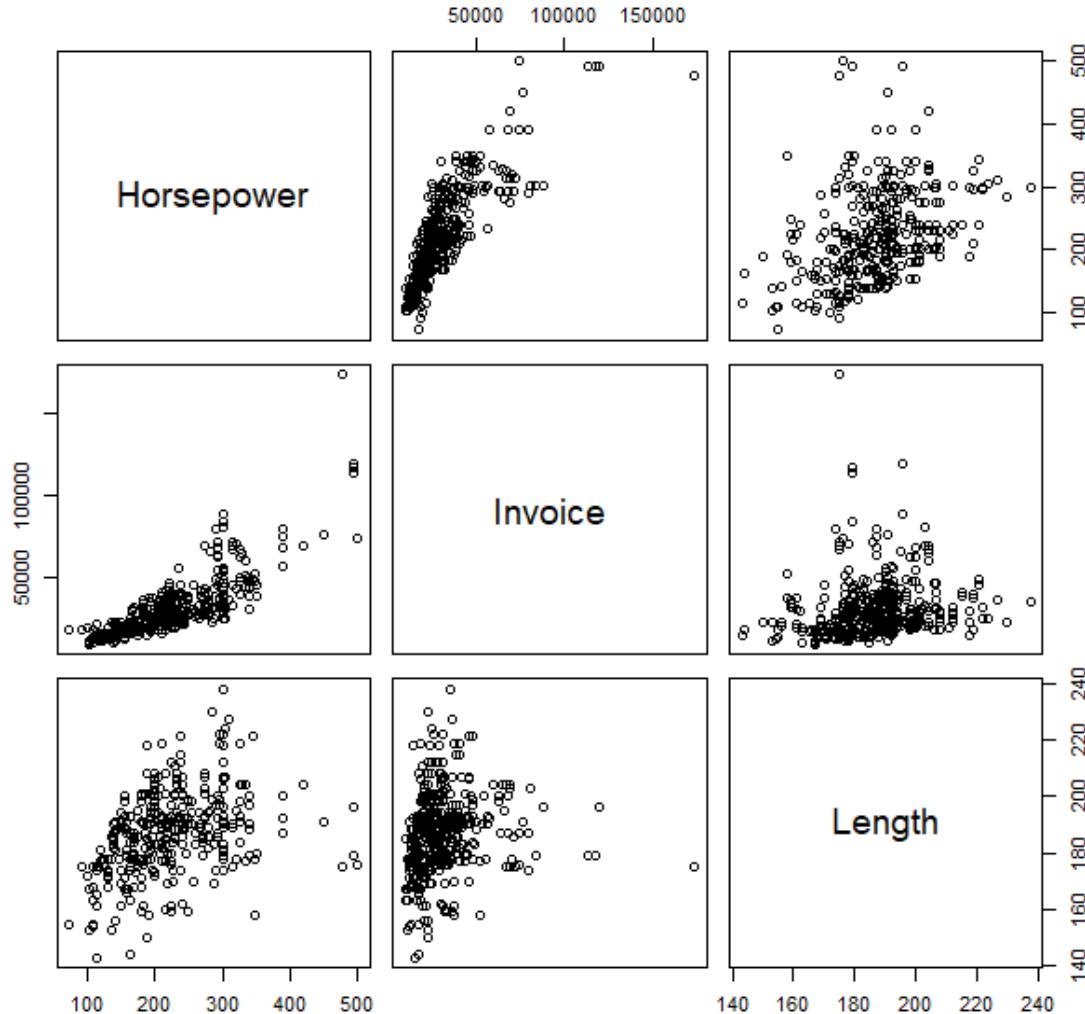
Расчет корреляций

```
> cor.test(cars$Horsepower, cars$Invoice,
+ method = "pearson")  
  
Pearson's product-moment correlation  
  
data: cars$Horsepower and cars$Invoice  
t = 29.988, df = 426, p-value < 2.2e-16  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
0.7906976 0.8520082  
sample estimates:  
cor  
0.8237465  
  
> cor.test(cars$Horsepower, cars$Invoice,
+ method = "kendall")  
  
Kendall's rank correlation tau  
  
data: cars$Horsepower and cars$Invoice  
z = 20.906, p-value < 2.2e-16  
alternative hypothesis: true tau is not equal to 0  
sample estimates:  
tau  
0.6812691  
  
> corr_res <- corr.test(cars[, c("Horsepower", "Invoice", "Length")],
+ method = "pearson", use = "complete")
> corr_res$p
      Horsepower      Invoice       Length
Horsepower 0.000000e+00 1.447446e-106 5.576157e-16
Invoice    4.824821e-107 0.000000e+00 5.391787e-04
Length     2.788079e-16  5.391787e-04 0.000000e+00
> corr_res$r
      Horsepower      Invoice       Length
Horsepower 1.0000000 0.8237465 0.3815539
Invoice    0.8237465 1.0000000 0.1665864
Length     0.3815539 0.1665864 1.0000000
```

Графики и вывод результатов

- Можно строить графики попарной корреляции

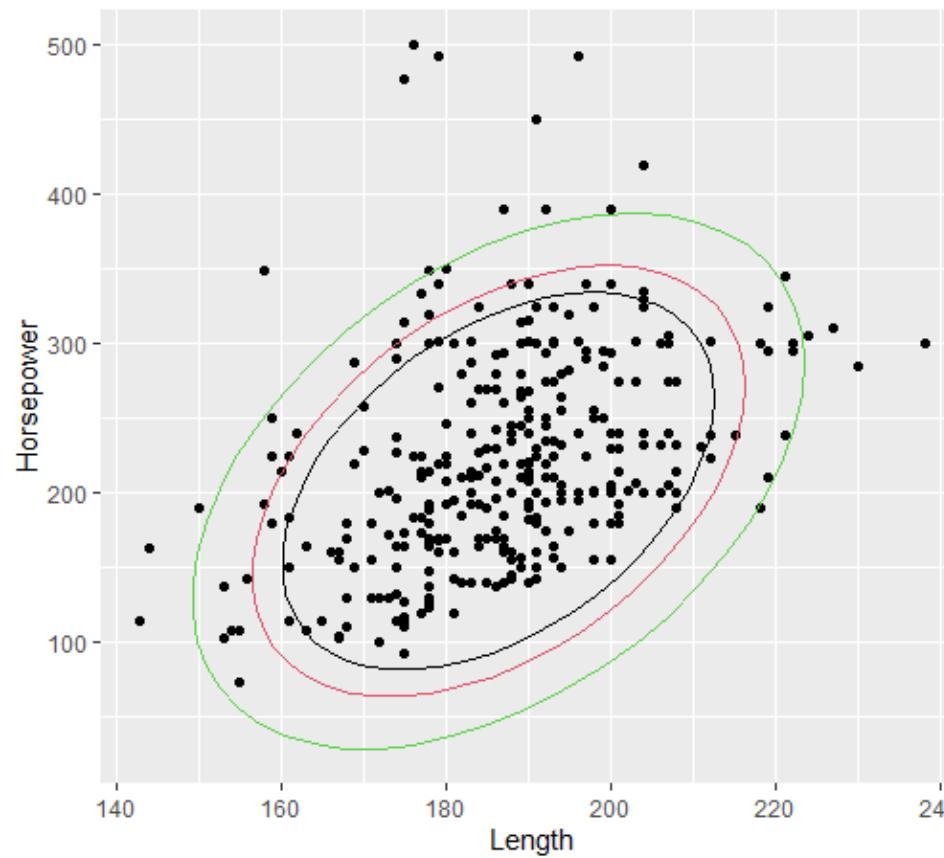
```
> pairs(cars[, c("Horsepower", "Invoice", "Length")])
```



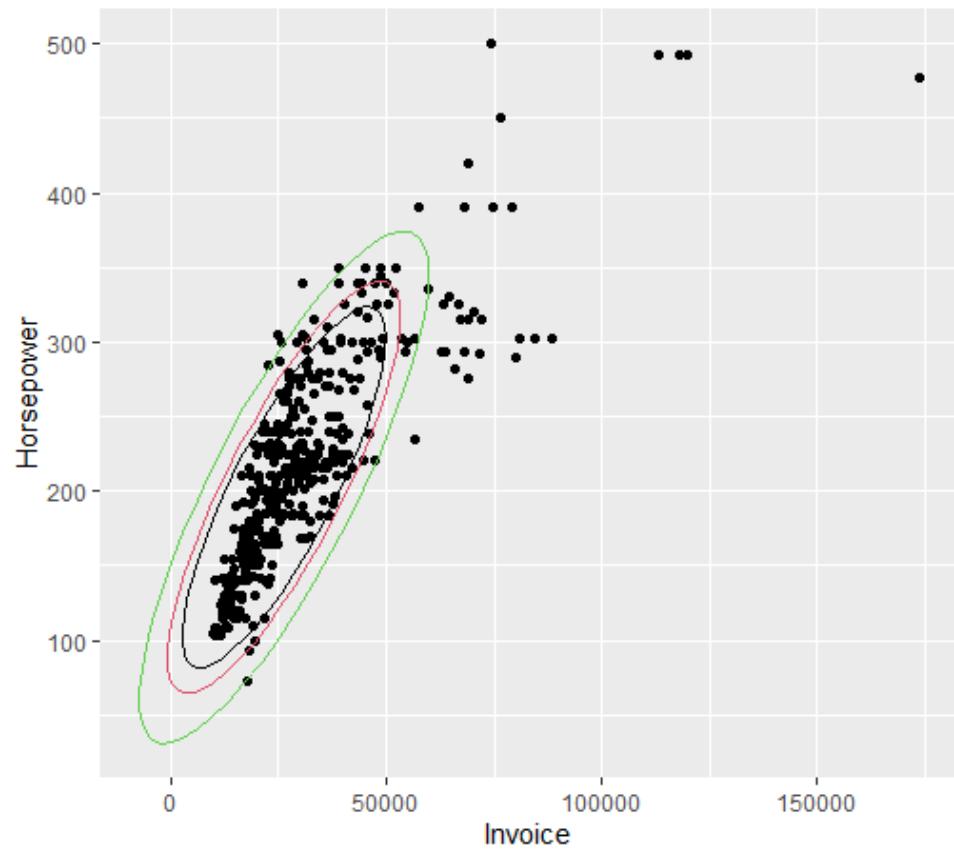
Графики и вывод результатов

- Можно строить графики разброса с эллипсоидом рассеивания

```
> ggplot(cars, aes(x = Length, y = Horsepower)) +  
+   geom_point() +  
+   stat_ellipse(level = 0.9) +  
+   stat_ellipse(level = 0.95, color = 2) +  
+   stat_ellipse(level = 0.99, color = 3)
```



```
> ggplot(cars, aes(x = Invoice, y = Horsepower)) +  
+   geom_point() +  
+   stat_ellipse(level = 0.9) +  
+   stat_ellipse(level = 0.95, color = 2) +  
+   stat_ellipse(level = 0.99, color = 3)
```



Пример с частичной корреляцией

```
> describe(cars[,c("Invoice", "Horsepower", "Length", "EngineSize")])  
      vars   n     mean       sd median trimmed      mad     min     max  
Invoice    1 428 30014.70 17642.12 25294.5 27187.34 11165.46 9875.0 173560.0  
Horsepower  2 428    215.89    71.84   210.0   210.83    66.72    73.0    500.0  
Length      3 428    186.36    14.36   187.0   186.16    13.34   143.0    238.0  
EngineSize   4 428      3.20     1.11     3.0     3.12     1.19     1.3     8.3
```

```
> pcor.test(cars$Horsepower, cars$Invoice, cars$EngineSize, method = "pearson")  
  estimate      p.value statistic   n gp Method  
1 0.7453697 7.435217e-77  23.04981 428  1 pearson  
> pcor.test(cars$Horsepower, cars$Invoice, cars$EngineSize, method = "spearman")  
  estimate      p.value statistic   n gp Method  
1 0.7429525 4.137485e-76  22.88263 428  1 spearman  
> pcor.test(cars$Horsepower, cars$Invoice, cars$EngineSize, method = "kendall")  
  estimate      p.value statistic   n gp Method  
1 0.5504002 9.65072e-65  16.99053 428  1 kendall
```

```
> pcor.test(cars$Horsepower, cars$Invoice, cars$Length, method = "pearson")  
  estimate      p.value statistic   n gp Method  
1 0.8340567 7.629278e-112  31.16764 428  1 pearson  
> pcor.test(cars$Horsepower, cars$Invoice, cars$Length, method = "spearman")  
  estimate      p.value statistic   n gp Method  
1 0.8515963 2.769776e-121  33.49136 428  1 spearman  
> pcor.test(cars$Horsepower, cars$Invoice, cars$Length, method = "kendall")  
  estimate      p.value statistic   n gp Method  
1 0.6625207 5.810566e-93  20.45163 428  1 kendall
```

Регрессионный анализ

- Задача регрессии:

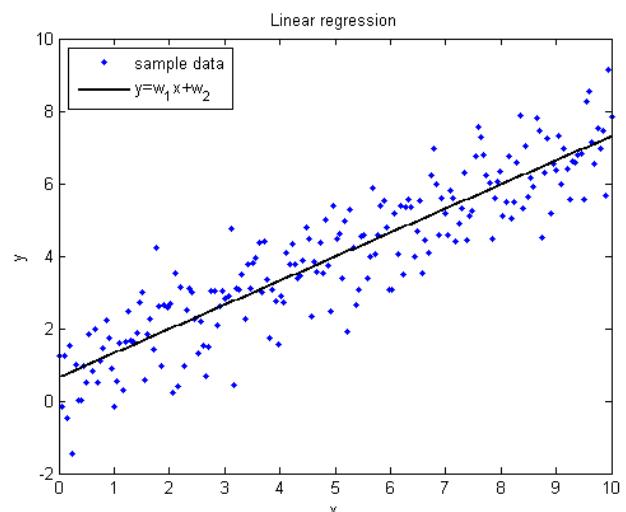
$$y(x_1, \dots, x_p) = E(Y | X_1 = x_1, \dots, X_p = x_p)$$

- Уравнение линейной регрессии:

- $$f(X) = b_0 + \sum_{j=1}^p X_j b_j + \varepsilon$$
- $\varepsilon = N(0, \sigma^2)$ - шум
 - Y – отклик (критериальная переменная)
 - $X = (X_1, \dots, X_p)$ - регрессоры (предикторы, факторы), b – параметры модели

- Линеаризуемые регрессии:

- Степенная
- Экспоненциальная
- Гиперболическая
- и другие



$$y = ax_1^{b_1} x_2^{b_2} \dots x_p^{b_p} \varepsilon,$$

$$y = e^{a+b_1x_1+b_2x_2+\dots+b_px_p+\varepsilon},$$

$$y = (a + b_1x_1 + b_2x_2 + \dots + b_px_p + \varepsilon)^{-1}$$

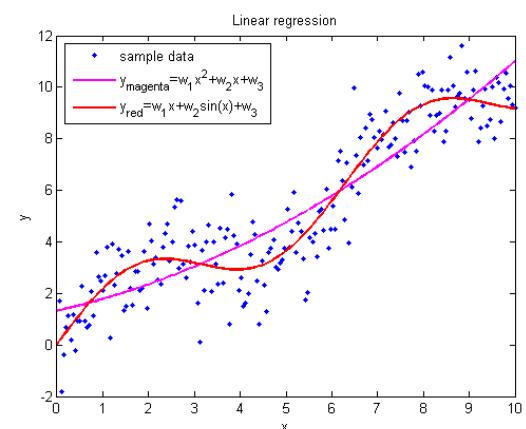
Регрессионный анализ

- Цель регрессионного анализа:
 - Определение наличия связи между переменными и характера этой связи (т. е. нахождение описывающего её математического уравнения)
 - Определение степени вариации целевой переменной предикторами
 - Предсказание значения зависимой переменной с помощью независимой
 - Определение вклада отдельных независимых переменных в вариацию зависимой
- Задача «обучения с учителем»:
 - Тренировочный набор из N векторов:
 - Искомая модель – уравнение регрессии

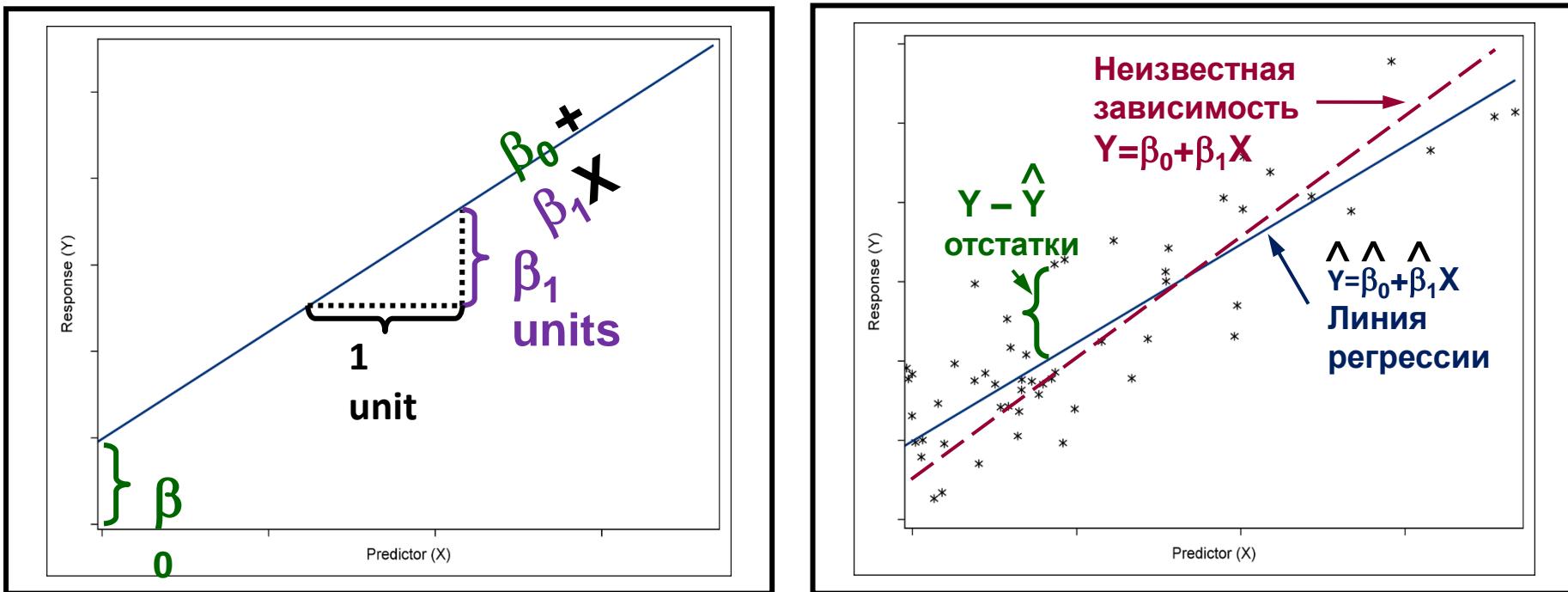
$$Z = \{(\bar{x}_i, y_i)\}_1^N$$

Предположения

- Независимость наблюдений
- Выбранное уравнение регрессии (например, линейное) соответствует истинной зависимости в данных
- Нормальность ошибки (с константной дисперсией по всем наблюдениям)



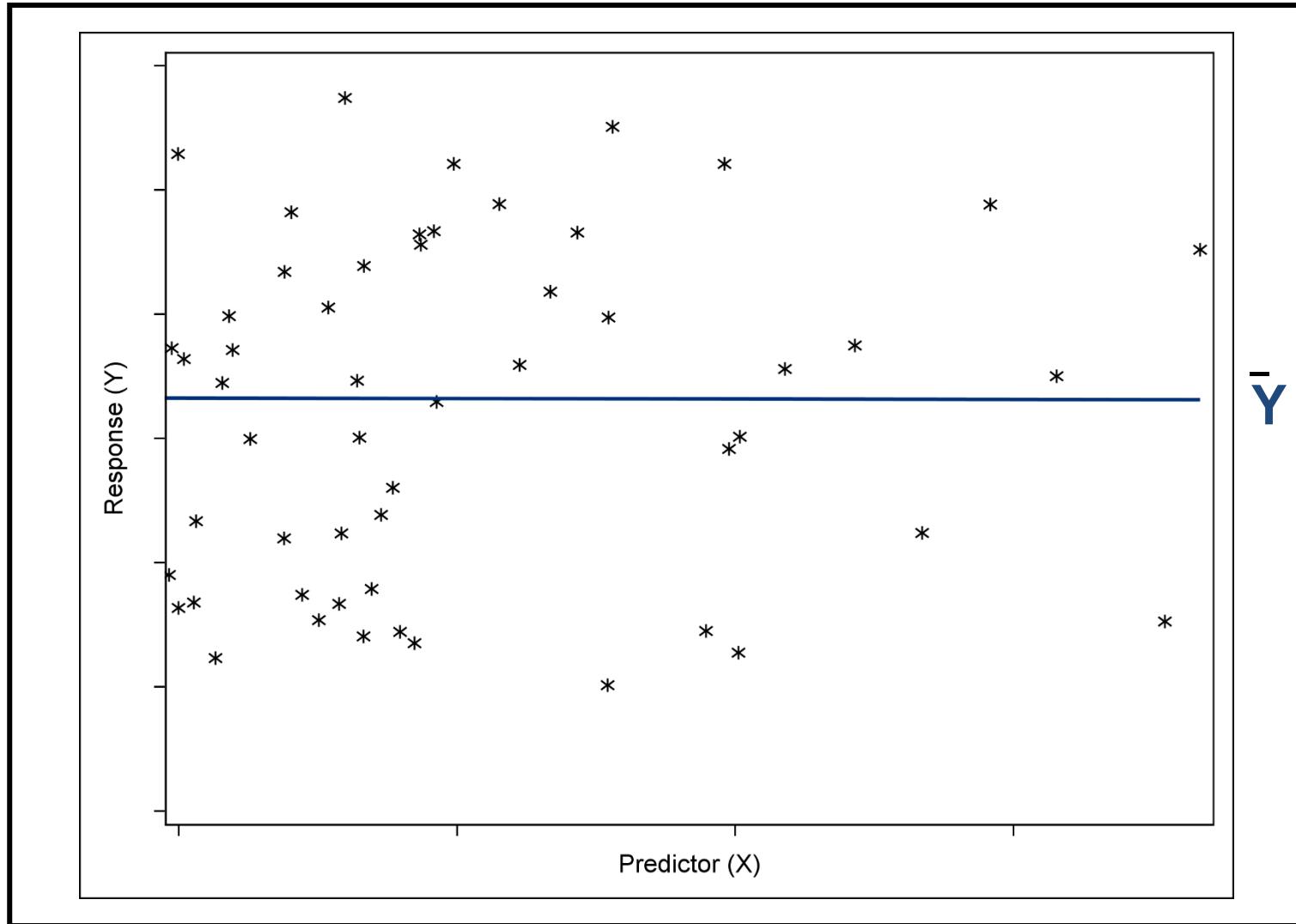
Простая линейная регрессия



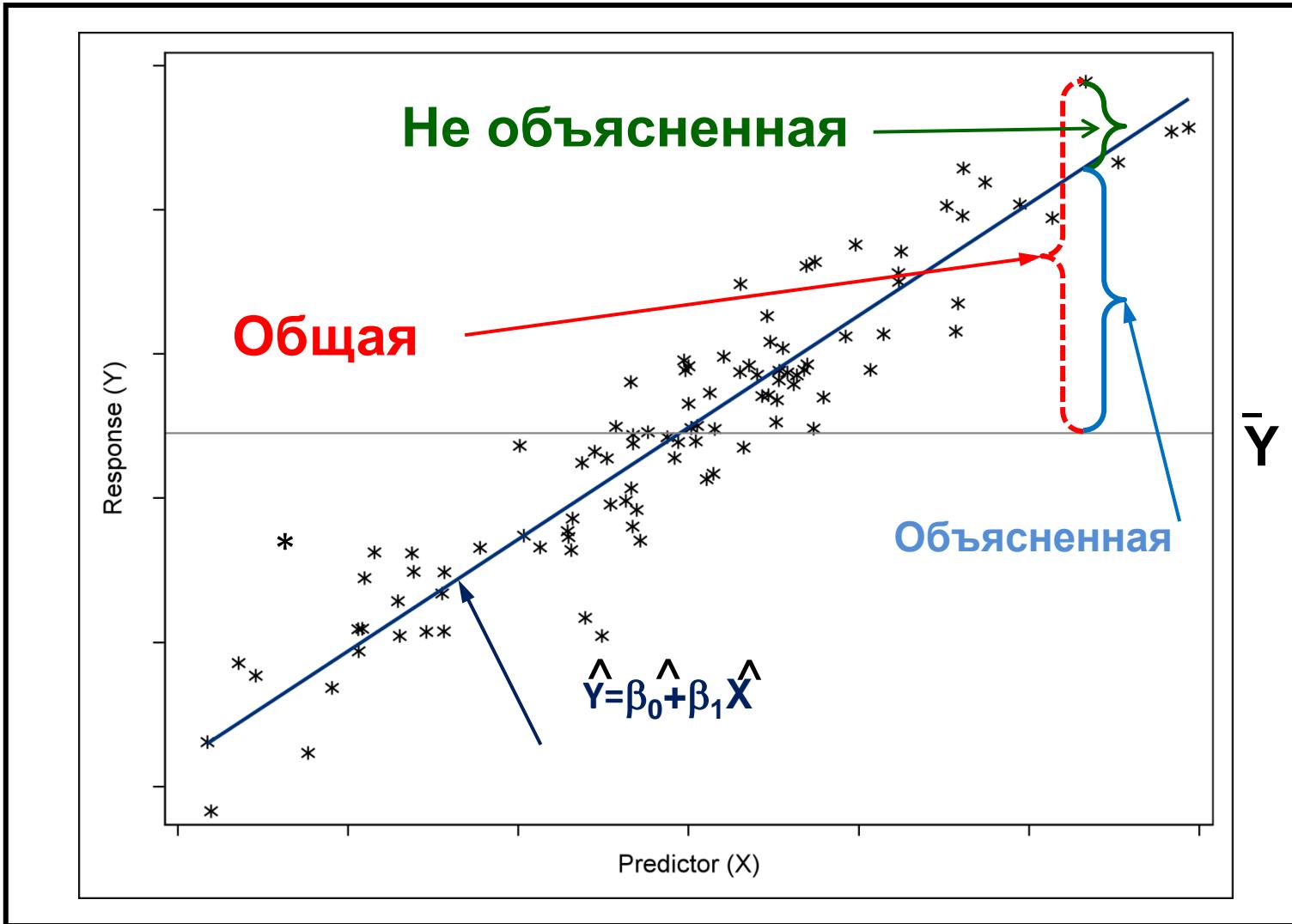
Цели регрессионного анализа:

- Оценить значимость предикторов с точки зрения влияния на вариацию отклика
- Предсказать значение отклика для заданного предиктора

Базовая модель (Нулевая гипотеза)



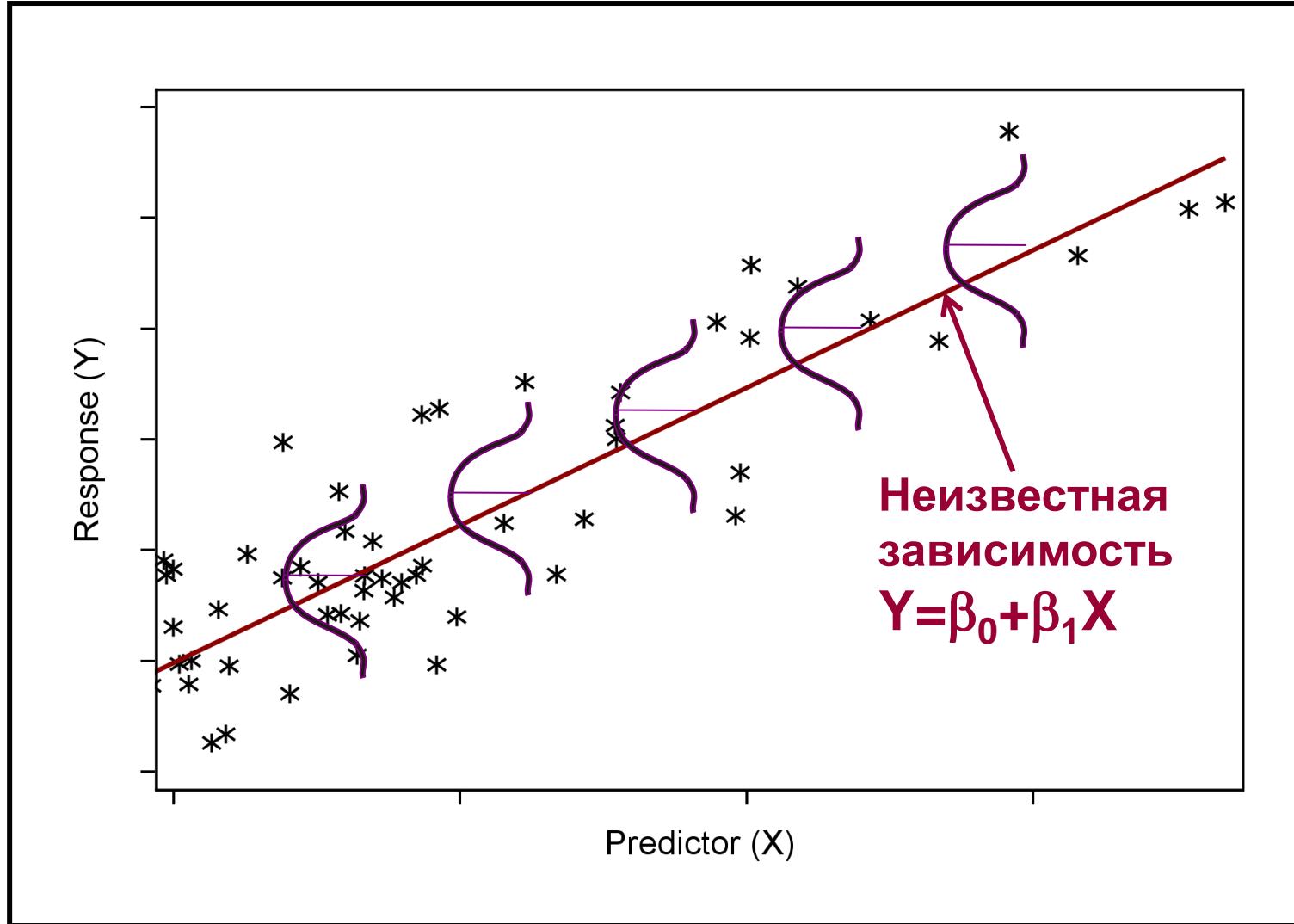
Объясненная и необъясненная вариация



Проверяемая гипотеза

- **Нулевая:**
 - Регрессионная модель приближает наблюдаемые данные **не** лучше базовой модели - константы.
 - $\beta_1=0$
- **Альтернативная:**
 - Регрессионная модель лучше приближает наблюдаемые данные чем базовая модель – константа.
 - $\beta_1 \neq 0$

Предположения линейной регрессии



Метод наименьших квадратов и проблема мультиколлинеарности

- Оценка ошибки = сумма регрессионных остатков (квадратичная функция потерь):

$$RSS(B) = \sum_{i=1}^N (y_i - f(\bar{x}_i))^2 = \sum_{i=1}^N (y_i - b_0 - \sum_{j=1}^p x_{ij} b_j)^2$$

- В матричной форме:

$$RSS(B) = (\bar{y} - XB)^T (\bar{y} - XB)$$

- Единственное оптимальное решение (если матрица данных не сингулярная)

- Недостатки: $B = (X^T X)^{-1} X^T \bar{y}$

- Сингулярная матрица данных из-за коррелированных факторов
 - Большое число регрессоров – плохая точность и интерпретируемость

- Основные подходы:

- Поиск и удаление зависимых и незначимых факторов
 - Использование «смещенных» регуляризованных моделей
 - переход к новым независимым факторам, например, с помощью метода главных компонент

Простая регрессия

```
> aov_model <- aov(Horsepower ~ EngineSize, cars)
> lm_model <- lm(Horsepower ~ EngineSize, cars)
>
> summary(aov_model)
      Df  Sum Sq Mean Sq F value Pr(>F)
EngineSize     1 1366287 1366287   695.2 <2e-16 ***
Residuals   426  837210    1965
>
> mean(lm_model$fitted)
[1] 215.8855
> sigma(lm_model)/mean(lm_model$fitted)*100
[1] 20.53473
>
> summary(lm_model)

Call:
lm(formula = Horsepower ~ EngineSize, data = cars)

Residuals:
    Min      1Q  Median      3Q      Max 
-89.743 -27.130 - 6.668 19.153 240.538 

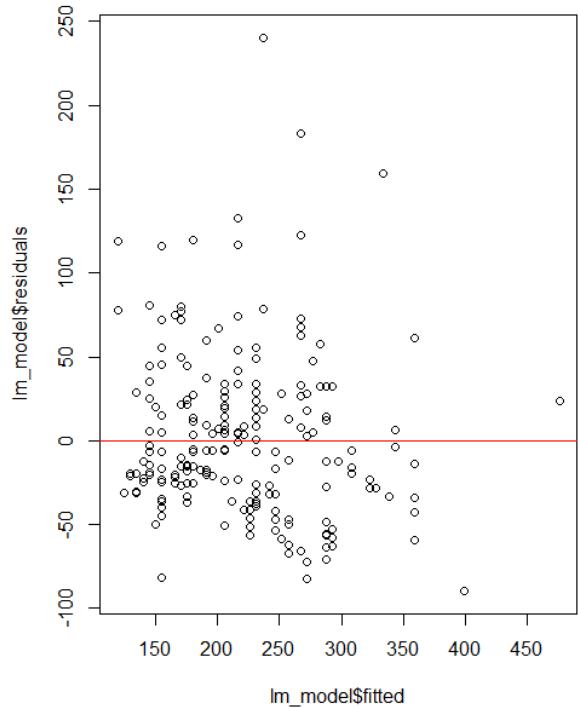
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 52.772     6.547   8.061 7.74e-15 ***
EngineSize   51.025     1.935  26.367 < 2e-16 ***
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1

Residual standard error: 44.33 on 426 degrees of freedom
Multiple R-squared:  0.6201,    Adjusted R-squared:  0.6192 
F-statistic: 695.2 on 1 and 426 DF,  p-value: < 2.2e-16
```

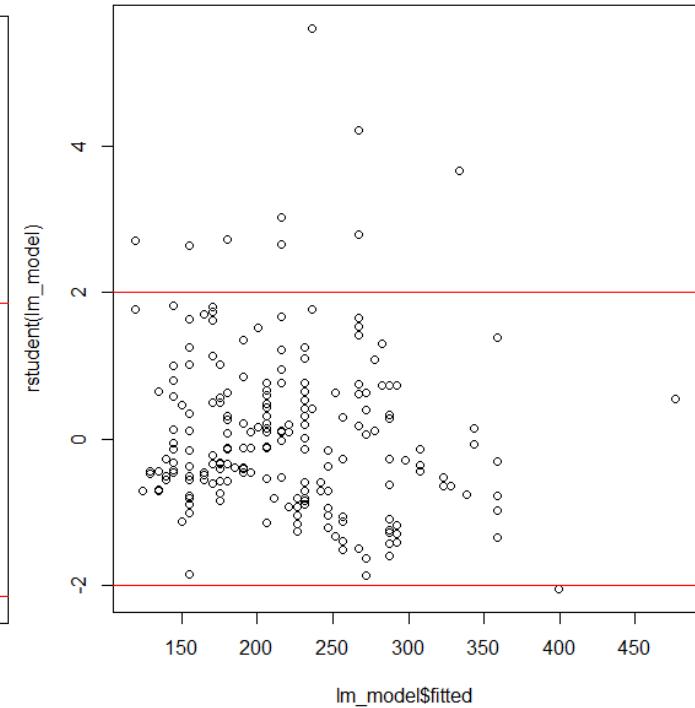
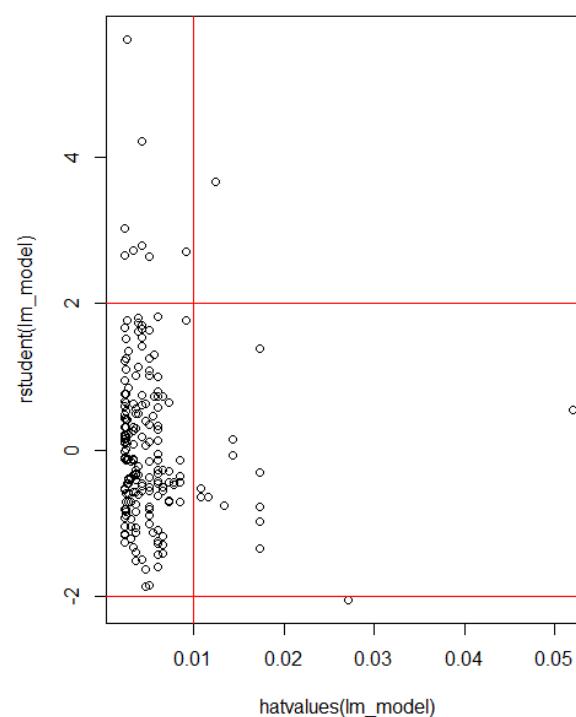
- Сумма квадратов
- Сумма квадратов деленная на DF
- F – отношение MS модели к MS модели ошибки
- P-value для F
- Среднее по отклику
- Отношение MSE к среднему по отклику
- Для каждого параметра модели проверка гипотезы о равенстве 0

Графики регрессионной модели

```
> plot(lm_model$fitted, lm_model$residuals)
> abline(h = 0, col = "red")
```



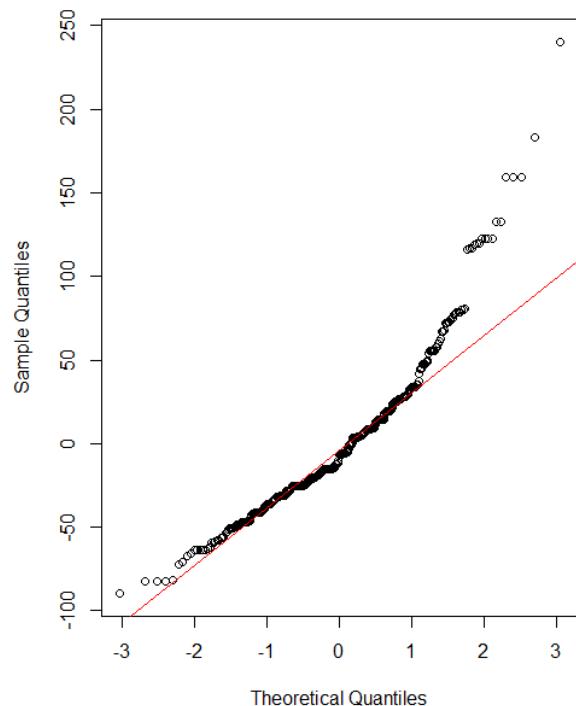
```
> plot(lm_model$fitted, rstudent(lm_model))
> abline(h = 2, col = "red")
> abline(h = -2, col = "red")
```



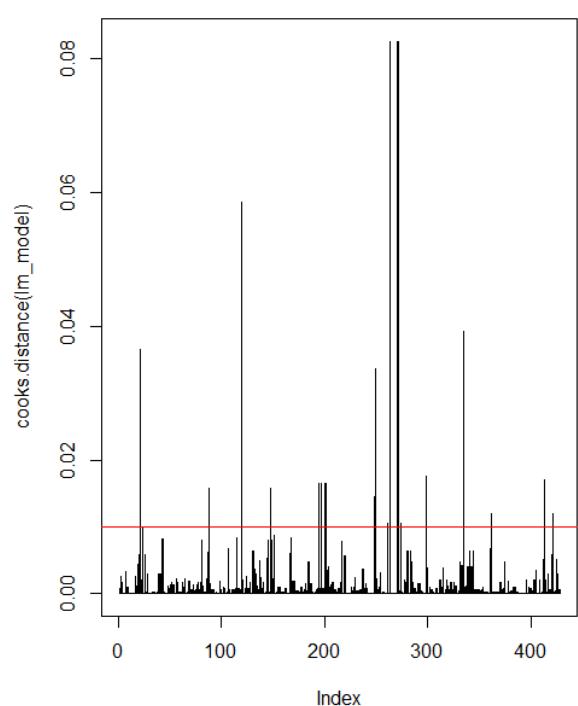
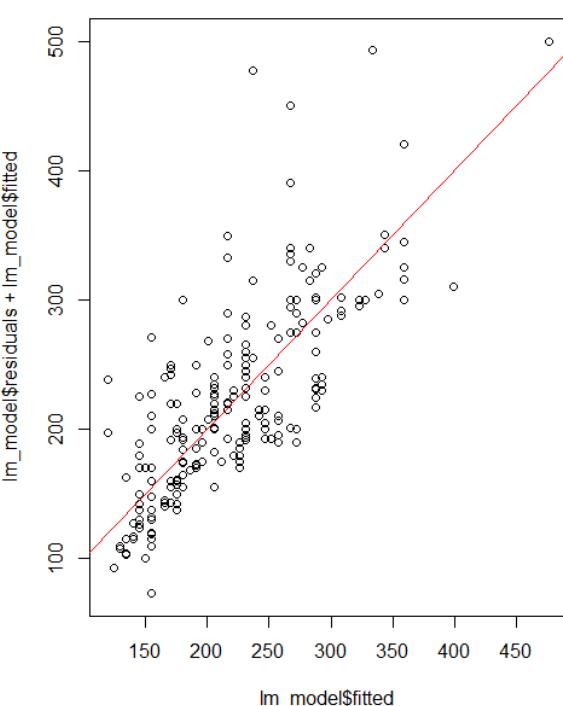
```
> plot(rstudent(lm_model) ~ hatvalues(lm_model))
> abline(h = 2, col = "red")
> abline(h = -2, col = "red")
> abline(v = 0.01, col = "red")
```

Графики регрессионной модели

```
> qqnorm(lm_model$residuals)
> qqline(lm_model$residuals, col
```

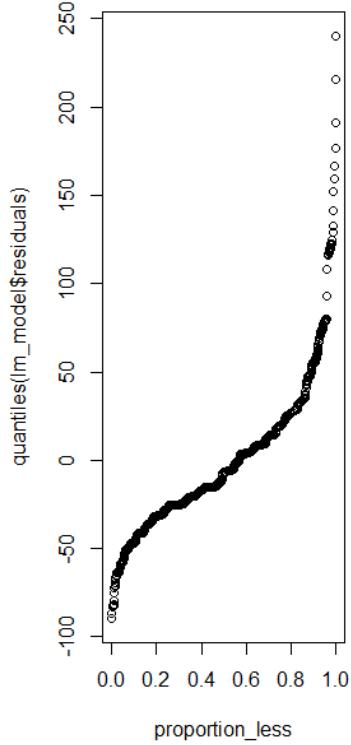
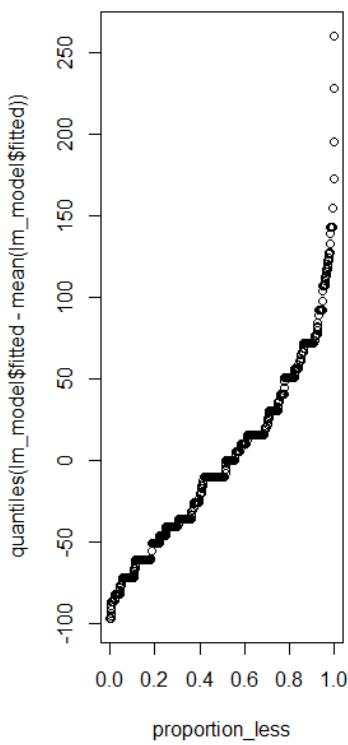


```
> plot(cooks.distance(lm_model), type = 'h')
> abline(h = 0.01, col = "red")
```



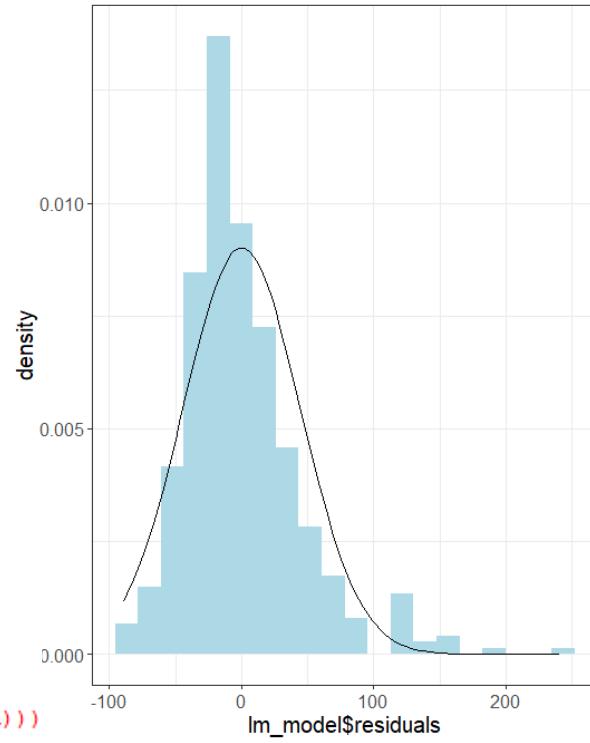
```
> plot(lm_model$fitted, lm_model$residuals + lm_model$fitted)
> abline(a = 0, b = 1, col = "red")
```

Графики регрессионной модели

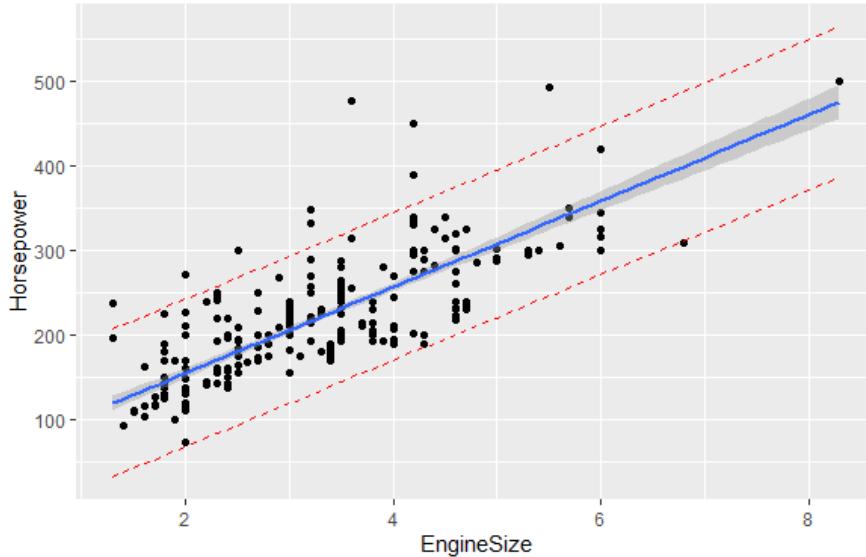


```
> par(mfrow = c(1, 2))
> proportion_less <- seq(0, 1, 0.001)
> quantiles <- function(x) array(quantile(x, probs = proportion_less))
>
> plot(proportion_less, quantiles(lm_model$fitted - mean(lm_model$fitted)))
> plot(proportion_less, quantiles(lm_model$residuals))
```

```
> dnorm_pars <- with(lm_model, c(mean = mean(residuals), sd = sd(residuals)))
>
> ggplot(lm_model, aes(x = lm_model$residuals)) +
+   geom_histogram(aes(y = ..density..), fill="light blue", bins=20) +
+   stat_function(fun = dnorm, args = dnorm_pars) +
+   theme_bw() + theme(text = element_text(size=15))
```



Графики регрессионной модели



```
> model <- lm(Horsepower ~ EngineSize, data = cars)
>
> pred.int <- predict(model, interval = "prediction")
> mydata <- cbind(cars, pred.int)
> p <- ggplot(mydata, aes(EngineSize, Horsepower)) +
+     geom_point() +
+     stat_smooth(method = lm)
>
> p + geom_line(aes(y = lwr), color = "red", linetype = "dashed") +
+     geom_line(aes(y = upr), color = "red", linetype = "dashed")
```

- Доверительный интервал определяет область куда с заданной вероятностью попадет среднее по отклику, т.е. сама регрессия
- Прогнозный интервал определяет область куда с заданной вероятностью попадет отдельное значение отклика

$$b_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\text{cov}(x, y)}{\sigma_x^2};$$

$$b_0 = \bar{y} - b_1 \bar{x};$$

$$s_e^2 = \frac{\sum_{i=1}^n (y_i - \hat{y})^2}{n - 2};$$

$$s_{b_0} = s_e \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}};$$

$$s_{b_1} = s_e \sqrt{\frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2}},$$

$$\hat{s}_{\hat{y}} = s_e \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}};$$

$$\hat{y} - t_{(1-\alpha/2, n-2)} s_{\hat{y}} < y < \hat{y} + t_{(1-\alpha/2, n-2)} s_{\hat{y}},$$

$$s_Y = s_e \sqrt{\frac{1}{m} + \frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}};$$

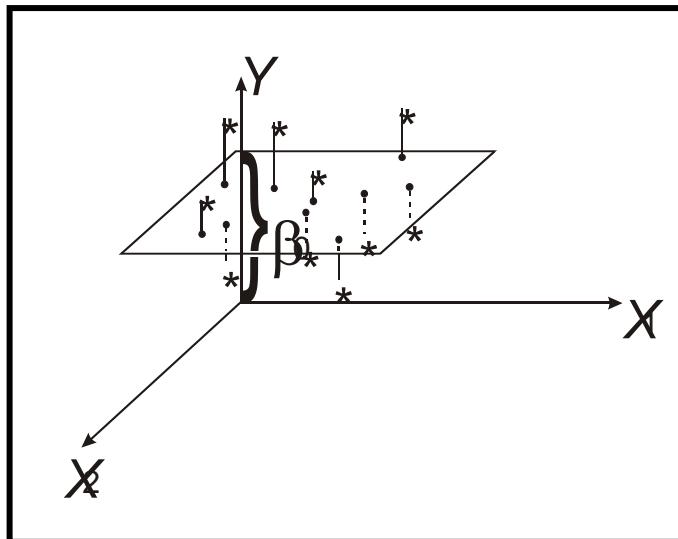
$$\hat{y} - t_{(1-\alpha/2, n-2)} s_Y < y < \hat{y} + t_{(1-\alpha/2, n-2)} s_Y.$$

Множественная линейная регрессия

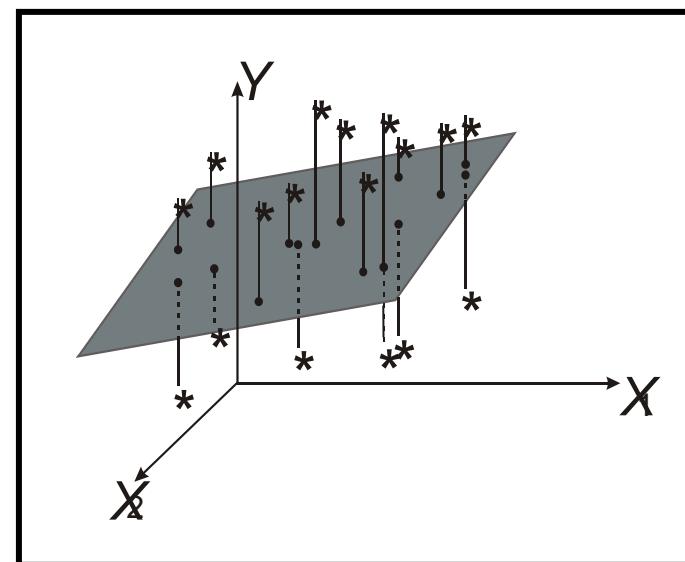
- Пример линейной модели с двумя переменными

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon, \text{ где}$$

Y – отклик, X_1 и X_2 – предикторы, ε – ошибка, β_0 , β_1 , и β_2 – параметры (неизвестные)



Нет зависимости

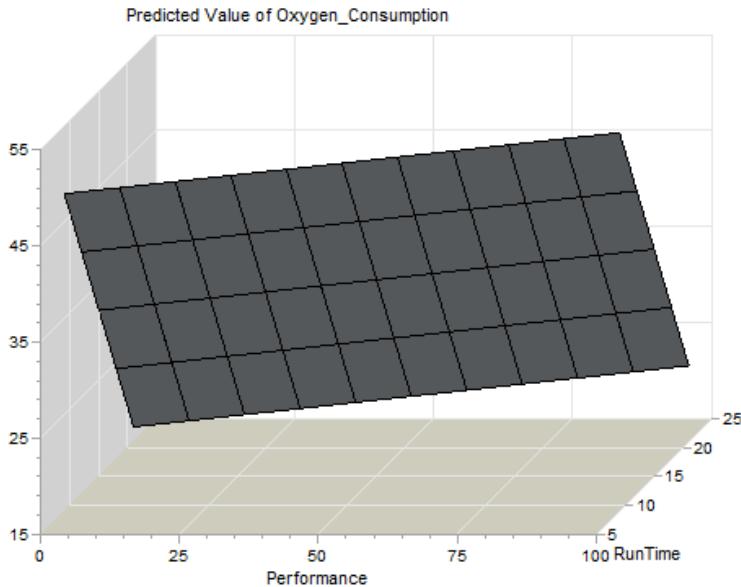


Есть зависимость

Множественная линейная регрессия

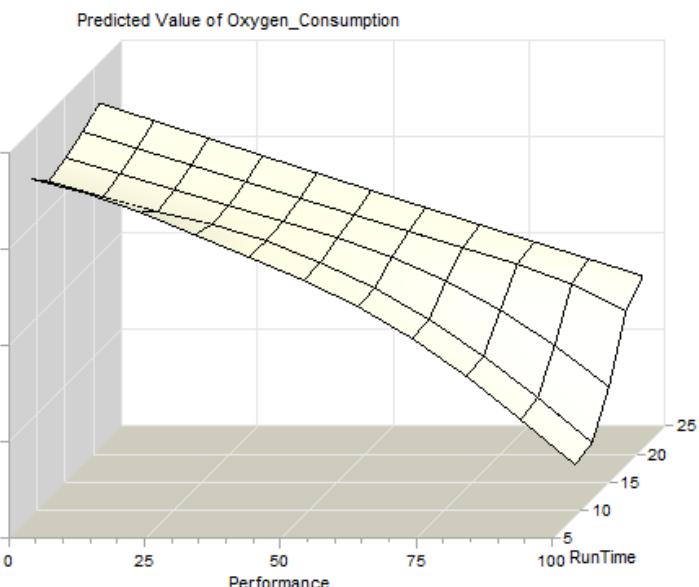
- В общем случае ищем зависимость как линейную комбинацию k предикторов $X_1 - X_k$:

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + \varepsilon$$



$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

Линейная модель с линейными эффектами



$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \varepsilon$$

Линейная модель с нелинейными эффектами

Формулы (для множественной регрессии и не только)

СИМВОЛ	ЗНАЧЕНИЕ
~	Отделяет отклик(и) от предикторов $y \sim x_1 + x_2 + x_3$
+	Отделяет предикторы друг от друга $y \sim x_1 + x_2 + x_3$
:	Задает взаимодействие предикторов $y \sim x_1 + x_2 + x_3 + x_1 : x_2$
*	Задает комбинации взаимодействующих предикторов $y \sim x_1 * x_2 + x_3$ эквивалентно $y \sim x_1 + x_2 + x_1 : x_2 + x_3$
^	Задает порядок взаимодействия предикторов $y \sim (x_1 + x_2 + x_3)^2$ эквивалентно $y \sim x_1 + x_2 + x_3 + x_1 : x_2 + x_1 : x_3 + x_2 : x_3$
.	Включает в модель все переменные источника $y \sim$. Эквивалентно $y \sim x_1 + x_2 + x_3$, если в наборе есть x_1, x_2, x_3
-	Исключает предиктор из формулы переменную или константу (если указано -1) $y \sim (x_1 + x_2 + x_3)^2 - x_2 : x_3$ эквивалентно $y \sim x_1 + x_2 + x_3 + x_1 : x_2 + x_1 : x_3$
I()	Внутри скобок арифметическое выражение $y \sim I((x_1 + x_2 + x_3)^2)$ означает $x_1 + x_2 + x_3$ в квадрате
func	Использование функции func от предикторов или отклика

Проверяемая гипотеза

- **Нулевая:**
 - Регрессионная модель приближает наблюдаемые данные **не** лучше базовой модели - константы.
 - Все $\beta_i = 0$ (всех вместе проверяем по критерию Фишера и каждый отдельно по критерию Стьюдента)
- **Альтернативная:**
 - Регрессионная модель лучше приближает наблюдаемые данные чем базовая модель – константа.
 - Существуют $\beta_i \neq 0$

Множественная линейная регрессия

Предположения множественной линейной регрессии:

- Зависимость условного мат. ожидания отклика от предикторов - линейная
- Ошибка ε из $N(0, \sigma^2)$ с константной дисперсией.
- Ошибки независимы

Применяется для:

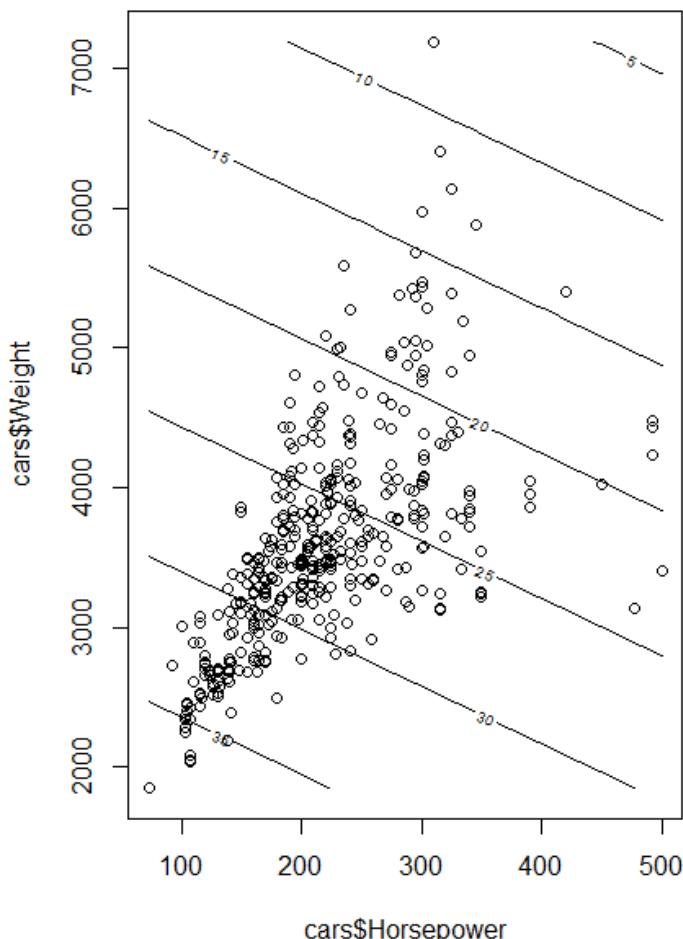
- Прогнозирования – важна не интерпретируемость модели, значимость коэф. и т.д., а точность на тестовом наборе
- Разведочный анализ – важны значения и знаки коэф., уровни значимости и доверительные интервалы, цель – выявить интерпретируемые зависимости в данных

Скорректированная R^2 :

- n – число наблюдений
- p – число параметров
- i – признак, есть ли константа в модели

$$R_{ADJ}^2 = 1 - \frac{(n - i)(1 - R^2)}{n - p}$$

Вывод множественной регрессии



```
> plot(cars$Horsepower, cars$Weight)
> contour(lm_model, Weight ~ Horsepower, add = TRUE)
```

```
> aov_model <- aov(MPG_Highway ~ Horsepower + Weight, cars)
> lm_model <- lm(MPG_Highway ~ Horsepower + Weight, cars)
>
> summary(aov_model)
   Df Sum Sq Mean Sq F value Pr(>F)
Horsepower     1    5895    5895  526.9 <2e-16 ***
Weight         1    3424    3424  306.1 <2e-16 ***
Residuals    425   4755      11
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(lm_model)

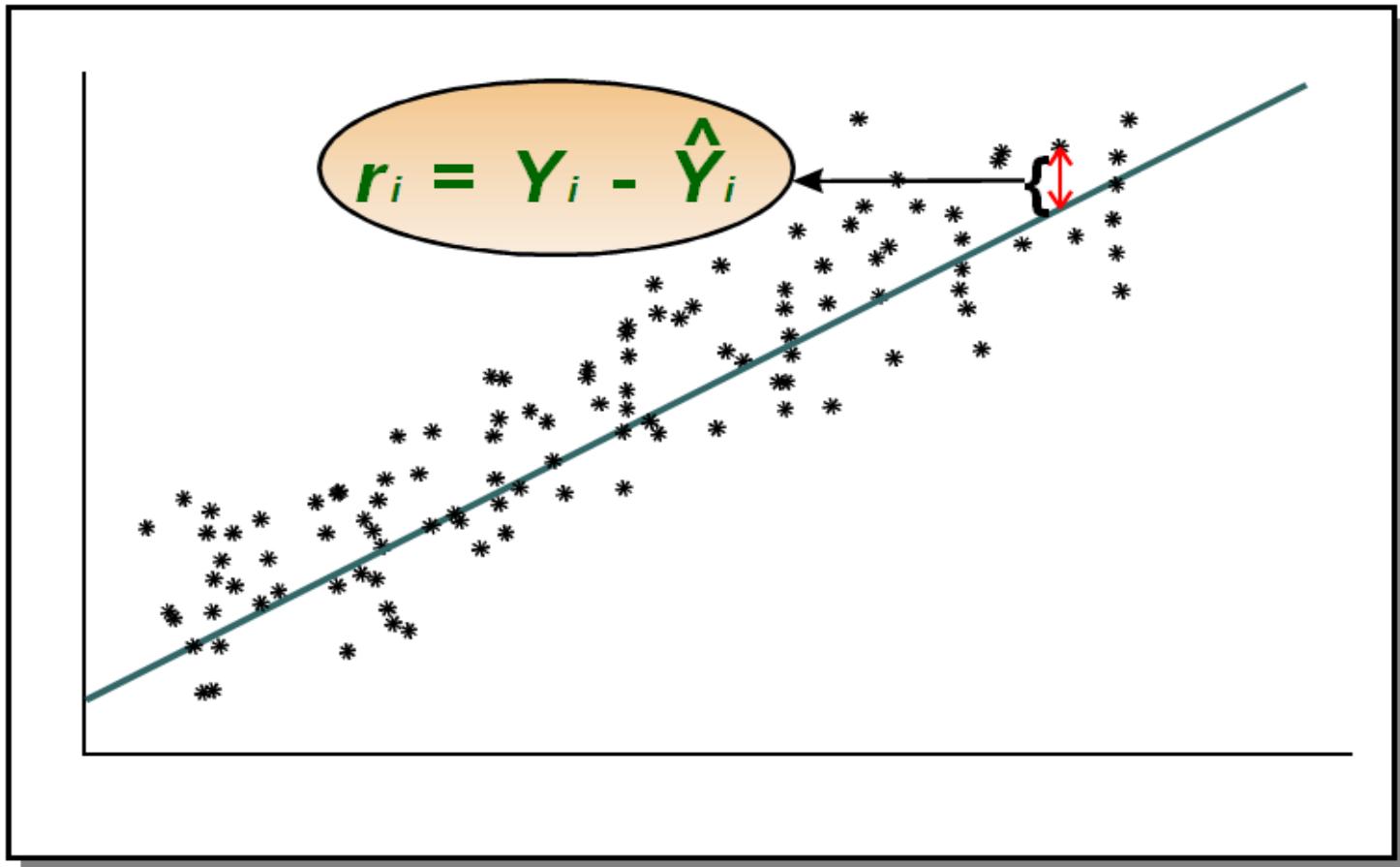
Call:
lm(formula = MPG_Highway ~ Horsepower + Weight, data = cars)

Residuals:
    Min      1Q  Median      3Q      Max 
-9.2682 -1.7478 -0.1446  1.7058 28.0361 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 48.2961097  0.7800378 61.915 < 2e-16 ***
Horsepower  -0.0196774  0.0029039 -6.776 4.13e-11 ***
Weight       -0.0048085  0.0002749 -17.495 < 2e-16 ***
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

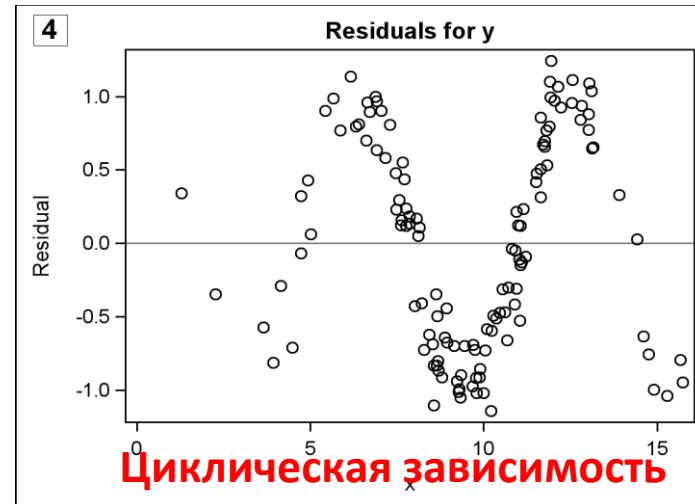
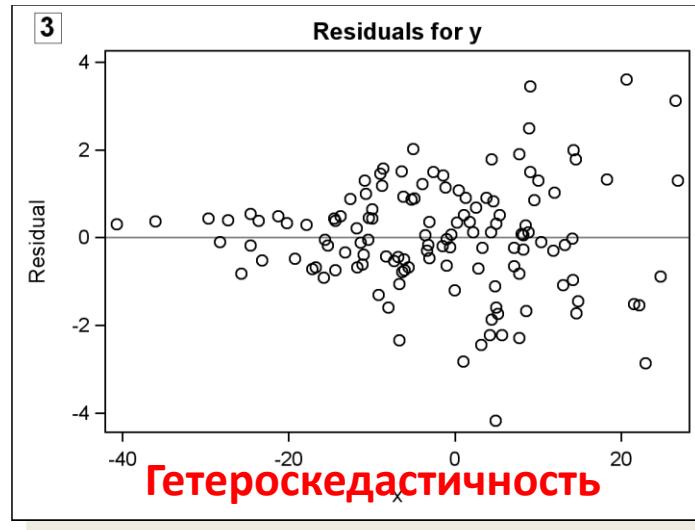
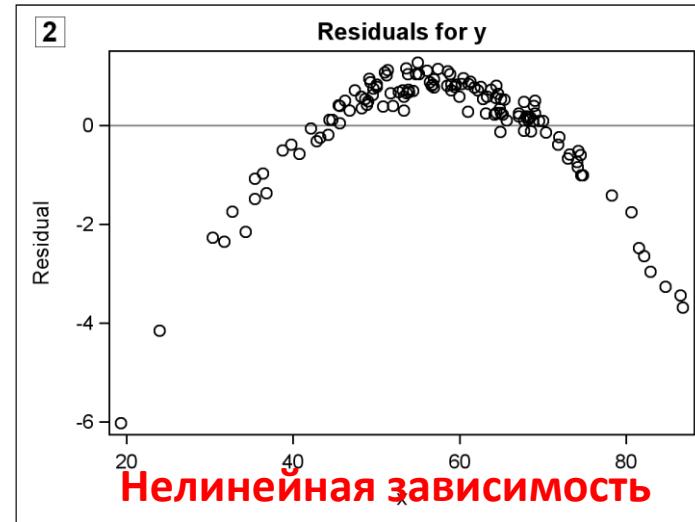
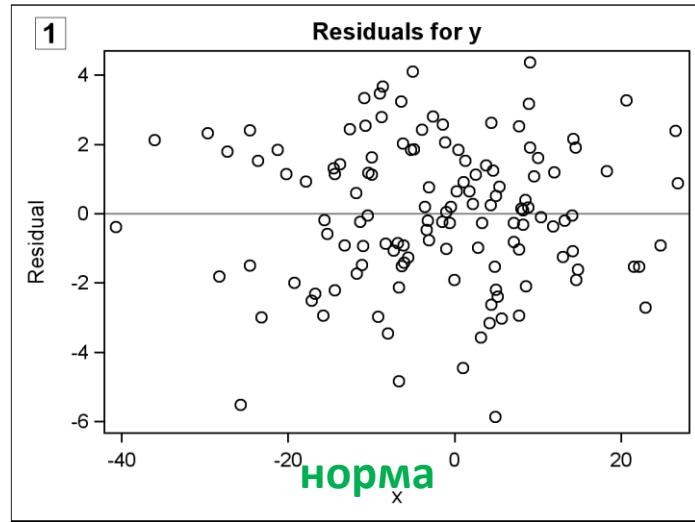
Residual standard error: 3.345 on 425 degrees of freedom
Multiple R-squared:  0.6622,    Adjusted R-squared:  0.6606 
F-statistic: 416.5 on 2 and 425 DF,  p-value: < 2.2e-16
```

Проверка предположений модели с помощью графиков остатков

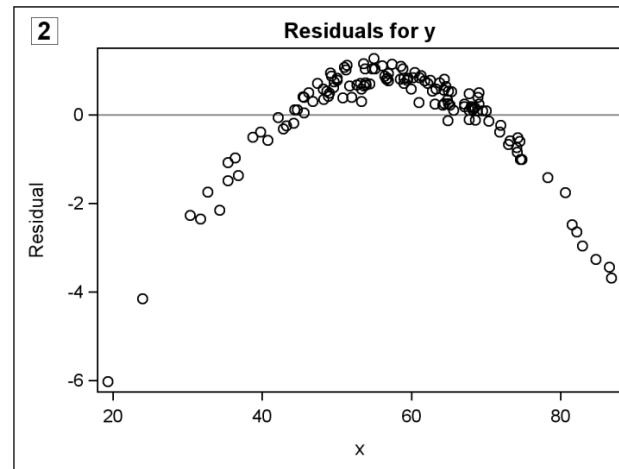
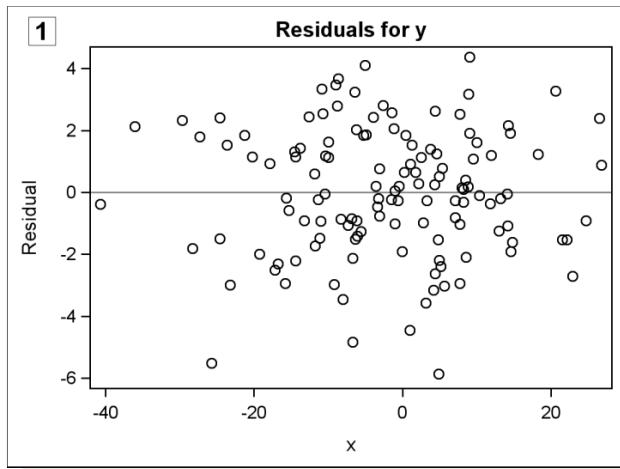


Графики: как остатки зависят от прогноза, от отклика, от предикторов

Графики остатков

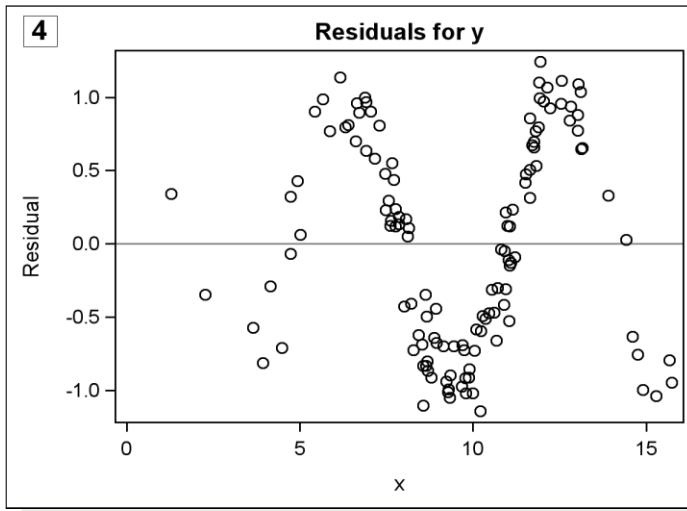


Проверка графиков остатков

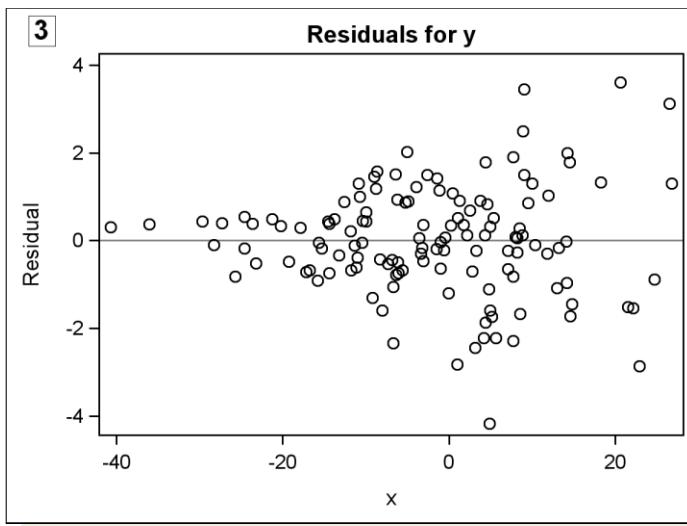


- Слева:
 - Остатки случайно расположены вокруг референсной линии = 0
 - Нет явных зависимостей и тенденций, модель адекватна
- Справа:
 - Есть явная зависимость, модель некорректна.
 - В зависимости от вида тенденции можно пробовать добавлять нелинейность в модель (полином, сплайны и т.д.)

Проверка графиков остатков

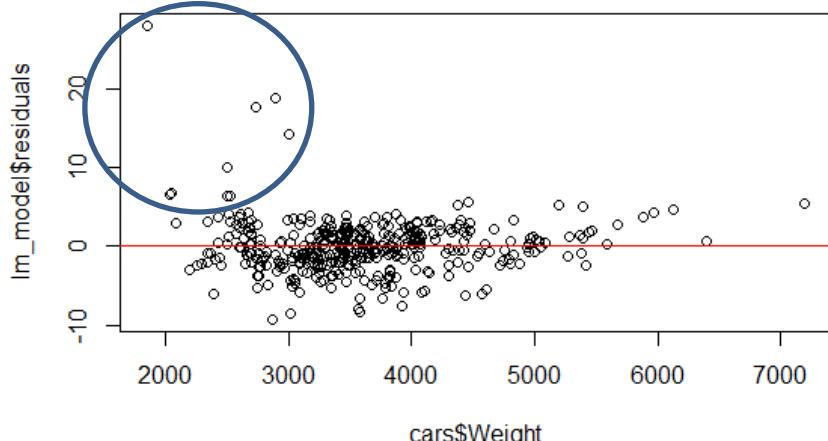
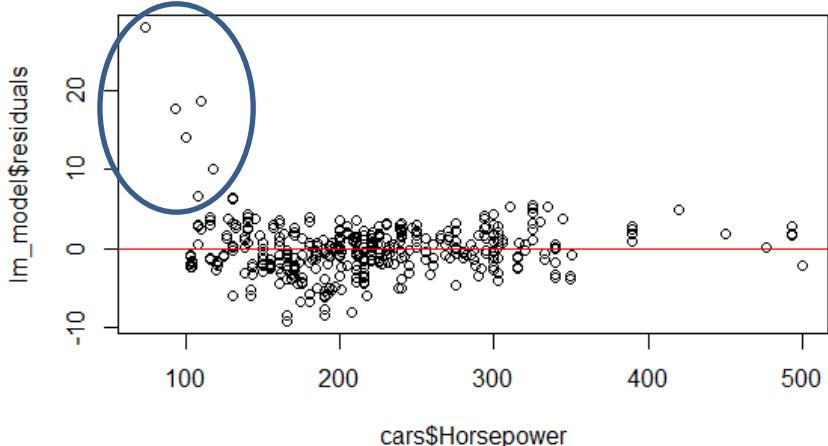


- Наблюдения не независимы, присутствует цикличность
- Попробовать анализ автокорреляций



- Гетероскедастичность
- Преобразовать переменные или использовать функцию связи в процедурах для обобщенных линейных моделей

Вывод множественной регрессии



```
> plot(cars$Horsepower, lm_model$residuals)
> abline(h = 0, col = "red")
>
> plot(cars$Weight, lm_model$residuals)
> abline(h = 0, col = "red")
```

```
> aov_model <- aov(MPG_Highway ~ Horsepower + Weight, cars)
> lm_model <- lm(MPG_Highway ~ Horsepower + Weight, cars)
>
> summary(aov_model)
   Df Sum Sq Mean Sq F value Pr(>F)
Horsepower     1    5895    5895  526.9 <2e-16 ***
Weight         1    3424    3424  306.1 <2e-16 ***
Residuals    425   4755      11
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(lm_model)

Call:
lm(formula = MPG_Highway ~ Horsepower + Weight, data = cars)

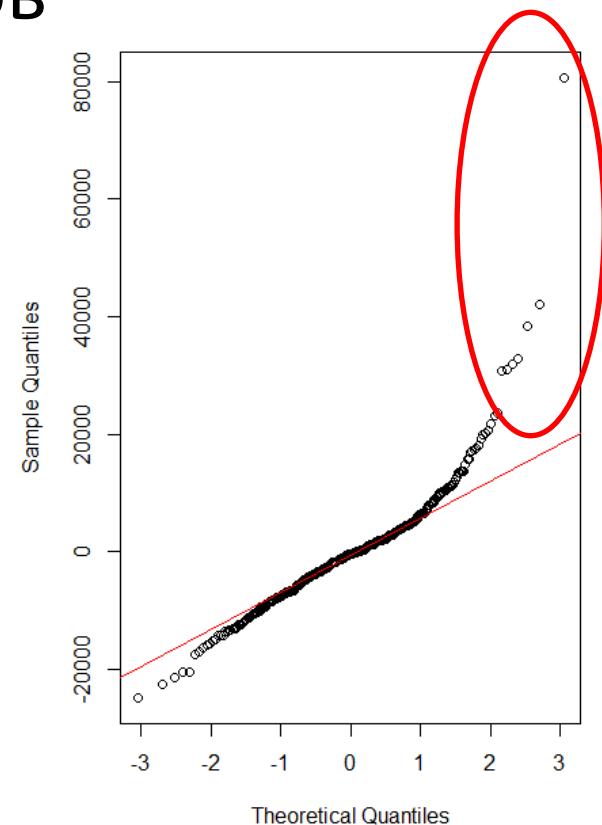
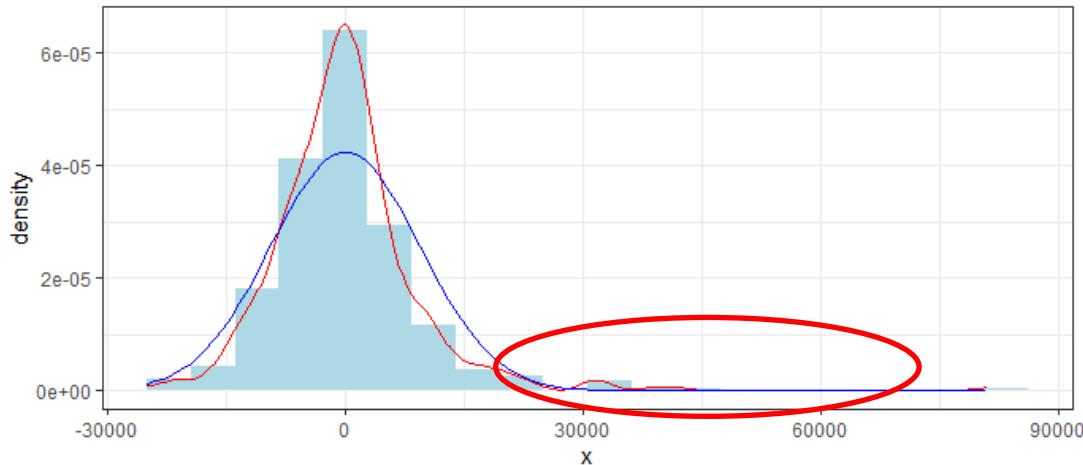
Residuals:
    Min      1Q  Median      3Q      Max 
-9.2682 -1.7478 -0.1446  1.7058 28.0361 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 48.2961097  0.7800378 61.915 < 2e-16 ***
Horsepower  -0.0196774  0.0029039 -6.776 4.13e-11 ***
Weight       -0.0048085  0.0002749 -17.495 < 2e-16 ***
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.345 on 425 degrees of freedom
Multiple R-squared:  0.6622,    Adjusted R-squared:  0.6606 
F-statistic: 416.5 on 2 and 425 DF,  p-value: < 2.2e-16
```

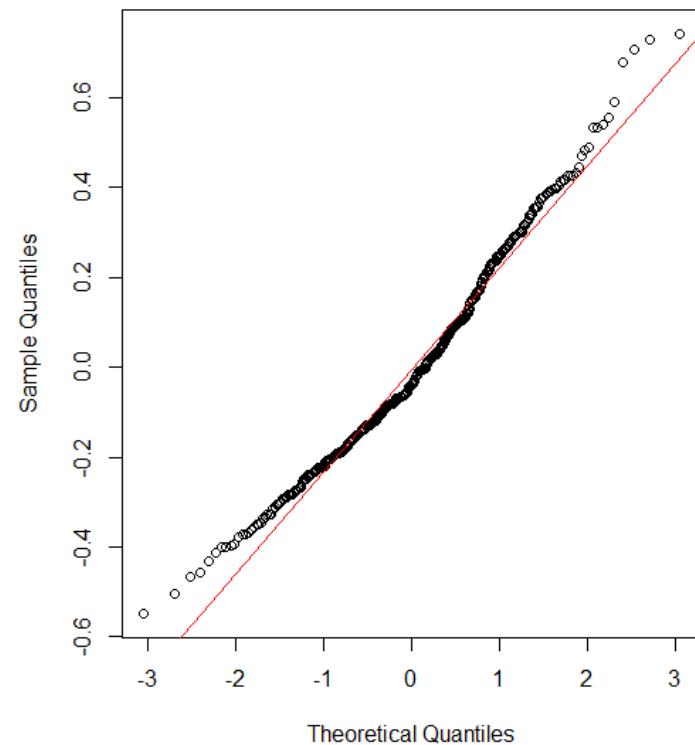
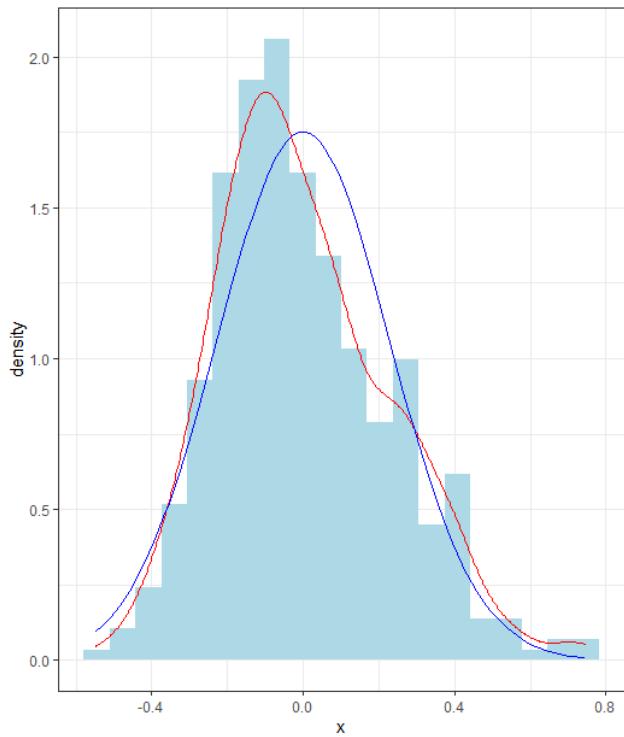
Оценка нормальности ошибки с помощью графиков остатков

```
> plot(cars$Horsepower, cars$Weight)
> contour(lm_model, Weight ~ Horsepower, add = TRUE)
>
>
> lm_model_inv <- lm(Invoice ~ Horsepower + Weight + Wheelbase, cars)
>
> print_qqplot <- function(x){
+   qqnorm(x)
+   qqline(x, col = "red")
+ }
>
> print_hist_kernel <- function(x) {
+   dnorm_pars <- c(mean = mean(x), sd = sd(x))
+   data <- data.frame(x = x)
+   ggplot(data, aes(x = x)) +
+     geom_histogram(aes(y = ..density..), fill="light blue", bins=20) +
+     geom_density(aes(y = ..density..), color = "red", alpha=.5) +
+     stat_function(fun = dnorm, color = "blue", args = dnorm_pars) +
+     theme_bw()
+ }
>
> print_qqplot(lm_model_inv$residuals)
> print_hist_kernel(lm_model_inv$residuals)|
```



Оценка нормальности ошибки с помощью графиков остатков

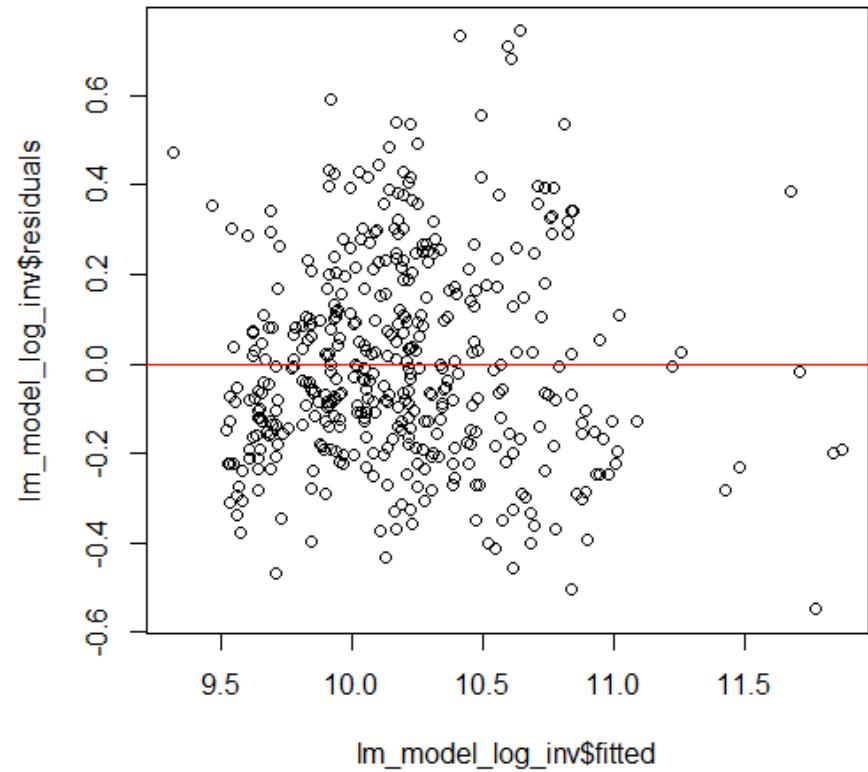
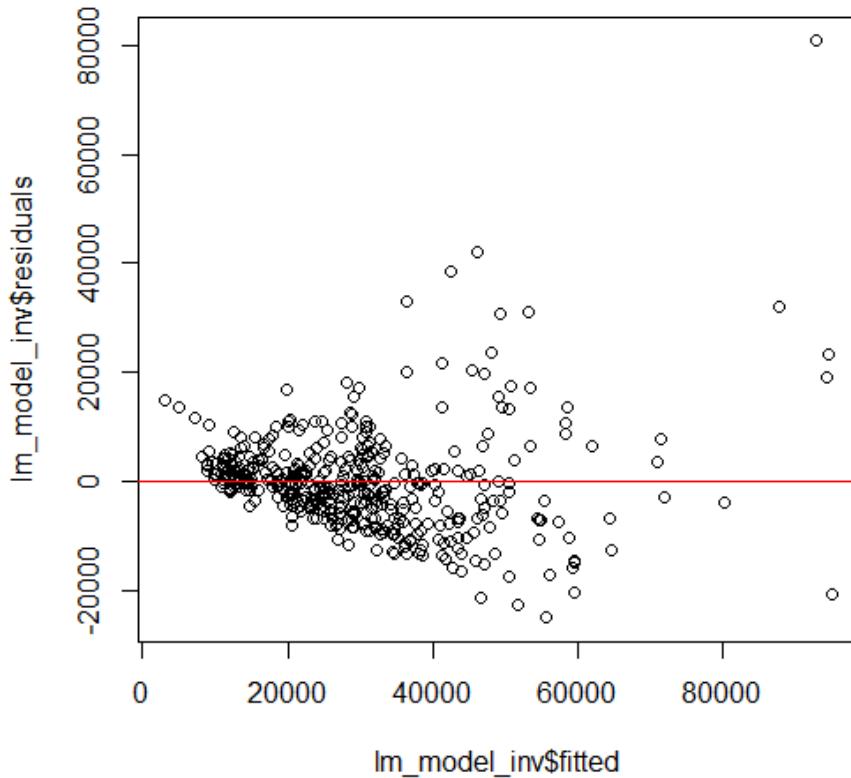
```
> cars$log_invoice <- log(cars$Invoice + 1)
>
> lm_model_log_inv <- lm(log_invoice ~ Horsepower + Weight + Wheelbase, cars)
>
> print_qqplot(lm_model_log_inv$residuals)
> print_hist_kernel(lm_model_log_inv$residuals)
```



Можно ли считать потом $\text{Invoice} = \text{Exp}(\text{Log_invoice}(X)) - 1$?
Нет! Т.к. $E(g(y|x)) \neq g(E(y|x))$

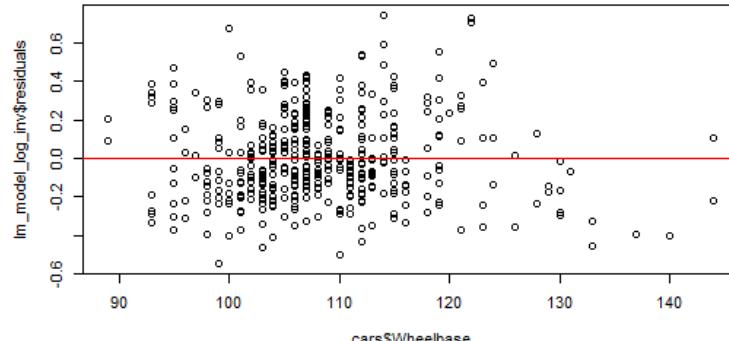
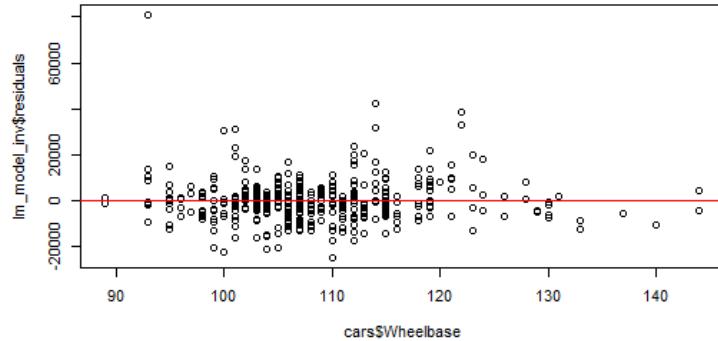
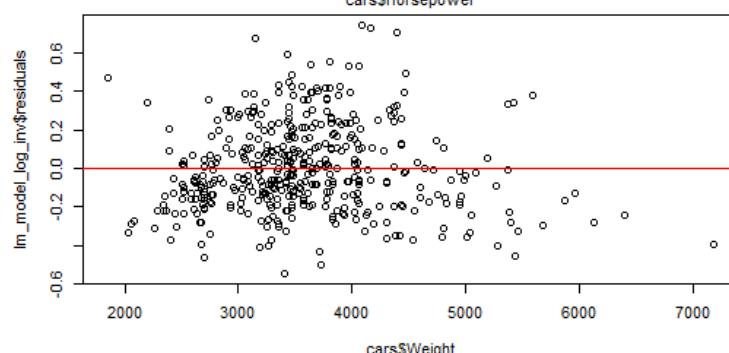
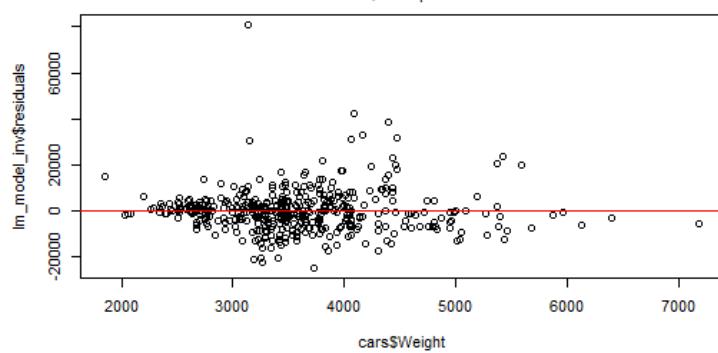
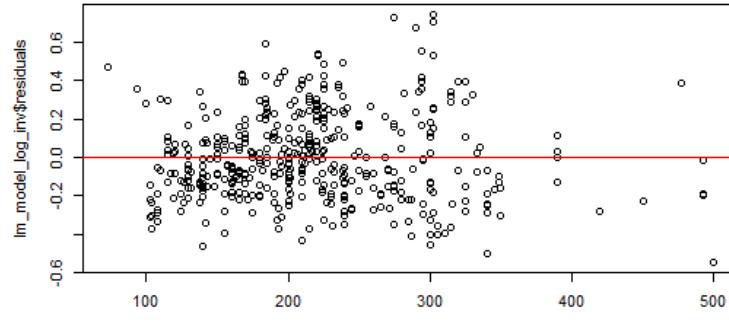
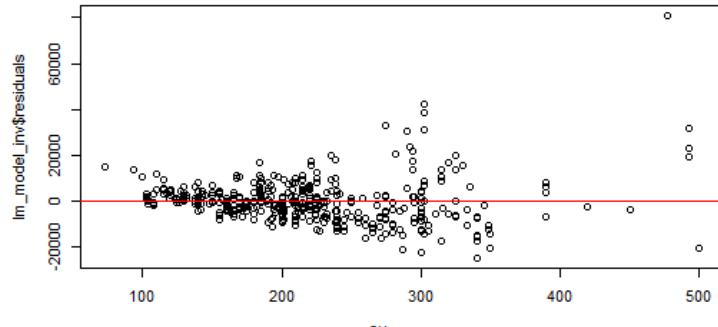
Проверка на постоянную дисперсию ошибки (неформально)

- Графики зависимости остатков от прогноза



Проверка на постоянную дисперсию ошибки (неформально)

- Графики зависимости остатков от предикторов



Оценки коэф в случае гетероскедастичности

- Точечные оценки те же, стандартные ошибки и доверительные интервалы нет
- «Консистентные» приближения ковариационной матрицы (можно задать в параметрах модели какую именно использовать), где e – остаток. а

$$h_{ii} = \mathbf{x}_i' (\mathbf{X}' \mathbf{X})^{-1} \mathbf{x}_i'$$

$$\text{HC}_0 = (\mathbf{X}' \mathbf{X})^{-1} (\mathbf{X}' \text{diag}(e_i^2) \mathbf{X}) (\mathbf{X}' \mathbf{X})^{-1}$$

$$\text{HC}_1 = \frac{n}{n-p} \text{HC}_0$$

$$\text{HC}_2 = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \text{diag}\left(\frac{e_i^2}{1 - h_{ii}}\right) \mathbf{X} (\mathbf{X}' \mathbf{X})^{-1}$$

$$\text{HC}_3 = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \text{diag}\left(\frac{e_i^2}{(1 - h_{ii})^2}\right) \mathbf{X} (\mathbf{X}' \mathbf{X})^{-1}$$

```
vcovHC(x, type = c("HC3",
  "const", "HC", "HC0",
  "HC1", "HC2", "HC4",
  "HC4m", "HC5"), ...)
```

Проверка на постоянную дисперсию ошибки (формально)

LOG_Invoice

```
> bptest(lm_model_inv)

studentized Breusch-Pagan test

data: lm_model_inv
BP = 69.405, df = 3, p-value = 5.724e-15

> summary(lm_model_inv)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 35434.018   7106.827   4.986 9.01e-07 *** 
Horsepower   212.507    8.343   25.471 < 2e-16 ***  
Weight        2.133     1.122   1.902  0.0578 .    
Wheelbase    -544.870   86.197  -6.321 6.58e-10 *** 
Residual standard error: 5446 on 424 degrees of freedom
Multiple R-squared:  0.7153, Adjusted R-squared:  0.7133 
F-statistic: 355.1 on 3 and 424 DF,  p-value: < 2.2e-16

> coeftest(lm_model_inv, vcov = vcovHC(lm_model_inv, "HC0"))

t test of coefficients:

            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 35434.0181  8328.6347  4.2545 2.581e-05 *** 
Horsepower   212.5072  20.7084 10.2619 < 2.2e-16 ***  
Weight        2.1335   1.4467  1.4747   0.141    
Wheelbase    -544.8702 101.0338 -5.3929 1.154e-07 *** 
```

Invoice

```
> bptest(lm_model_log_inv)

studentized Breusch-Pagan test

data: lm_model_log_inv
BP = 17.844, df = 3, p-value = 0.0004736

> summary(lm_model_log_inv)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.853e+00  1.719e-01  57.323 < 2e-16 *** 
Horsepower   5.251e-03  2.018e-04  26.023 < 2e-16 ***  
Weight       1.626e-04  2.712e-05  5.995 4.36e-09 *** 
Wheelbase    -1.279e-02  2.085e-03  -6.134 1.97e-09 *** 
Residual standard error: 0.2285 on 424 degrees of freedom
Multiple R-squared:  0.773, Adjusted R-squared:  0.7714 
F-statistic: 481.4 on 3 and 424 DF,  p-value: < 2.2e-16

> coeftest(lm_model_log_inv, vcov = vcovHC(lm_model_log_inv, "HC0"))

t test of coefficients:

            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.8528e+00  1.9238e-01  51.2143 < 2.2e-16 *** 
Horsepower   5.2510e-03  2.4234e-04  21.6681 < 2.2e-16 ***  
Weight       1.6261e-04  2.8796e-05  5.6472 2.993e-08 *** 
Wheelbase    -1.2787e-02  2.2499e-03 -5.6835 2.458e-08 *** 
```

Тест Бройша — Пагана: вспомогательная модель – зависимость стандартизованных остатков от исходных предикторов, статистика – сумма кв. остатков вспомогательной модели распределена по Хи²

$$y_t = x_t^T b + \varepsilon_t \quad \hat{\sigma}^2 = RSS/n \quad e_t^2 / \hat{\sigma}^2 = \gamma_0 + x_t^T \gamma + \nu_t$$

Проверка на постоянную дисперсию ошибки (формально)

- Коэф. ранговой корреляции Спирмана между модулем остатков и прогнозом:
 - **Близко к нулю** – дисперсия постоянная
 - **Больше/меньше нуля** – дисперсия растет/уменьшается вместе с прогнозом

```
> cor.test(abs(lm_model_inv$residuals), lm_model_inv$fitted, method = "spearman")
> cor.test(abs(lm_model_log_inv$residuals), lm_model_log_inv$fitted, method = "spearman")

          Spearman's rank correlation rho

data: abs(lm_model_inv$residuals) and lm_model_inv$fitted
S = 6855128, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.4753884

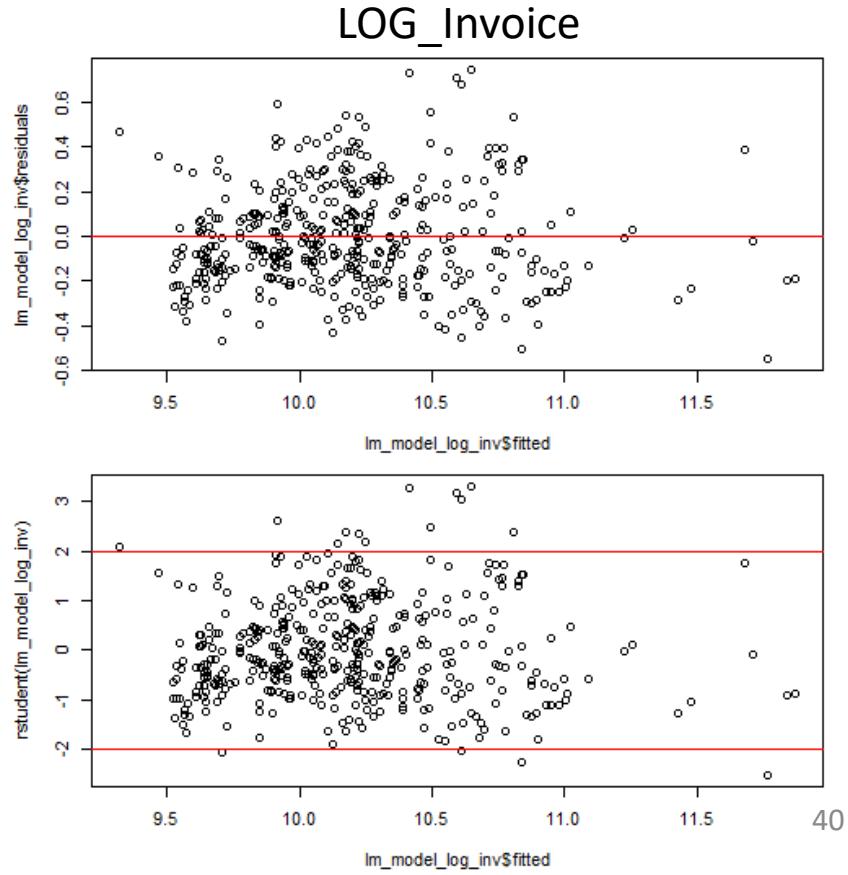
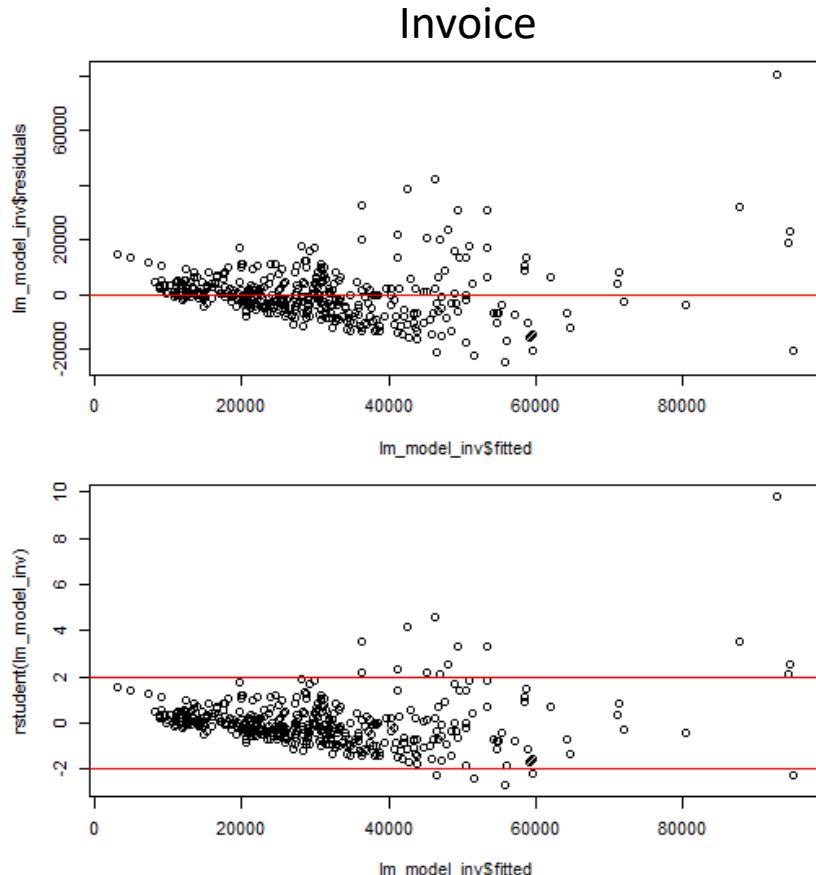
          Spearman's rank correlation rho

data: abs(lm_model_log_inv$residuals) and lm_model_log_inv$fitted
S = 10550075, p-value = 6.048e-05
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.1926202
```

Проверка на корректность уравнения регрессии (линейность)

— Графики:

- Зависимость остатков и «стьюдентизованных» остатков от прогноза
- другие ...

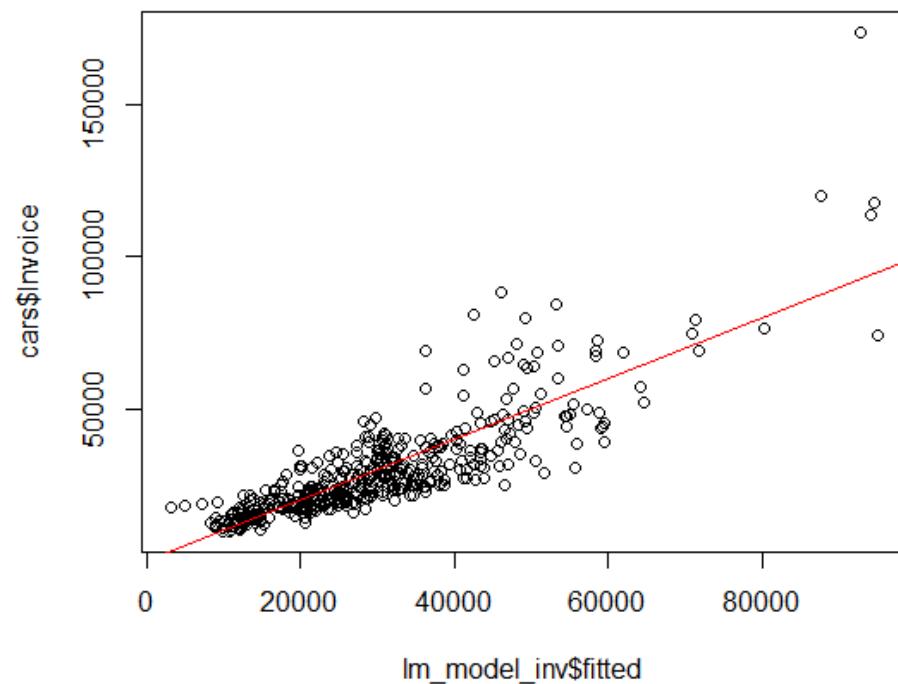


Проверка на корректность уравнения регрессии (линейность)

— Графики:

- Зависимость реального отклика от прогноза
- другие ...

```
> plot(lm_model_inv$fitted, cars$Invoice)
> abline(a = 0, b = 1, col = "red")
```



```
> plot(lm_model_log_inv$fitted, cars$log_invoice)
> abline(a = 0, b = 1, col = "red")
```

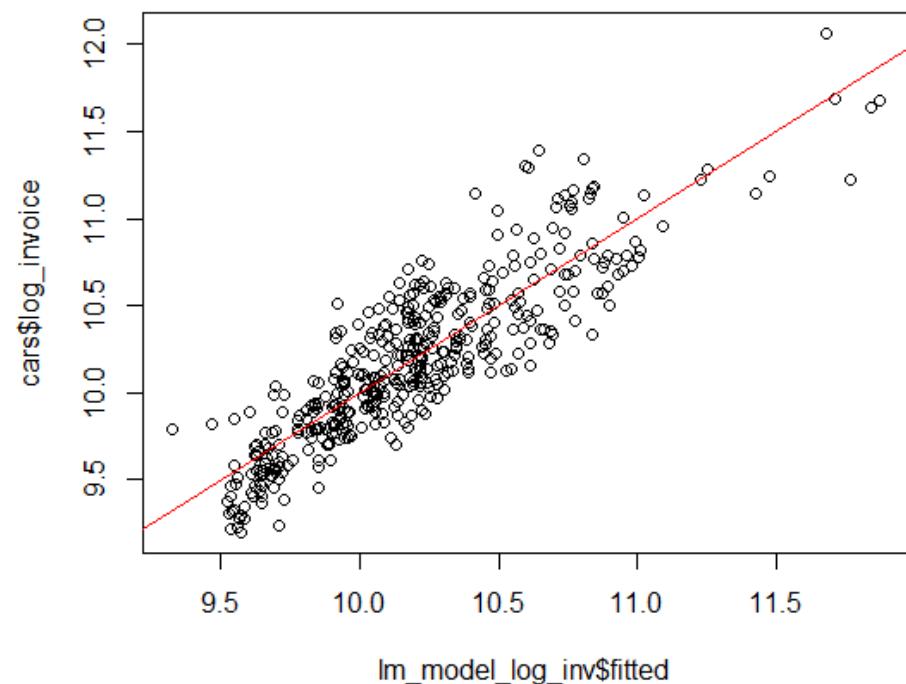
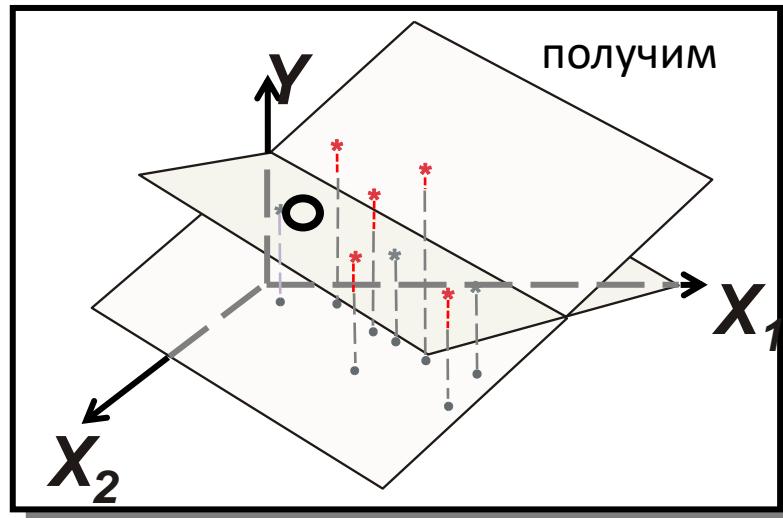
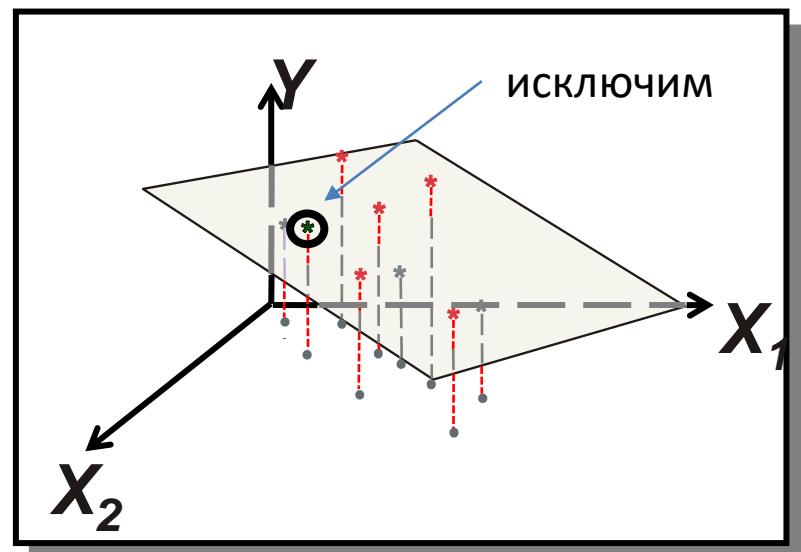
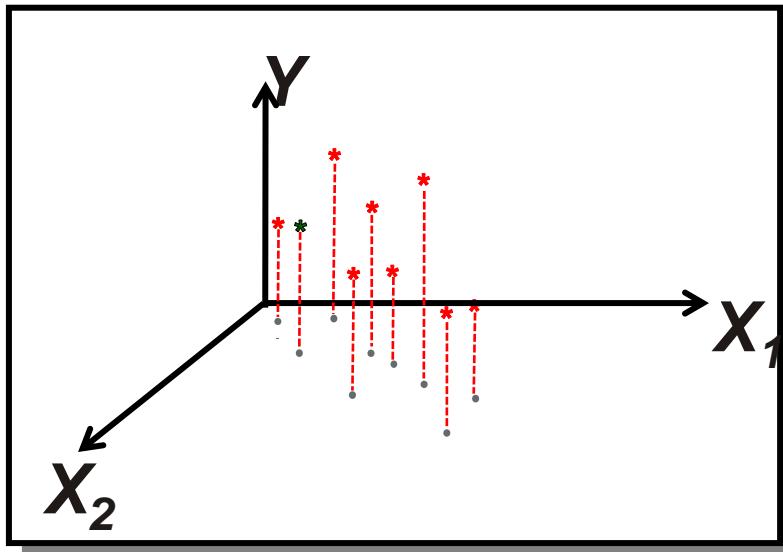


Иллюстрация мультиколлинеарности



- Портятся статистики с оценкой значимости переменных
- Увеличивается вариативность оценки параметров и как следствие ошибка
- Есть тенденция к неограниченному росту коэф.

«Ручная» проверка на мультиколлинеарность

- С помощью Variance inflation factors: $VIF_i = \frac{1}{1 - R_i^2}$
- R_i – коэф. детерминации i -го предиктора на остальные, например
- Model $Y=X_1 X_2 X_3 \Rightarrow$ Model $X_2 = X_1 X_3$
- Больше 10 – плохо

```
> full_lm <- lm(Invoice ~ Horsepower + Weight + Wheelbase +
+                  Length + EngineSize + MPG_City + MPG_Highway, cars)
> summary(full_lm)

Call:
lm(formula = Invoice ~ Horsepower + Weight + Wheelbase + Length +
    EngineSize + MPG_City + MPG_Highway, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-22833  -4912   -726   3641  78723 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7175.373   9299.445   0.772  0.44079    
Horsepower   234.994    11.109  21.153 < 2e-16 ***  
Weight        6.544     1.428   4.584 6.02e-06 ***  
Wheelbase    -574.582   135.424  -4.243 2.72e-05 ***  
Length        6.288     74.689   0.084  0.93294    
EngineSize   -1871.859   912.341  -2.052  0.04082 *   
MPG_City     -174.530    280.910  -0.621  0.53474    
MPG_Highway   713.343    274.821   2.596  0.00977 **  

```

```
ols_coll_diag(model)
ols_vif_tol(model)
ols_eigen_cindex(model)
```

	> ols_vif_tol(full_lm)		
	Variables	Tolerance	VIF
1	Horsepower	0.31352833	3.189504
2	Weight	0.17008308	5.879480
3	Wheelbase	0.15760185	6.345103
4	Length	0.17363812	5.759104
5	EngineSize	0.19520129	5.122917
6	MPG_City	0.09222327	10.843250
7	MPG_Highway	0.08021158	12.467027

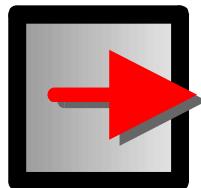
«Ручная» проверка на мультиколлинеарность

- Condition index values:
 - разложение на с.в. нормализованной $X^T X$, CI – $\sqrt{\text{макс с.зн./с.зн.}}$
 - для каждой переменной оценка описываемой пропорции вариации по каждой из компонент, если 2 и больше переменных имеют вес больше 0.5 и сi больше 30 – плохо!

```
> full_lm <- lm(Invoice ~ Horsepower + Weight + Wheelbase +
+ Length + EngineSize + MPG_City + MPG_Highway, cars)
> ols_eigen_cindex(full_lm)
   Eigenvalue Condition Index    intercept      Horsepower       Weight      Wheelbase
1  7.6746631486        1.000000 3.888577e-05 0.0004926381 0.0001208389 1.561406e-05
2  0.2659552226        5.371869 9.466457e-05 0.0216393533 0.0015814576 8.160238e-07
3  0.0325177900       15.362757 3.643828e-04 0.5615447901 0.0465824073 1.851947e-03
4  0.0162045748       21.762603 8.474146e-03 0.1723068820 0.0288778815 2.395106e-03
5  0.0062490930       35.044605 2.556455e-02 0.0429480466 0.4997653288 8.149857e-03
6  0.0022093923       58.937708 1.336544e-01 0.0021235389 0.3369727335 1.534491e-02
7  0.0016624732       67.944192 8.255279e-01 0.1957060516 0.0559115483 8.533264e-02
8  0.0005383055      119.403002 6.281105e-03 0.0032386995 0.0301878041 8.869091e-01
   Length     EngineSize      MPG_City      MPG_Highway
1 1.728727e-05 0.0003296357 9.141311e-05 5.578906e-05
2 1.272837e-06 0.0180856411 6.567945e-03 2.883016e-03
3 1.759678e-03 0.0056648073 5.305543e-03 4.596978e-03
4 1.882002e-03 0.7934265948 2.996303e-02 3.772070e-03
5 2.508165e-02 0.0548283581 2.190477e-01 2.058049e-02
6 6.774152e-05 0.0063787864 5.262876e-01 8.465682e-01
7 1.083358e-01 0.0965487193 1.078299e-01 2.744389e-02
8 8.628546e-01 0.0247374573 1.049068e-01 9.409957e-02
```

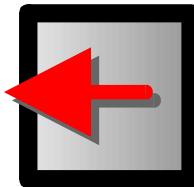
Пошаговые методы выбора значимых переменных

FORWARD
SELECTION



```
ols_step_forward_p(model, penter = 0.1,...)  
ols_step_forward_aic(model, ...)
```

BACKWARD
ELIMINATION



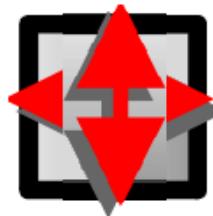
```
ols_step_backward_p(model, prem = 0.1,...)  
ols_step_backward_aic(model,...)
```

STEPWISE
SELECTION



```
ols_step_both_p(model,  
    pent = 0.1, prem = 0.3,...)  
ols_step_both_aic(model, ...)
```

BEST SUBSET
SELECTION



```
ols_step_best_subset(model, ...)  
ols_step_all_possible(model, ...)
```

Информационные критерии

- Информационный критерий равен
 $n \log(SSE/n) + \text{Штраф}$ (у каждого свой, см. таблицу):

Information Criteria	Penalty Component
AIC	$2p + n + 2$
AICC	$\frac{n(n + p)}{n - p - 2}$
BIC	$2(p + 2)q - 2q^2$
SBC	$p \log(n)$

- p – число параметров
- n – число наблюдений
- SSE – сумма квадратов ошибок
- $q = \frac{n\hat{\sigma}^2}{SSE}$. $\hat{\sigma}^2$ – оценка дисперсии для полной модели

Forward Selection

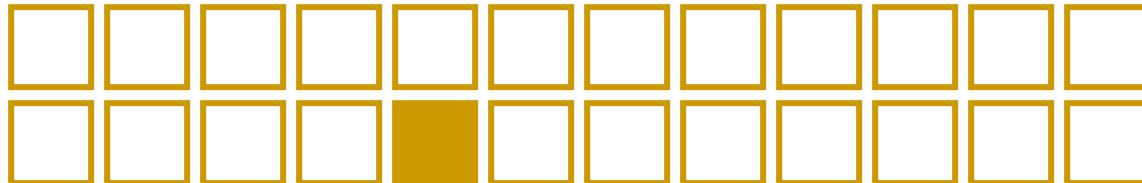
0



...

Forward Selection

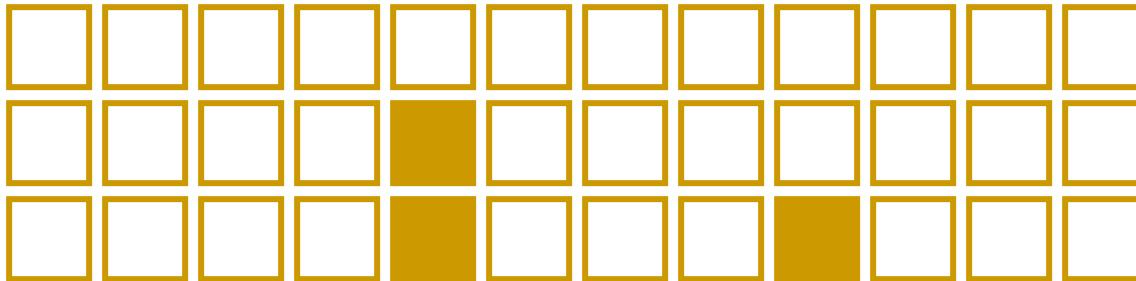
0
1



...

Forward Selection

0

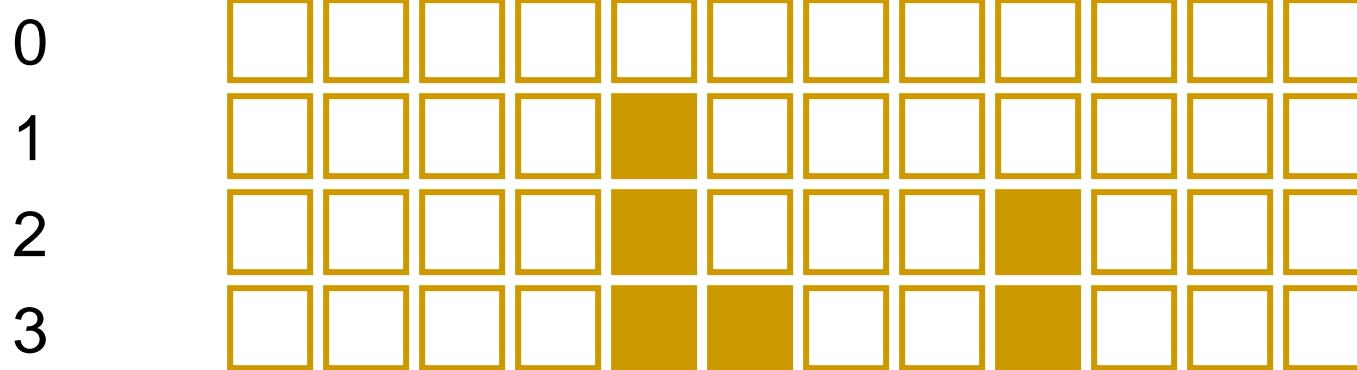


1

2

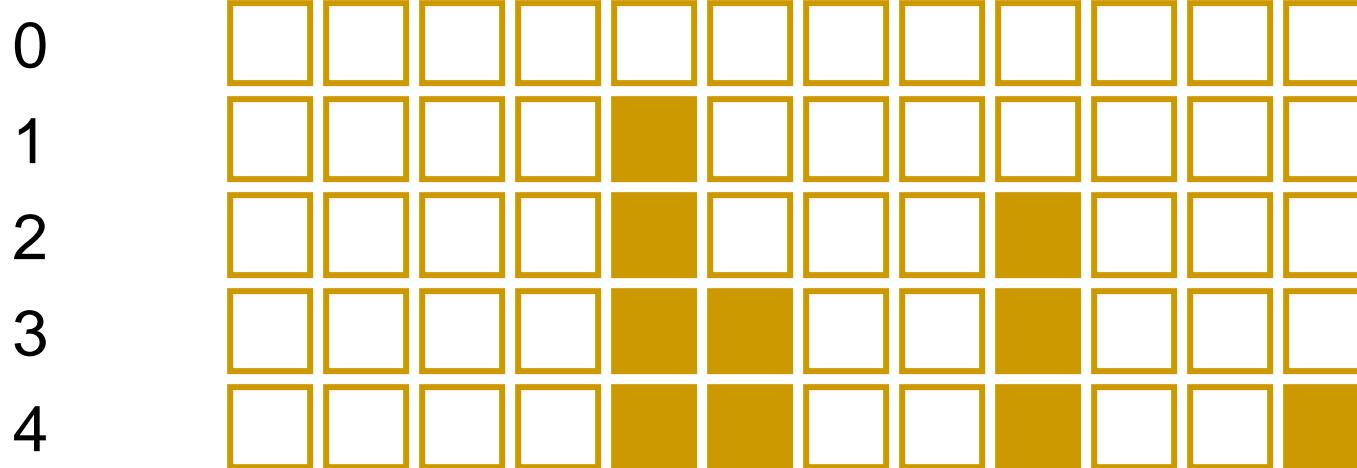
...

Forward Selection

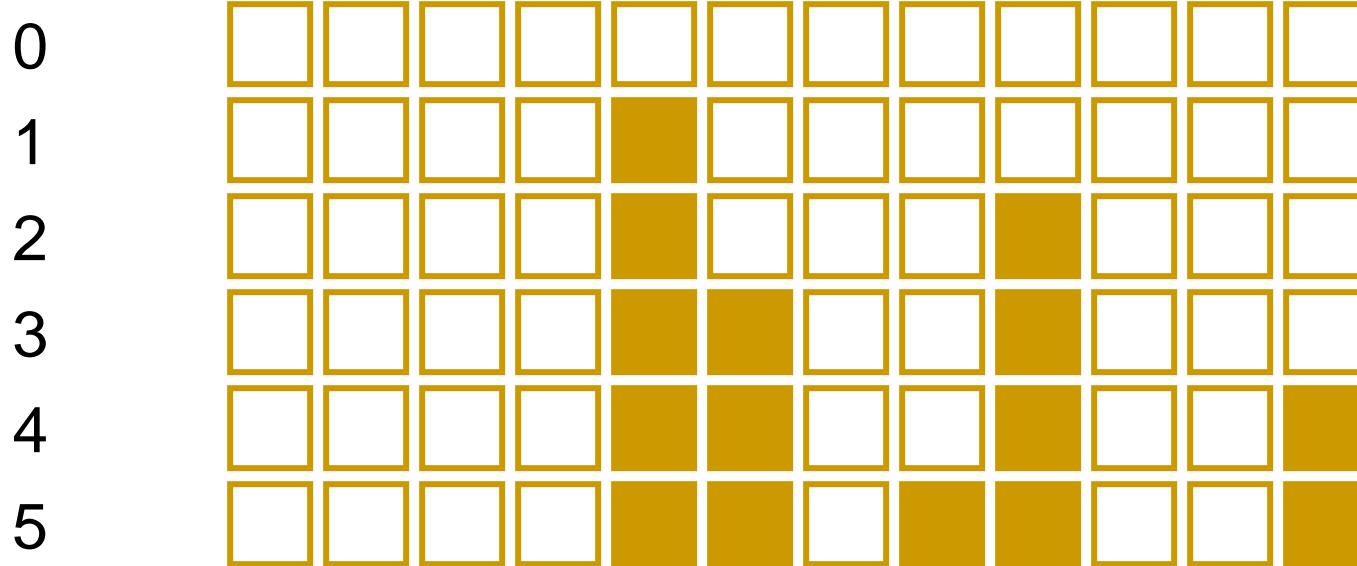


...

Forward Selection



Forward Selection



...

Forward Selection

0
1
2
3
4
5
Stop

Backward Elimination

0



...

Backward Elimination

0

1



...

Backward Elimination

0	
1	
2	

...

Backward Elimination

0										
1										
2										
3										
...										

...

Backward Elimination

0										
1										
2										
3										
4										

...

Backward Elimination

0											
1											
2											
3											
4											
5											

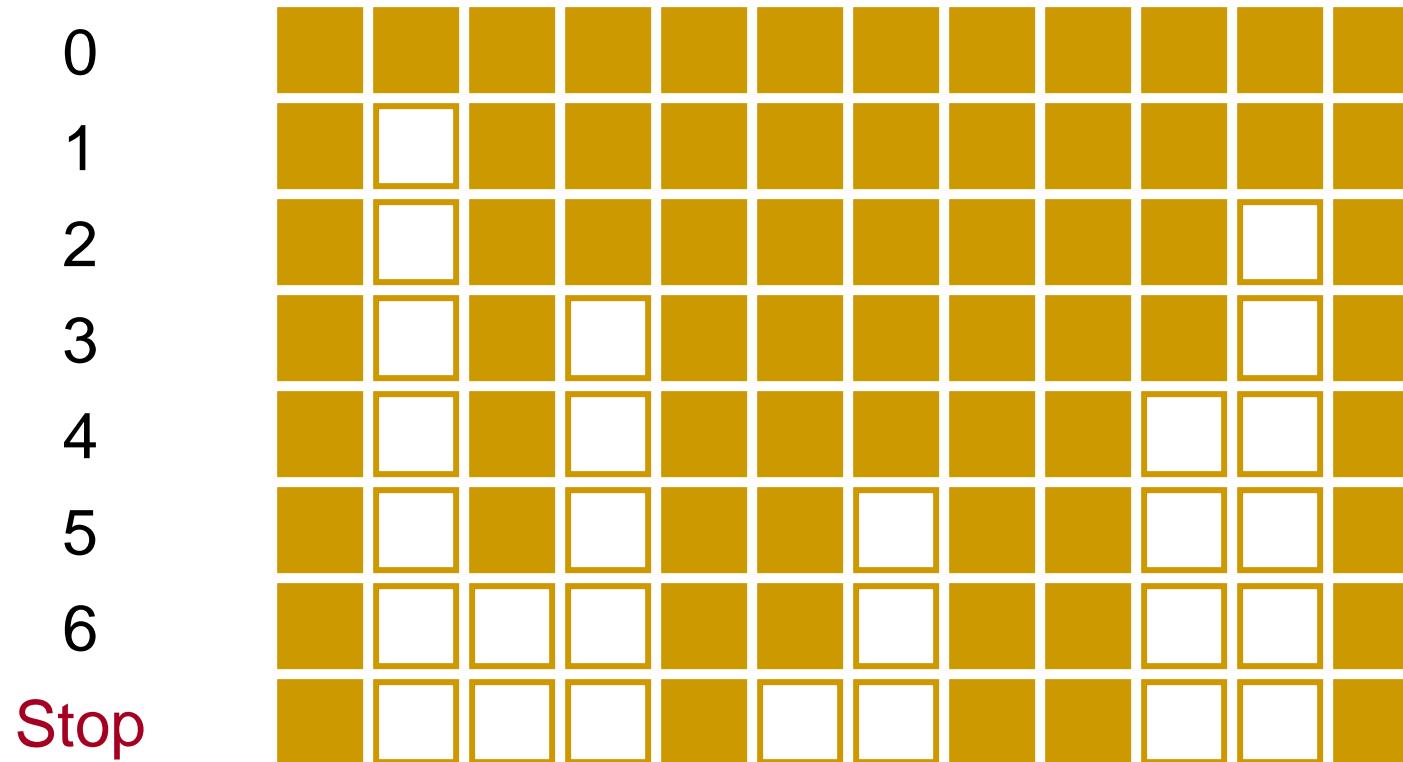
...

Backward Elimination

0											
1											
2											
3											
4											
5											
6											

...

Backward Elimination



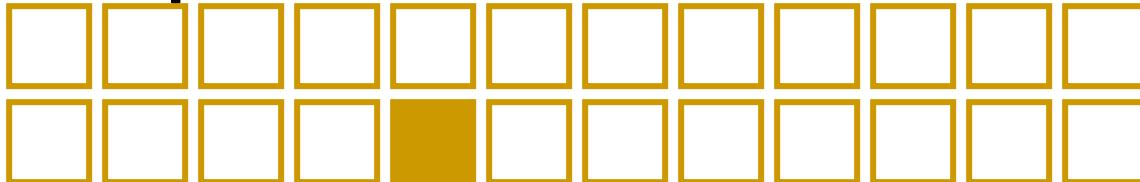
Stepwise Selection

0



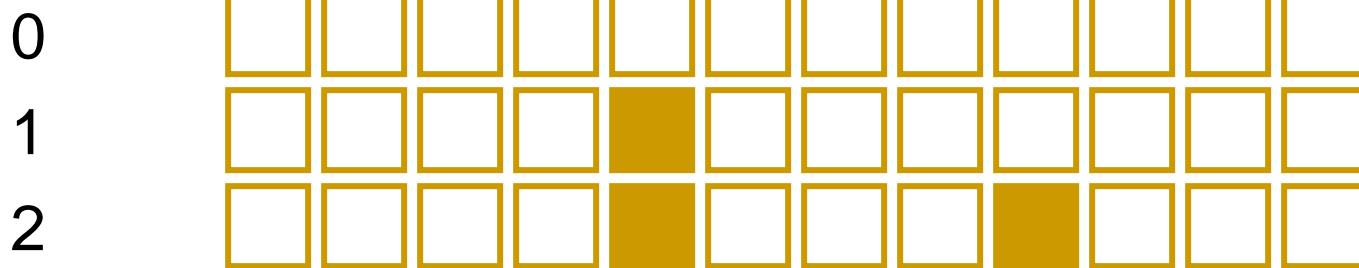
Stepwise Selection

0
1



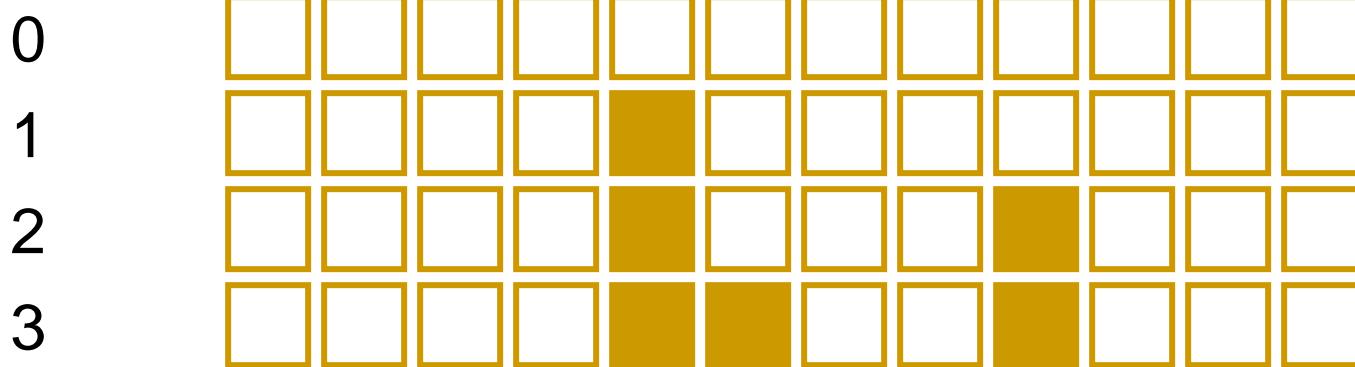
...

Stepwise Selection



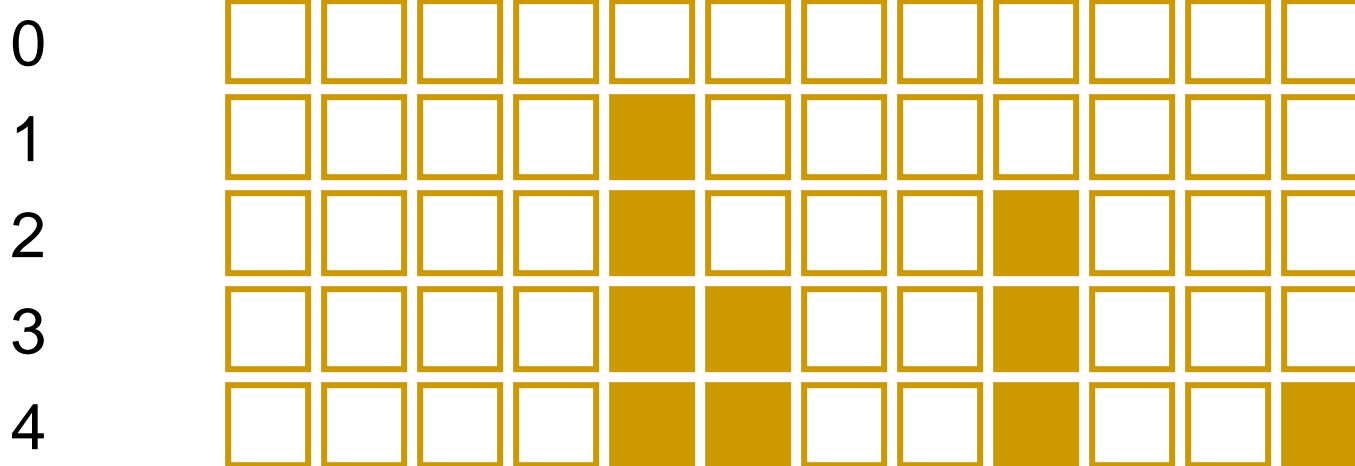
...

Stepwise Selection



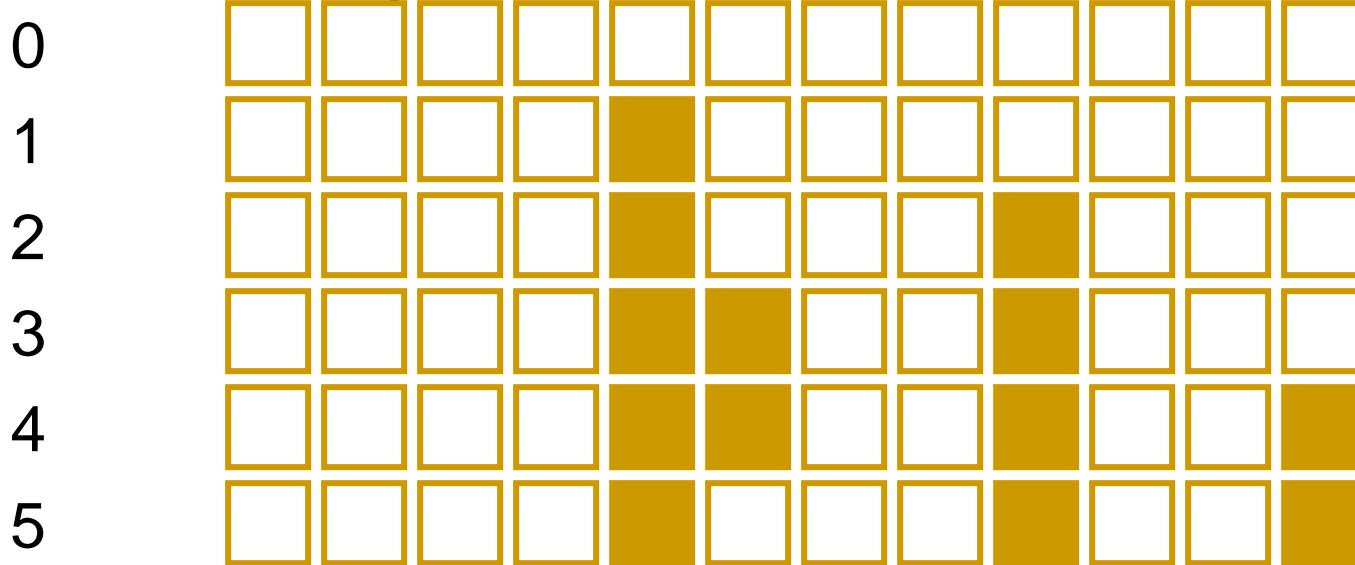
...

Stepwise Selection

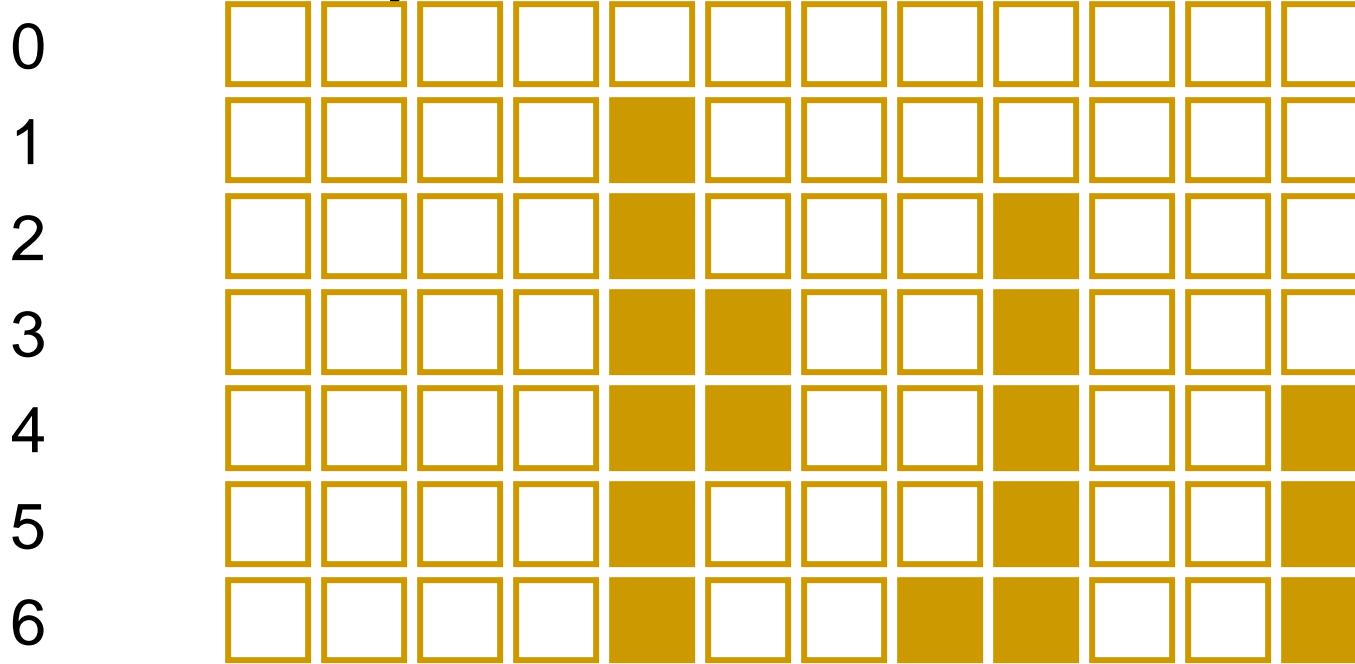


...

Stepwise Selection

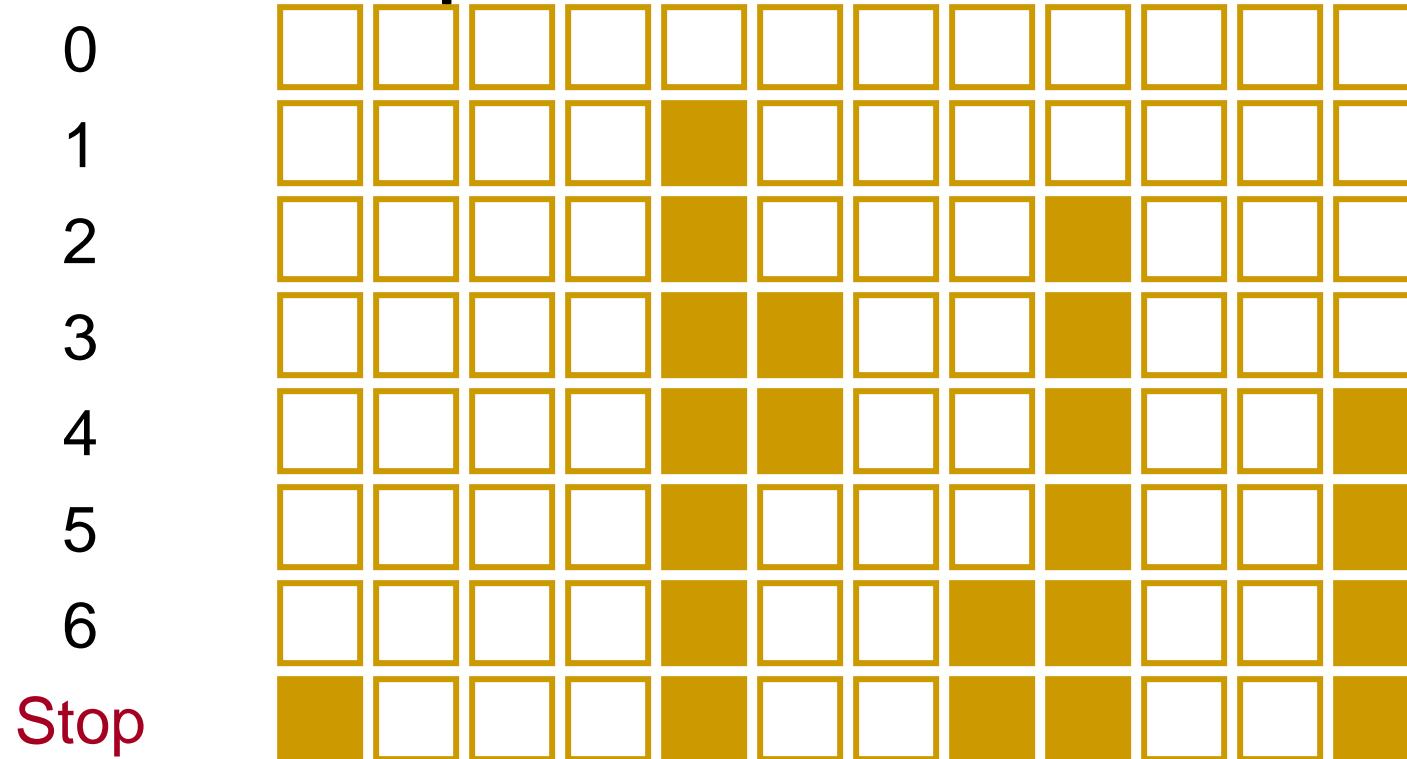


Stepwise Selection

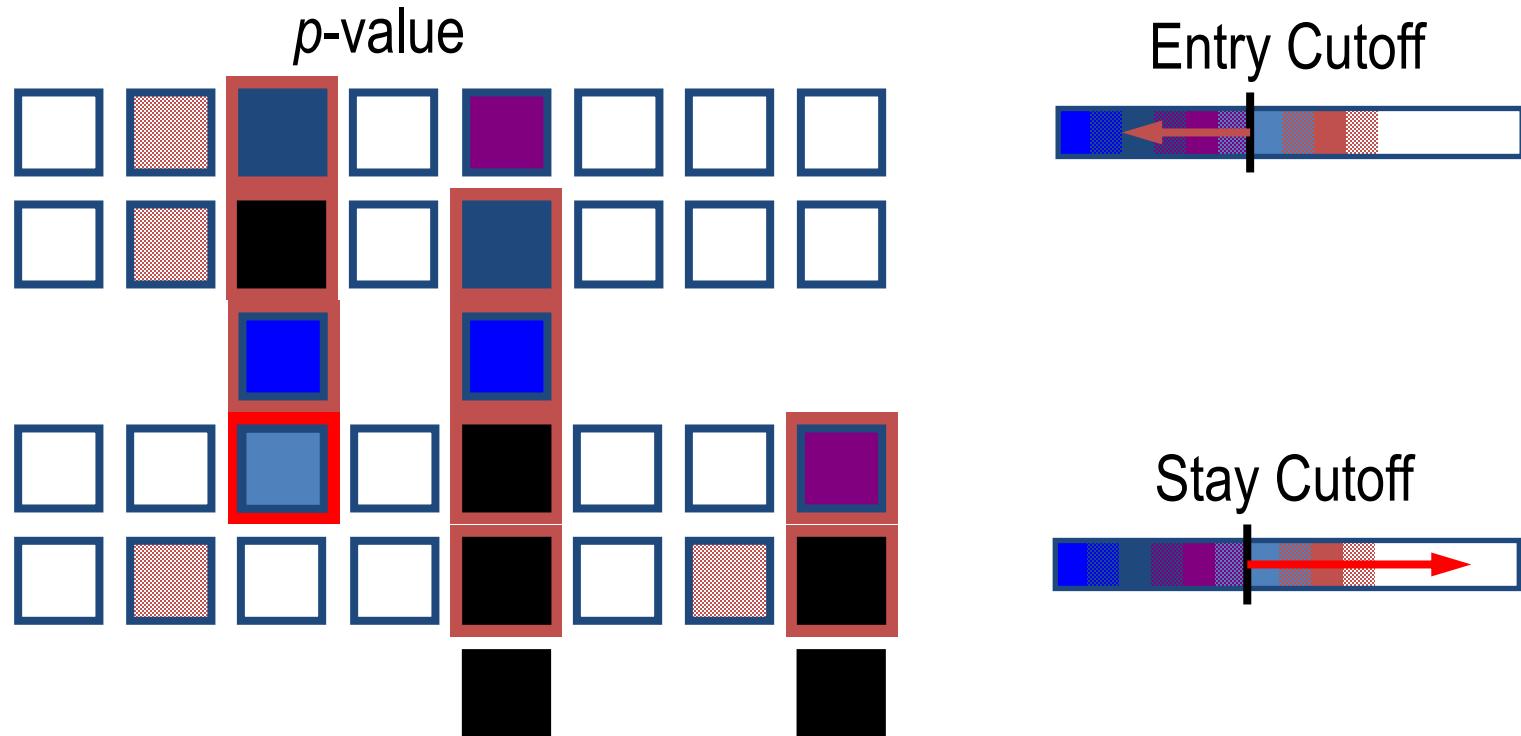


...

Stepwise Selection



Пошаговый отбор на основе p-value критерия Фишера



Полный перебор

Число предикторов (k)

0
1
2
3
4
5

Intercept
Only

1 Variable

2 Variables

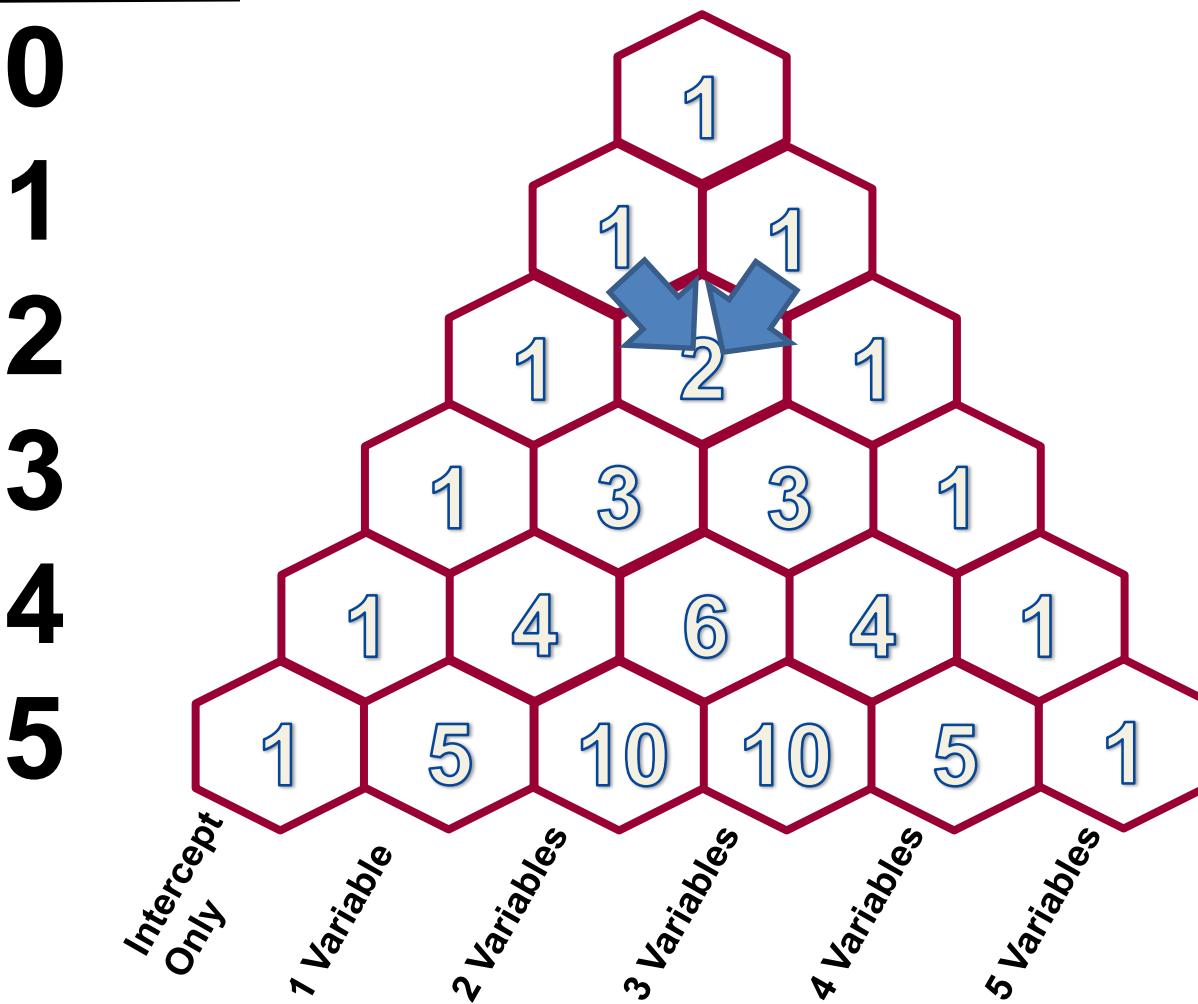
3 Variables

4 Variables

5 Variables

Число моделей (2^k)

1
2
4
8
16
32



Пример пошаговой регрессии

```
> model <- lm(MPG_Highway ~ Horsepower + Weight + Cylinders +
+                 Length + Wheelbase + EngineSize + Invoice, data = cars)
> k <- ols_step_both_p(model, progress = TRUE, details = TRUE)
```

Stepwise Selection: Step 1

+ Weight

Model Summary

R	0.791	RMSE	3.517
R-Squared	0.626	Coef. Var	13.101
Adj. R-Squared	0.625	MSE	12.368
Pred R-Squared	0.620	MAE	2.381

RMSE: Root Mean Square Error

MSE: Mean Square Error

MAE: Mean Absolute Error

ANOVA

	Sum of Squares	DF	Mean Square	F	Sig.
Regression	8805.921	1	8805.921	712.016	0.0000
Residual	5268.591	426	12.368		
Total	14074.512	427			

Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig
(Intercept)	48.251	0.820		58.836	0.000
Weight	-0.006	0.000	-0.791	-26.684	0.000

Stepwise Selection: Step 2

+ Horsepower

Model Summary

R	0.814	RMSE	3.345
R-Squared	0.662	Coef. Var	12.461
Adj. R-Squared	0.661	MSE	11.188
Pred R-Squared	0.656	MAE	2.272

RMSE: Root Mean Square Error

MSE: Mean Square Error

MAE: Mean Absolute Error

ANOVA

	Sum of Squares	DF	Mean Square	F	Sig.
Regression	9319.624	2	4659.812	416.502	0.0000
Residual	4754.887	425	11.188		
Total	14074.512	427			

Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig
(Intercept)	48.296	0.780		61.915	0.000
Weight	-0.005	0.000	-0.636	-17.495	0.000
Horsepower	-0.020	0.003	-0.246	-6.776	0.000

Пример пошаговой регрессии

Final Model Output

Model Summary			
R	0.827	RMSE	3.247
R-Squared	0.683	Coef. Var	12.094
Adj. R-Squared	0.680	MSE	10.540
Pred R-Squared	0.672	MAE	2.159

RMSE: Root Mean Square Error

MSE: Mean Square Error

MAE: Mean Absolute Error

ANOVA

	Sum of Squares	DF	Mean Square	F	Sig.
Regression	9616.138	4	2404.035	228.089	0.0000
Residual	4458.374	423	10.540		
Total	14074.512	427			

Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig
(Intercept)	39.634	2.283		17.363	0.000
Weight	-0.006	0.000	-0.731	-16.167	0.000
Horsepower	-0.033	0.005	-0.418	-7.324	0.000
Length	0.065	0.016	0.164	4.203	0.000
Invoice	0.000	0.000	0.206	4.094	0.000

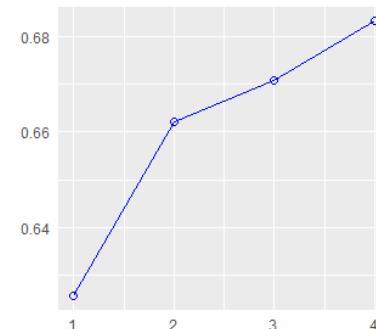
> k

Stepwise Selection Summary

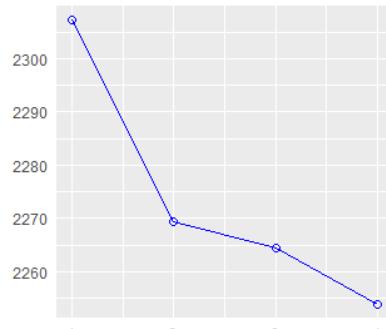
Step	Variable	Added/ Removed	R-Square	Adj. R-Square	C(p)	AIC	RMSE
1	Weight	addition	0.626	0.625	81.5110	2295.0605	3.5168
2	Horsepower	addition	0.662	0.661	34.2220	2253.1520	3.3448
3	Length	addition	0.671	0.668	24.7230	2244.2258	3.3063
4	Invoice	addition	0.683	0.680	9.7720	2229.5935	3.2465

> plot(k)

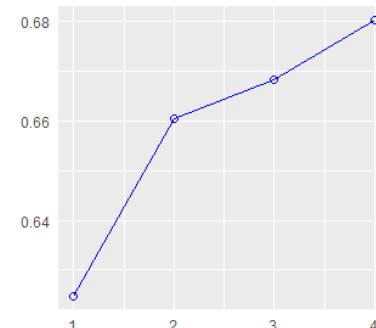
R-Square



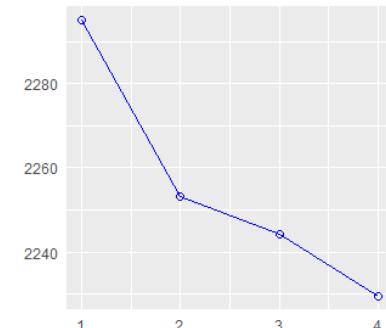
SBC



Adj. R-Square



AIC



Пример пошаговой регрессии

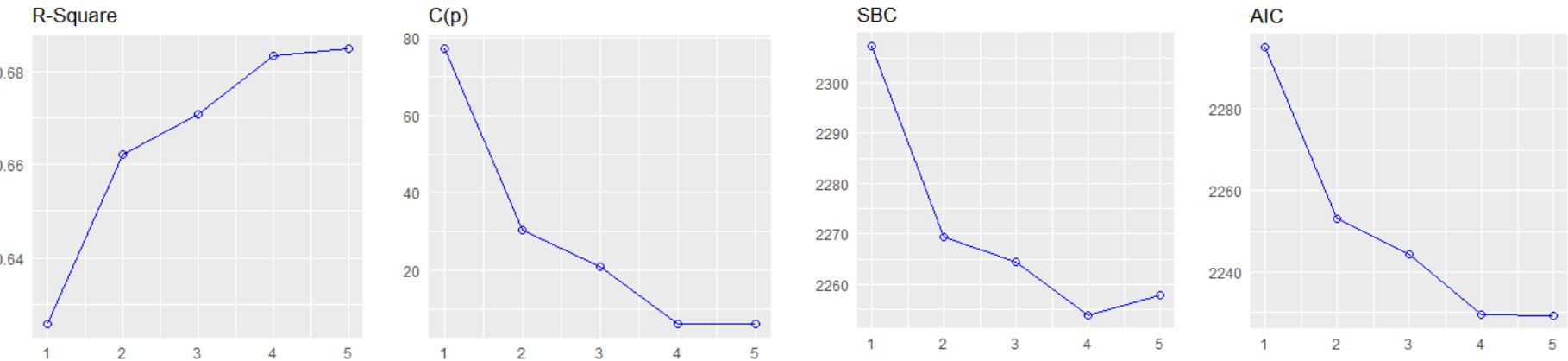
- Выбор лучшей модели в семействе по разным критериям

```
> model <- lm(MPG_Highway ~ Weight + Length +
+                 Invoice + Horsepower + EngineSize, data = cars)
> k <- ols_step_best_subset(model, metric = "AIC")
```

```
> k
```

Best Subsets Regression		Subsets Regression Summary						
Model Index	Predictors	Model	R-Square	Adj. R-Square	Pred R-Square	C(p)	AIC	SBIC
1	Weight	1	0.6257	0.6248	0.6203	77.2926	2295.0605	1079.8215
2	Weight Horsepower	2	0.6622	0.6606	0.656	30.4150	2253.1520	1038.2110
3	Weight Length Horsepower	3	0.6707	0.6683	0.6626	21.0118	2244.2258	1029.3767
4	Weight Length Invoice Horsepower	4	0.6832	0.6802	0.6725	6.2025	2229.5935	1015.0715
5	Weight Length Invoice Horsepower EngineSize	5	0.6849	0.6811	0.6708	6.0000	2229.3655	1014.9243

```
> plot(k)
```



Смещенные регуляризованные модели

- Регуляризация в пространстве параметров:

$$B^{ridge} = \arg \min_B \left\{ \sum_{i=1}^N (y_i - b_0 - \sum_{j=1}^p x_{ij} b_j)^2 + C \sum_{j=1}^p (b_j)^2 \right\}$$

Точность приближения

Штраф за сложность модели

- Решение (в матричном виде):

$$B^{ridge} = (X^T X + CI)^{-1} X^T \bar{y}$$

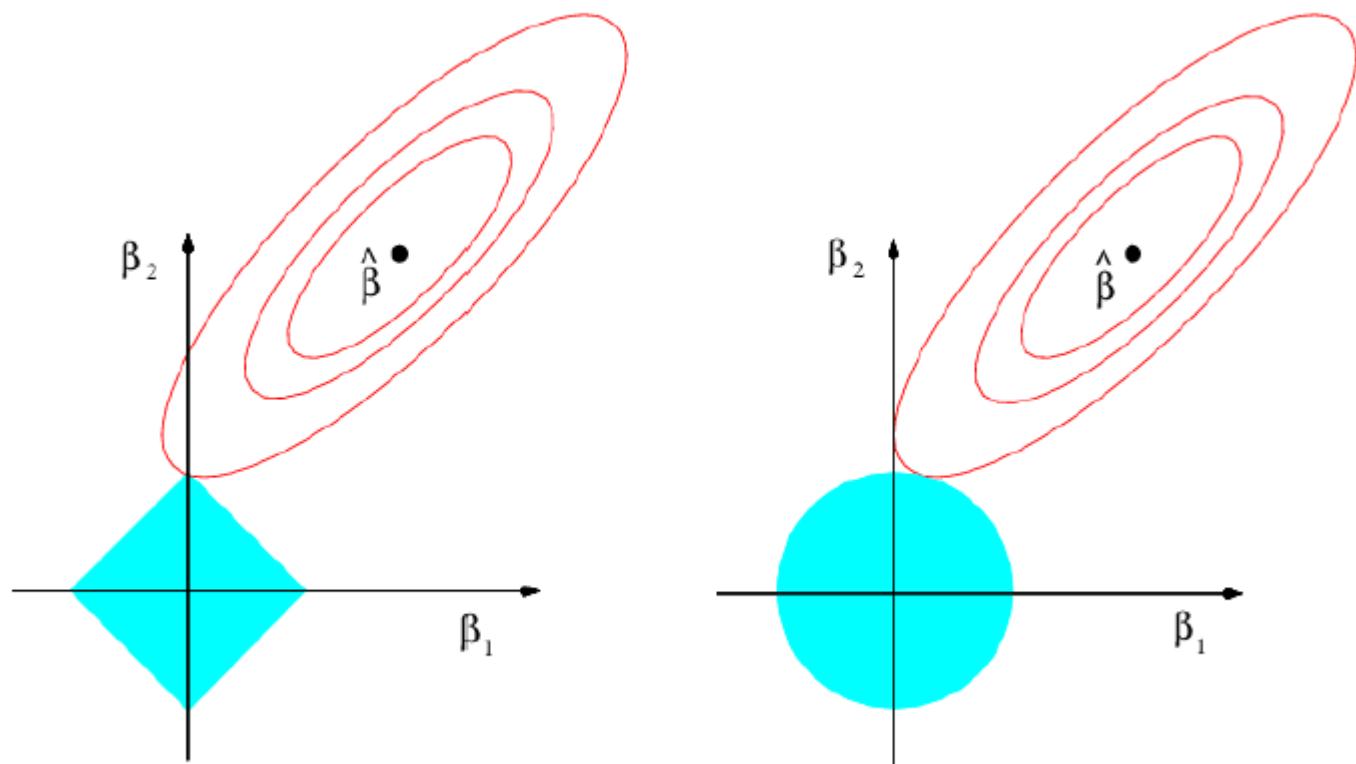
- Метод Лассо:

- Аналогично, но штраф модуля:

$$B^{lasso} = \arg \min_B \left\{ \sum_{i=1}^N (y_i - b_0 - \sum_{j=1}^p x_{ij} b_j)^2 \right\}, \sum_{j=1}^p |b_j| \leq C$$

Штраф

Иллюстрация регуляризации



Гребневая регрессия

- Основные проблемы:
 - подбор параметра регуляризации, не «обнуляет» незначимые коэф., дает смещенную оценку, не всегда корректные оценки для коэф., интервалов, ошибок и т.д.
- задается перебором параметра Ridge:

```
lmridge(formula, data, K = 0, scaling=c("sc",
  "scaled", "centered"), ...)
```

```
> rstats1(mod)
```

```
Ridge Regression Statistics 1:
```

	Variance	Bias^2	MSE	rsigma2	F	R2	adj-R2
K=0	37644570574	0	37644570574	82029139	92.8009	0.7450	0.7376
K=1	268779873	84133660627	84402440500	131731369	57.7871	0.3182	0.2984
K=2	140485339	102163004395	102303489734	158943092	47.8937	0.2148	0.1920
K=3	90486564	111173123515	111263610079	177166734	42.9673	0.1603	0.1359
K=4	64435834	116762371177	116826807011	190707088	39.9166	0.1258	0.1004
K=5	48771565	120625344624	120674116189	201390027	37.7992	0.1019	0.0758
K=6	38472354	123479618744	123518091098	210142720	36.2248	0.0845	0.0579
K=7	31270208	125687239855	125718510063	217500274	34.9994	0.0713	0.0443
K=8	26001070	127452594236	127478595306	223800906	34.0141	0.0610	0.0337
K=9	22010586	128900615992	128922626578	229273287	33.2022	0.0528	0.0253
K=10	18904841	130112370037	130131274878	234080317	32.5204	0.0462	0.0185

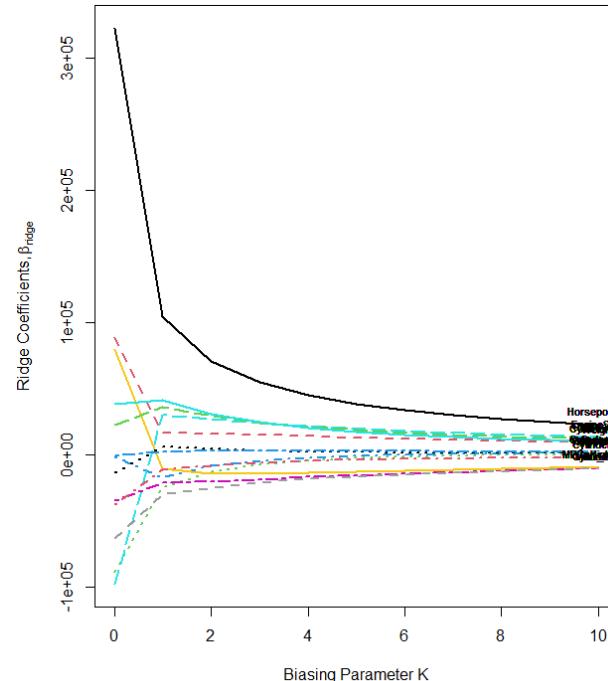
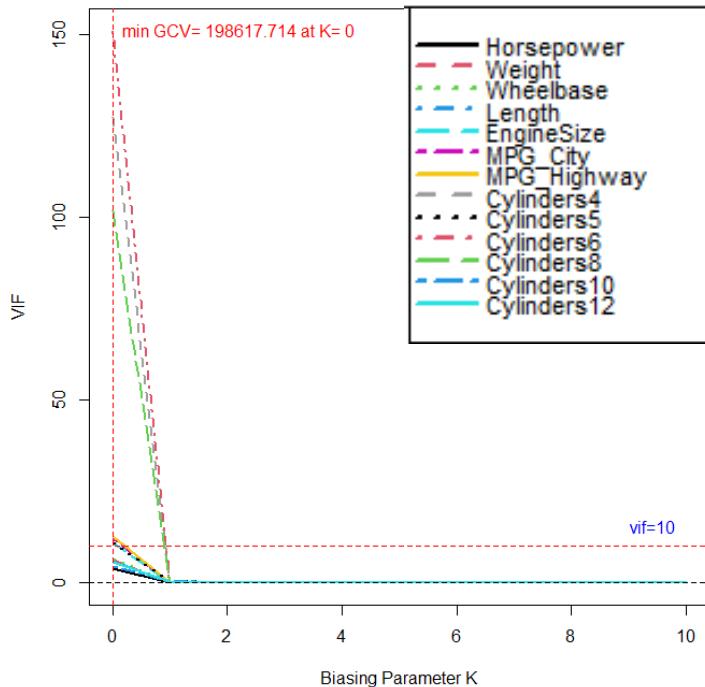
```
> mod <- lmridge(Invoice ~ Horsepower + Weight +
+                           Wheelbase + Length + EngineSize +
+                           MPG_City + MPG_Highway + Cylinders,
+                           cars, K = seq(0, 10, 1))
```

```
> t(mod$coef)
      Horsepower     Weight     Wheelbase      Length
K=0    322799.26  89149.099 -88480.26841 -1232.8985
K=1    104252.79  16888.518 -24844.66225 -17048.3494
K=2    70649.80   16046.368 -12602.75331 -8466.7403
K=3    54805.88   15070.787 -7035.63001 -4264.1339
K=4    45235.45   14038.906 -4046.13365 -1983.4987
K=5    38714.81   13061.603 -2278.55947 -647.0564
K=6    33939.49   12172.940 -1165.55394  177.7427
K=7    30269.14   11376.674 -433.39835  704.5767
K=8    27348.37   10666.137       63.48785  1048.3002
K=9    24962.34   10031.768      408.28010  1274.7758
K=10   22972.69   9463.901      651.18743  1423.7377
```

Гребневая регрессия

```
> mod <- lmridge(Invoice ~ Horsepower + Weight +
+                  Wheelbase + Length + EngineSize +
+                  MPG_City + MPG_Highway + Cylinders,
+                  cars, K = seq(0, 10, 1))
>
> ## VIF trace
> plot(mod, type = "vif")
> ## Ridge trace without abline
> plot(mod, type = "ridge", abline = FALSE)
```

```
> t(mod$coef)
            Horsepower      Weight
K=0    322799.26  89149.099
K=1    104252.79 16888.518
K=2     70649.80 16046.368
K=3     54805.88 15070.787
K=4     45235.45 14038.906
K=5     38714.81 13061.603
K=6     33939.49 12172.940
K=7     30269.14 11376.674
K=8     27348.37 10666.137
K=9     24962.34 10031.768
K=10    22972.69  9463.901
```



Гребневая регрессия: масштабирование предикторов

- Оценки коэффициентов стандартным методом наименьших квадратов являются масштабируемыми: умножая X_j на константу c просто приводит к масштабированию оценок коэффициентов наименьших квадратов на коэффициент. Другими словами, независимо от того, как масштабируется j -ый предиктор, останется прежним. $X_j \hat{\beta}_j$
- Оценки коэффициентов гребневой регрессии наоборот могут существенно измениться при умножении заданного предиктора на константу, из-за суммы квадратов коэффициентов в штрафной части целевой функции регрессии.
- Поэтому, лучше всего применять гребневую-регрессию после *стандартизации предикторов*, используя формулу

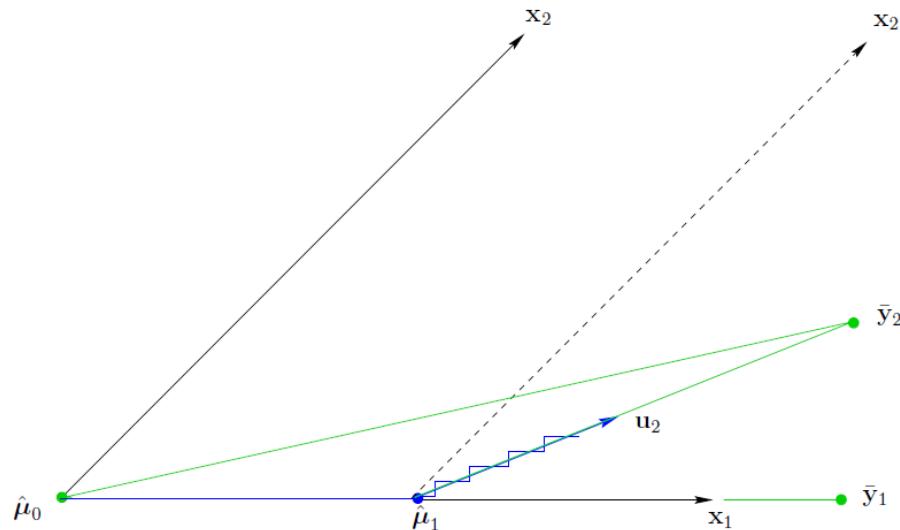
$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}}$$

Алгоритм LAR

- Суть LAR:
 - вместо последовательного добавления переменных, каждый раз минимизируя целевую функцию, на каждом шаге изменяет веса уже добавленных переменных в направлении биссектрисы между ними и новым добавляемым предиктором с учетом регрессионных остатков
- Алгоритм:
 - Сначала все переменные стандартизуются, а отклик центрируется, и все коэффициенты равны 0.
 - Находится предиктор наиболее коррелирующий с откликом и делается максимальный шаг (рассчитывается его коэффициент) по направлению этого предиктора пока не найдется другой предиктор с такой же корреляцией с остатком, как уже добавленный
 - Второй предиктор добавляется в модель и делается максимальный шаг в направлении биссектрисы между добавленными предикторами, пока не найдется следующий предиктор, коррелирующий с остатком так же как уже добавленные
 - Аналогично находится третий предиктор и так далее.

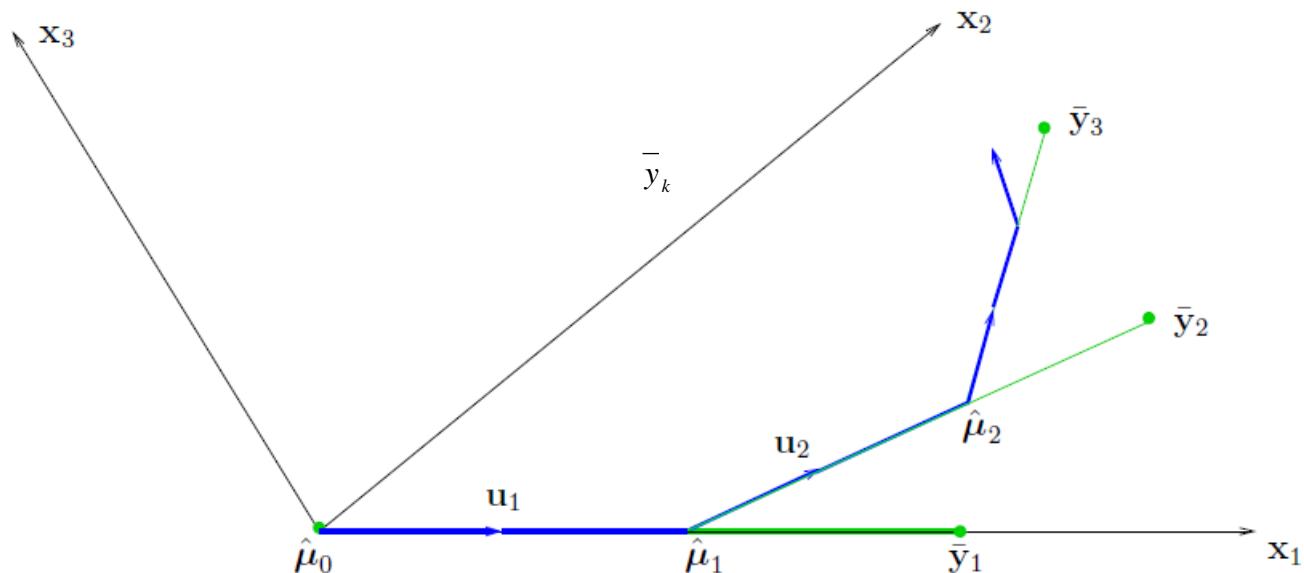
LAR: Двухмерный пример

- LARS $m = 2$ предиктора.
- \bar{y}_2 проекция y в $L(x_1, x_2)$ - решение.
- Начинаем $\hat{\mu}_0 = 0$, вектор остатков $\bar{y}_2 - \hat{\mu}_0$ больше коррелирует с x_1 чем с x_2 .
- LARS находит $\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 x_1$ где $\hat{\gamma}_1$ выбран так что $\bar{y}_2 - \hat{\mu}_1$ биссектриса угла между x_1 и x_2 .
- Далее $\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 u_2$ где u_2 единичный вектор вдоль биссектрисы; $\hat{\mu}_2 = \bar{y}_2$.



LAR: Трехмерный пример

На каждом шаге LARS оценивает $\hat{\mu}_k$ сходясь в результате к МНК решению \bar{y}_k

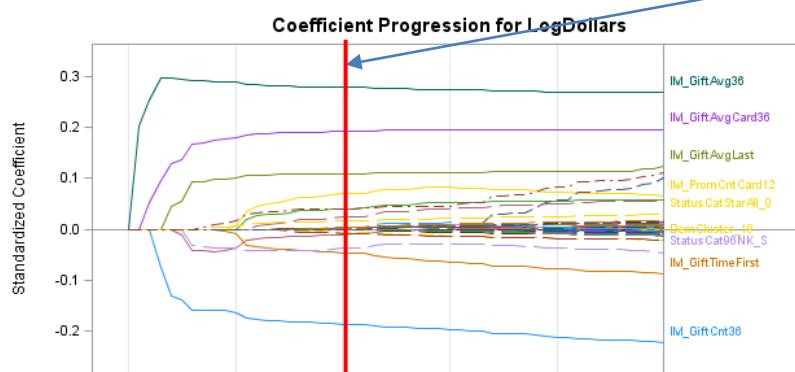


LAR и LASSO

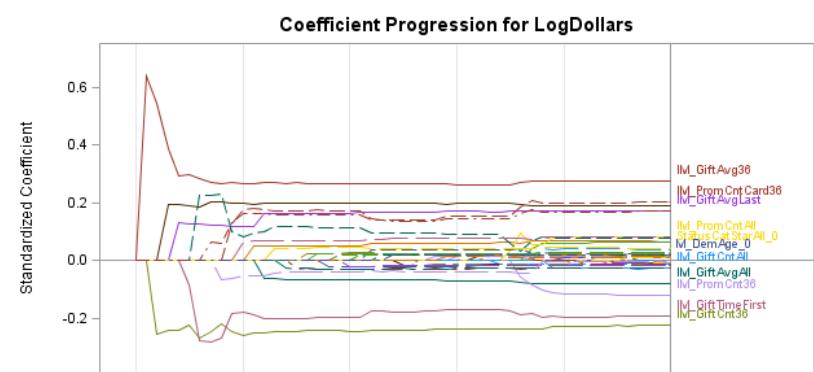
- До появления LAR LASSO (как и RIDGE) требовал перебора константы регуляризации и решения оптим. задачи кв. программирования
- Но LAR позволяет пошаговым методом перебрать все оптимальные значения константы регуляризации C , поскольку каждому шагу в LAR соответствует свое допустимое значение C и соответственно подмножество предикторов с фиксированным набором ненулевых коэффициентов, а все возможные решения LASSO получаются линейной интерполяцией набора LASSO решений, соответствующих шагам LAR

$$B^{lasso} = \arg \min_B \left\{ \sum_{i=1}^N (y_i - b_0 - \sum_{j=1}^p x_{ij} b_j)^2 \right\}, \sum_{j=1}^p |b_j| \leq C$$

LAR:



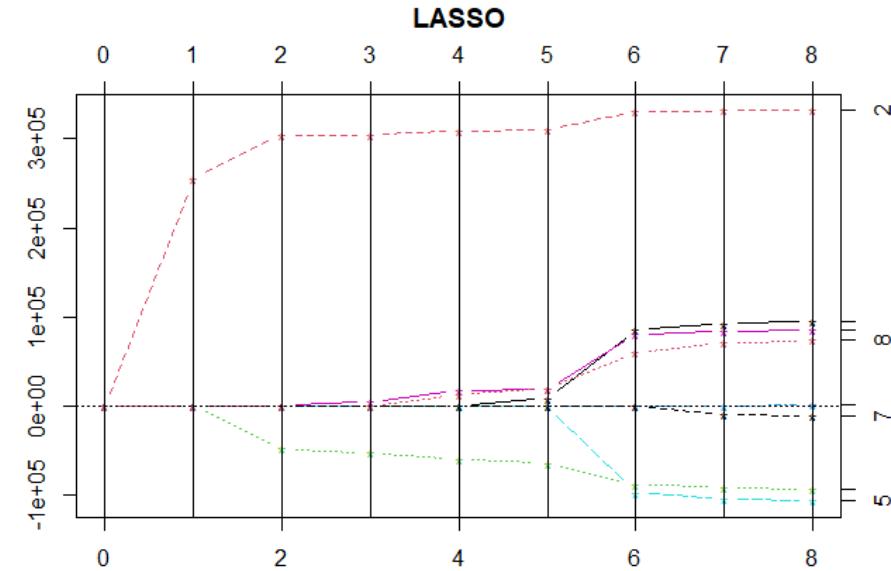
Forward Stepwise:



Пример LAR и LASSO

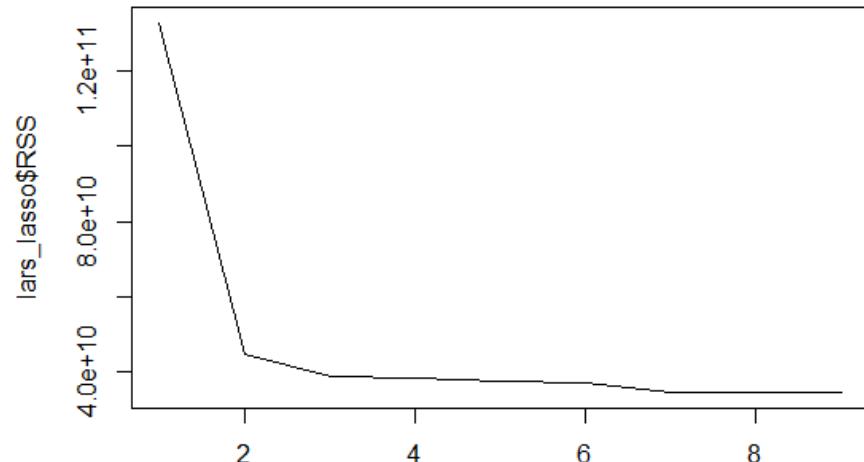
```
> cars_ <- na.omit(cars)
> feats <- c("Weight", "Horsepower", "Wheelbase", "Length",
+           "EngineSize", "Cylinders", "MPG_City", "MPG_Highway")
> lars_lasso <- lars(as.matrix(cars_[,feats]),
+                      cars_$Invoice, type="lasso")
R-squared: 0.74
Sequence of LASSO moves:
  Horsepower Wheelbase Cylinders MPG_Highway Weight EngineSize MPG_City Length
Var      2          3          6          8          1          5          7          4
Step     1          2          3          4          5          6          7          8
```

```
> plot(lars_lasso, xvar = "step", breaks = TRUE)
```



```
> summary(lars_lasso)
      Df    Rss      Cp
0  1 1.3284e+11 1177.0721
1  2 4.4654e+10 116.1987
2  3 3.8830e+10 48.0073
3  4 3.8535e+10 46.4498
4  5 3.7698e+10 38.3645
5  6 3.7344e+10 36.0965
6  7 3.4633e+10 5.4195
7  8 3.4600e+10 7.0295
8  9 3.4598e+10 9.0000
```

```
> plot(lars_lasso$RSS, type = "l")
```



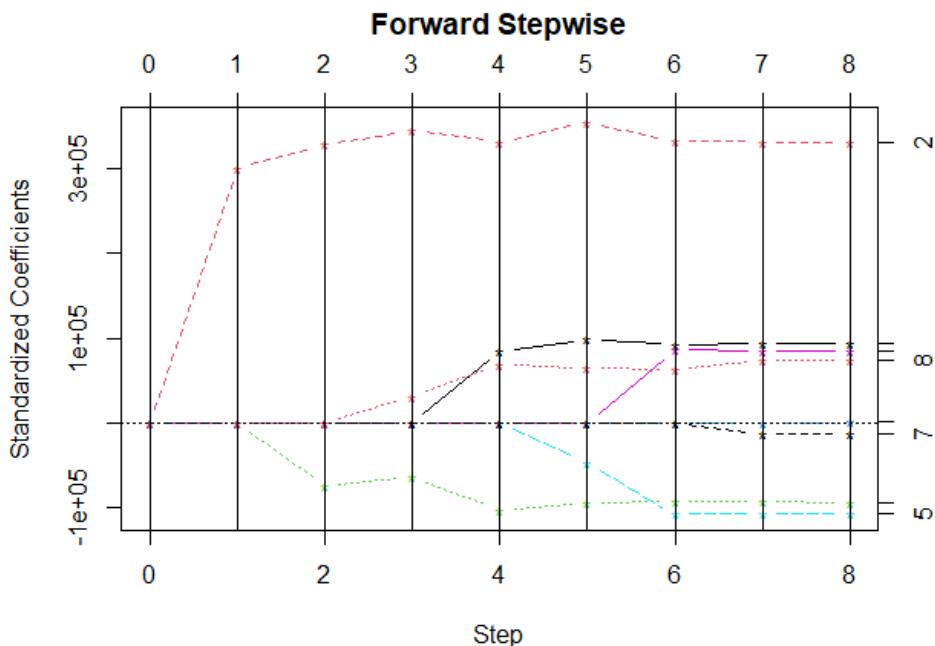
```
lars(x, y, type = c("lasso", "lar", "forward.stagewise", "stepwise"),
      trace = FALSE, normalize = TRUE, intercept = TRUE, Gram, eps =
      1e-12, max.steps, use.Gram = TRUE)
```

Пример stepwise LAR не LASSO

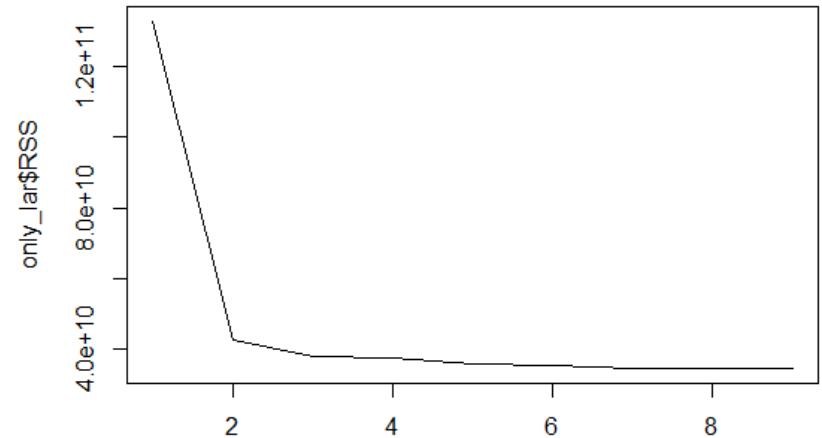
```
> only_lar <- lars(as.matrix(cars_[,feats]),
+                     cars_$Invoice, type="stepwise")
>
> only_lar
R-squared: 0.74
Sequence of Forward Stepwise moves:
  Horsepower Wheelbase MPG_Highway Weight EngineSize Cylinders MPG_City Length
Var          2           3           8           1           5           6           7           4
Step         1           2           3           4           5           6           7           8
```

```
> summary(only_lar)
      Df      Rss      Cp
0 1 1.3284e+11 1177.0721
1 2 4.2627e+10 91.7749
2 3 3.8036e+10 38.4397
3 4 3.7544e+10 34.5049
4 5 3.6181e+10 20.0817
5 6 3.5749e+10 16.8761
6 7 3.4613e+10 5.1854
7 8 3.4598e+10 7.0027
8 9 3.4598e+10 9.0000
```

```
> plot(only_lar, xvar = "step", breaks = TRUE)
```



```
> plot(only_lar$RSS, type = "l")
```



Преобразование предикторов для уменьшения корреляции

- Использовать РСА (Principal Component Regression) :
 - для перехода в новое пространство независимых ортогональных признаков меньшей размерности:

$$X^p \rightarrow Z^M = (z_1, \dots, z_M), M \ll p, z_1 = Xv_m$$

- Поскольку ортогональны, то просто сумма M одномерных задач регрессии:

$$f(z) = \bar{y} + \sum_{m=1}^M \theta_m z_m$$

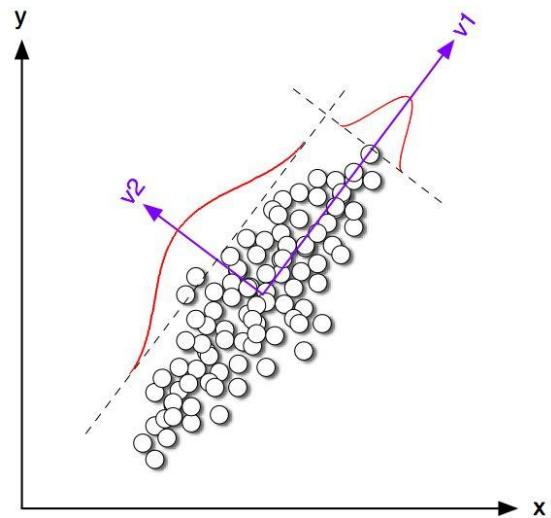
Среднее по исходному
отклику

- где

$$\theta_m = \langle z_m, y \rangle / \langle z_m, z_m \rangle$$

Общая идея РСА

- Строится новый базис (линейное преобразование исходного пространства) такой, что:
 - Центр координат совпадает с мат. ожиданием наблюдений
 - Первый вектор направлен таким образом, что дисперсия вдоль него была максимальной
 - Каждый последующий вектор ортогонален предыдущим и направлен по направлению максимальной дисперсии
 - Последние компоненты – не важны!!!
- Формально:
$$\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|=1} E\{(\mathbf{w}^T \mathbf{x})^2\}$$
$$\mathbf{w}_k = \arg \max_{\|\mathbf{w}\|=1} E\{[\mathbf{w}^T (\mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w}_i \mathbf{w}_i^T \mathbf{x})]^2\}$$
- - SVD разложение матрицы данных
 - Собственные значения ковариационной матрицы



Поиск собственных значений и собственных векторов ковариационной матрицы в РСА

- Рассчитаем ковариационную матрицу:
 - Ковариация = 0 – независимы
 - Ковариация > 0 – вместе растут и убывают
 - Ковариация < 0 – противофаза
- Проблема с.зн.:
 $\mathbf{C} * \mathbf{v} = \lambda * \mathbf{v}$

решение: поиск корней

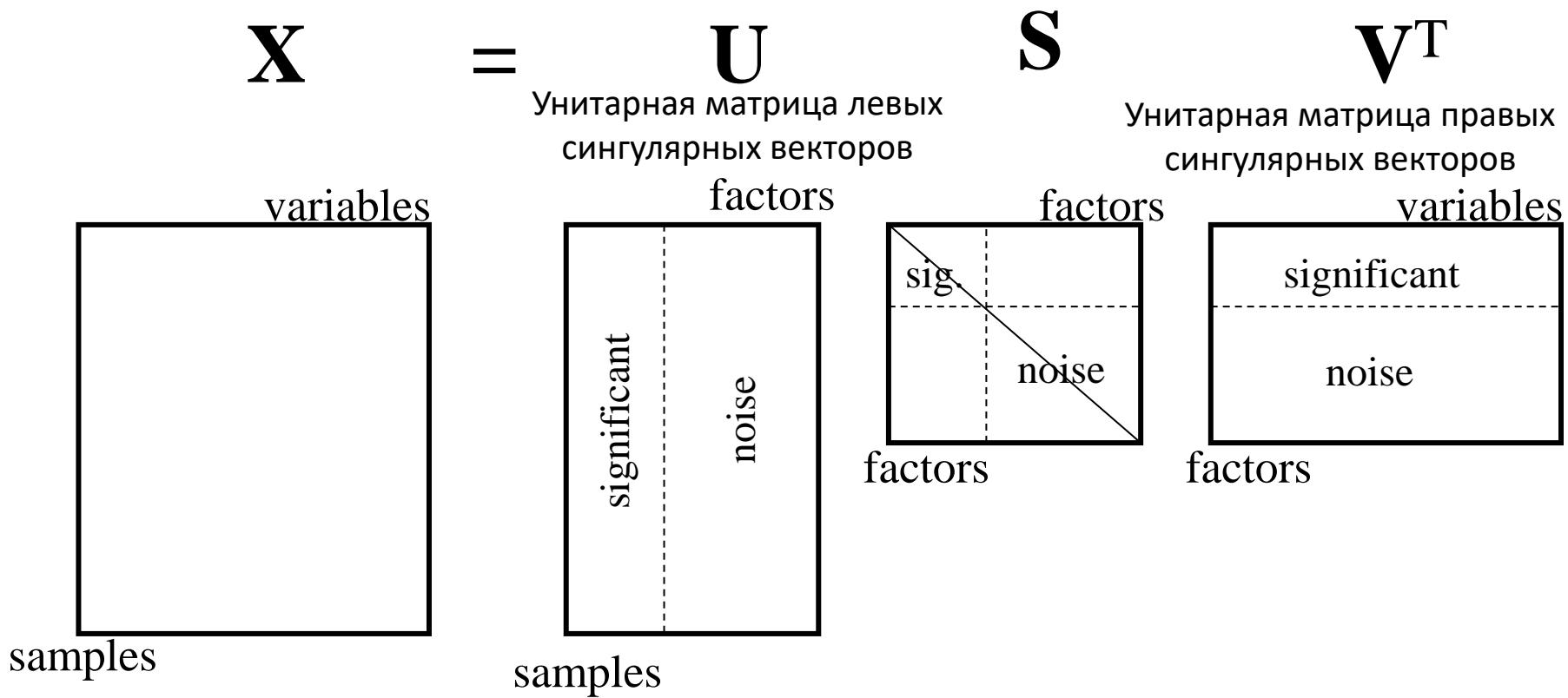
$$|\mathbf{C} - \lambda \cdot \mathbf{I}| = 0$$

матрица положительно определенная – есть вещественные корни

- Результат:
 - λ – дисперсии
 - С.в. – главные компоненты

$$\mathbf{C} = \begin{pmatrix} \text{cov}(x^1, x^1) & \text{cov}(x^1, x^2) & \dots & \text{cov}(x^1, x^d) \\ \text{cov}(x^2, x^1) & \text{cov}(x^2, x^2) & \dots & \text{cov}(x^2, x^d) \\ \dots & \dots & \dots & \dots \\ \text{cov}(x^d, x^1) & \text{cov}(x^d, x^2) & \dots & \text{cov}(x^d, x^d) \end{pmatrix}$$

Сингулярное разложение в РСА



SVD разложение и обратная проекция

- SVD разложение матрицы X: $X_{n \times m} = U_{n \times n} D_{n \times m} V_{m \times m}^T$
- SVD приближение (метод главных компонент):
 - отбрасываются с.в., соотв. наименьшим с.з.
 - остается р-я часть главных с.в., которые характеризуют основные зависимости в X
 - с их помощью приближается исходная матрица:

$$\min_{U_p, D_p, V_p} \|X - U_p D_p V_p^T\|$$

$$X^{(l+1)} = V_p V_p^T X^{(l)}$$

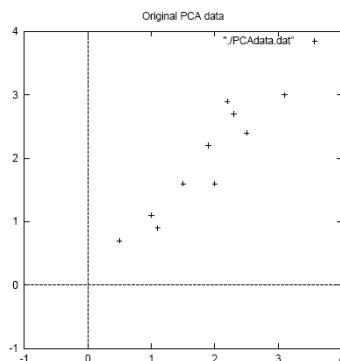


Figure 3.1: PCA example data, original data on the left, data with the means subtracted on the right, and a plot of the data

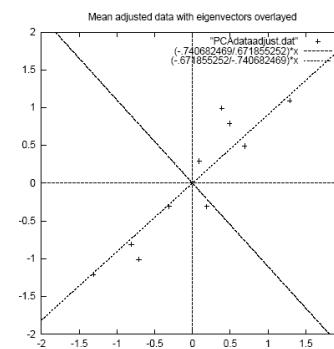


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlaid on top.

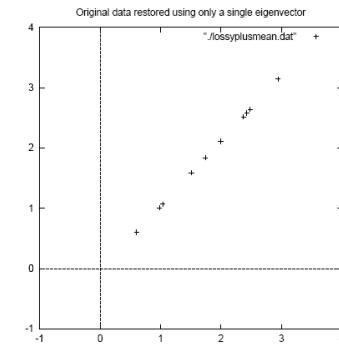


Figure 3.5: The reconstruction from the data that was derived using only a single eigenvector

PRCOMP (анализ с.зн.)

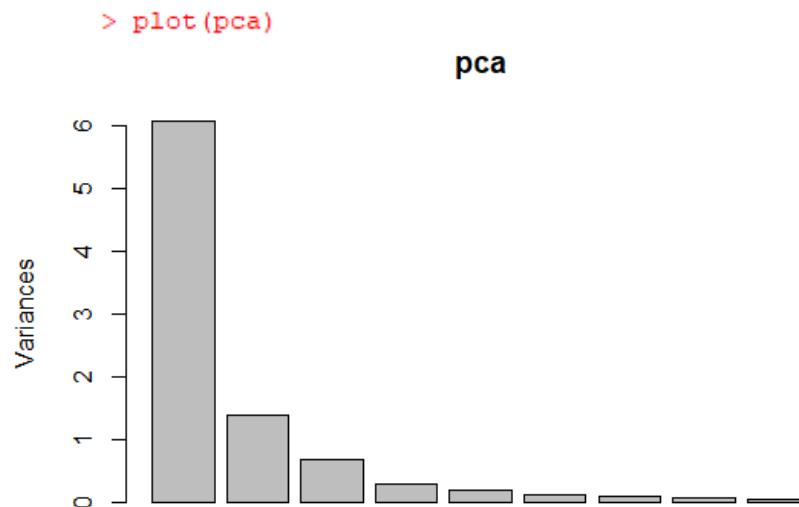
```
prcomp(formula, data = NULL, subset, na.action, ...)
prcomp(x, retx = TRUE, center = TRUE, scale. = FALSE, ...)
```

```
> pca <- prcomp(~ Invoice + Weight + Horsepower +
+                 Wheelbase + Length + EngineSize +
+                 Cylinders + MPG_City + MPG_Highway,
+                 data = cars, scale = TRUE)

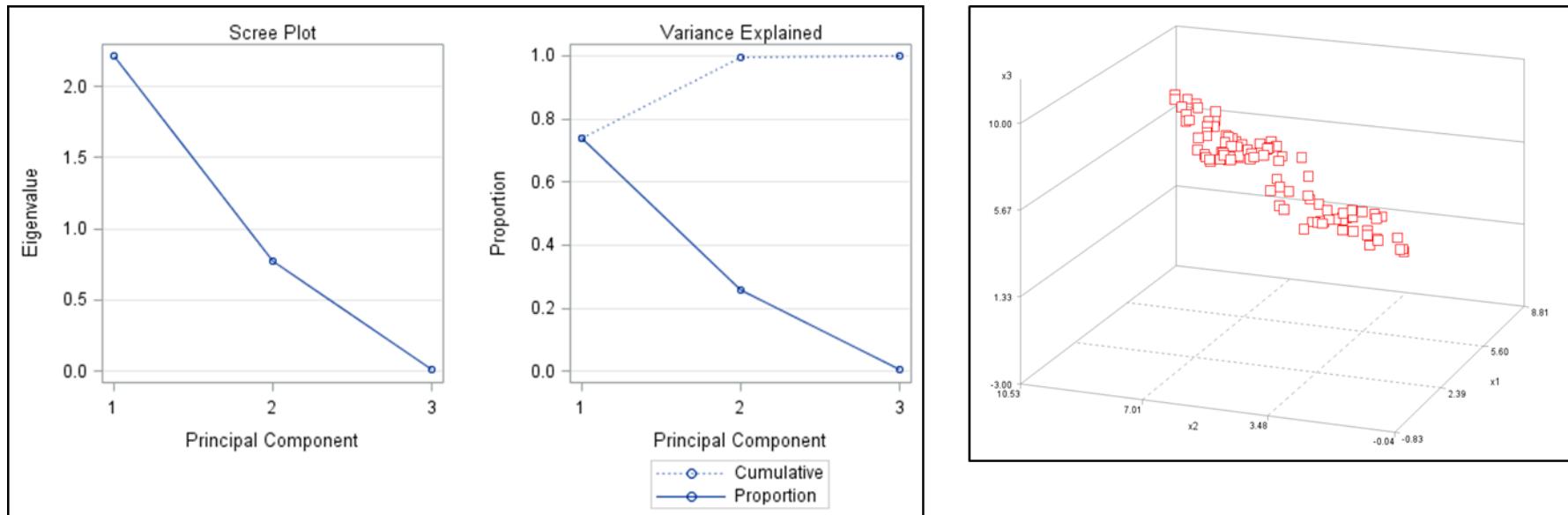
> summary(pca)
Importance of components:

PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9
Standard deviation   2.465 1.1845 0.83473 0.5441 0.45551 0.34838 0.30044 0.26267 0.2035
Proportion of Variance 0.675 0.1559 0.07742 0.0329 0.02305 0.01349 0.01003 0.00767 0.0046
Cumulative Proportion 0.675 0.8308 0.90826 0.9412 0.96422 0.97770 0.98773 0.99540 1.0000
```

```
> eigenvalues <- pca$sdev^2
> difference <- c(abs(diff(eigenvalues)), NA)
> proportion <- eigenvalues/sum(eigenvalues)
> cumulative <- cumsum(proportion)
>
> data.frame(eigenvalues = eigenvalues,
+             difference = difference,
+             proportion = proportion,
+             cumulative = cumulative)
  eigenvalues difference proportion cumulative
1  6.07458835 4.67156520 0.674954261 0.6749543
2  1.40302315 0.70625053 0.155891461 0.8308457
3  0.69677262 0.40068744 0.077419180 0.9082649
4  0.29608518 0.08859551 0.032898354 0.9411633
5  0.20748967 0.08611842 0.023054408 0.9642177
6  0.12137125 0.03110411 0.013485695 0.9777034
7  0.09026715 0.02126978 0.010029683 0.9877330
8  0.06899737 0.02759211 0.007666374 0.9953994
9  0.04140526      NA 0.004600584 1.0000000
```



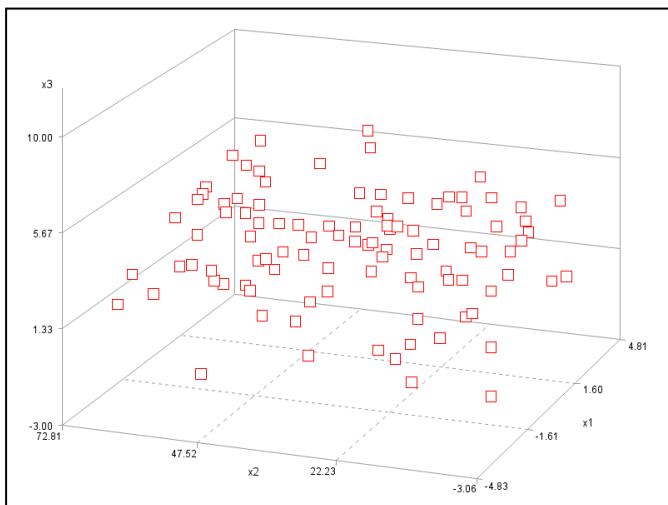
Пример 1



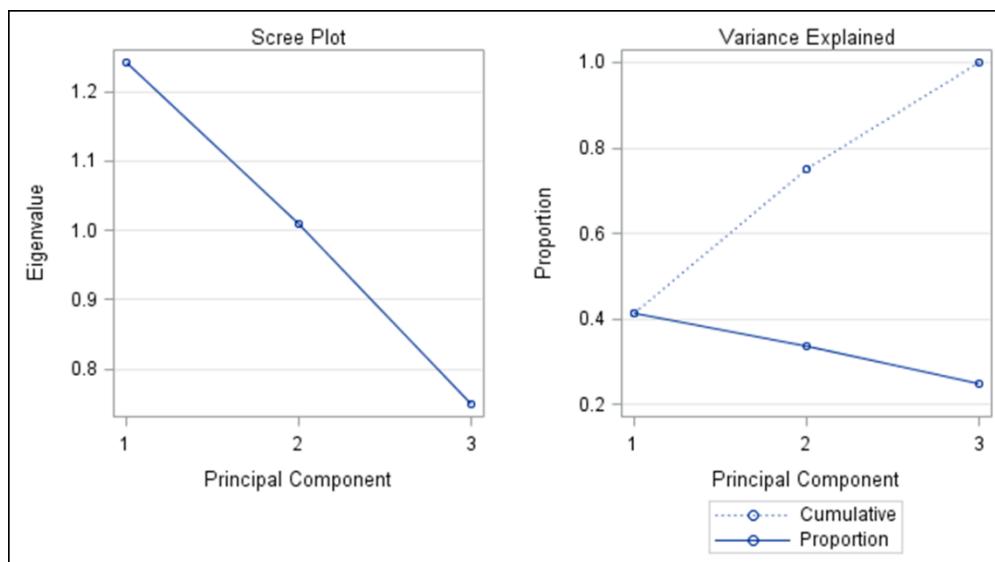
Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	2.21358154	1.44403082	0.7379	0.7379
2	0.76955072	0.75268297	0.2565	0.9944
3	0.01686775		0.0056	1.0000

Eigenvectors			
	Prin1	Prin2	Prin3
x1	0.650940	0.263685	0.711862
x2	0.645235	0.301851	-.701825
x3	-.399937	0.916164	0.026348

Пример 2



Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	1.24205697	0.23274599	0.4140	0.4140
2	1.00931098	0.26067894	0.3364	0.7505
3	0.74863205		0.2495	1.0000



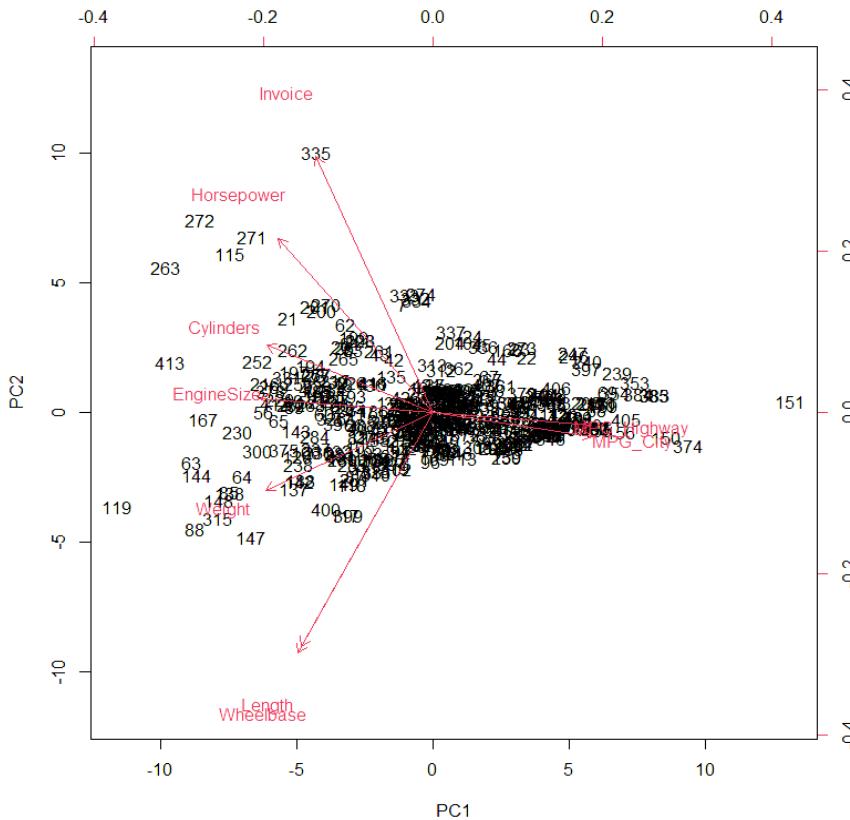
Eigenvectors			
	Prin1	Prin2	Prin3
x1	0.704619	-0.156546	0.692102
x2	0.708942	0.113759	-0.696032
x3	0.030228	0.981097	0.191139

PRCOMP (анализ с.зн.)

```
> -1*pca$rotation
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Invoice	0.2552471	-0.54641743	0.32903272	-0.52793821	-0.13541745	0.37829088	-0.08071216	-0.28506199	-0.008885885
Weight	0.3661241	0.16558925	-0.08486733	-0.03044575	-0.77487827	0.06160837	0.30678915	0.32689949	0.158711005
Horsepower	0.3389224	-0.37097220	0.18986204	-0.10828483	0.19592420	-0.71437747	0.13222971	0.35801645	-0.073185522
Wheelbase	0.2961067	0.51252917	0.21273076	-0.22518388	-0.11010667	-0.32073559	-0.59321443	-0.26413918	0.130402630
Length	0.2885281	0.49868494	0.29193248	-0.24539864	0.43001639	0.30024213	0.39670161	0.18315243	-0.237998715
EngineSize	0.3769966	-0.03047054	0.14712245	0.51138412	0.03467458	-0.09626338	0.37730343	-0.63630056	0.126811441
Cylinders	0.3636287	-0.14398536	0.20908216	0.55473389	0.08341815	0.35120753	-0.46729437	0.37653934	-0.071736358
MPG_City	-0.3472769	0.05098788	0.54796099	0.15949823	-0.36955170	-0.13002962	0.03386051	-0.05864567	-0.626925843
MPG_Highway	-0.3464859	0.02747729	0.59362181	0.05493025	0.03274970	0.01508361	0.10362096	0.17403074	0.693876246

```
> biplot(pca)
```

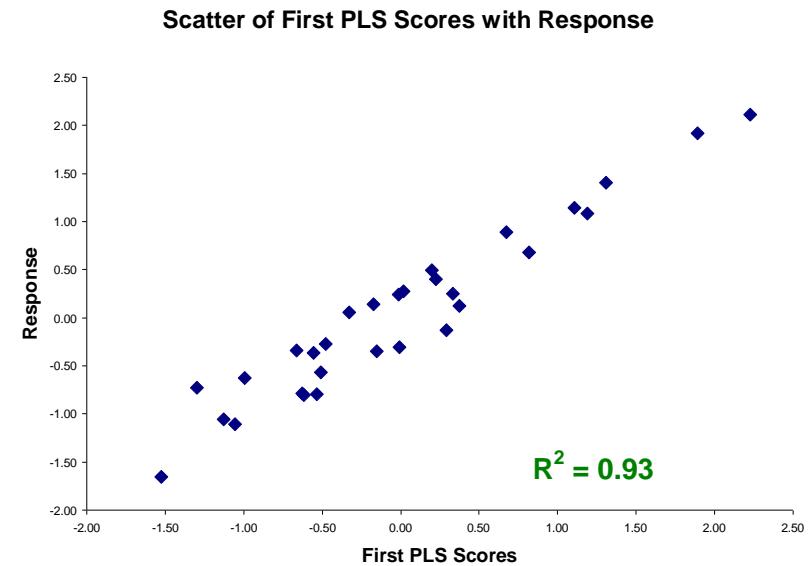
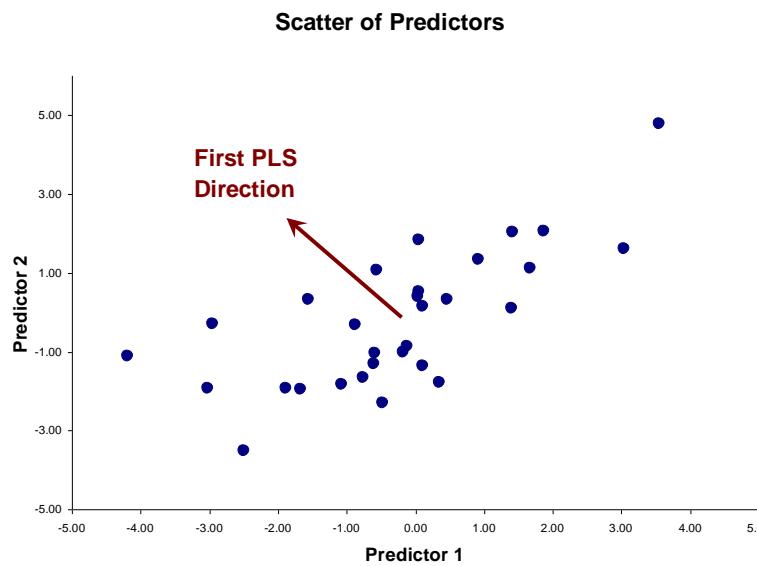


PLS регрессия

Последовательный поиск скрытых факторов (латентных переменных), таких что:

$$\max_{|\alpha|=1, v_l^T S \alpha = 0, l=1, \dots, m-1} \text{Corr}^2(y, X\alpha) / \text{Var}(X\alpha)$$

Число факторов определяет сложность модели

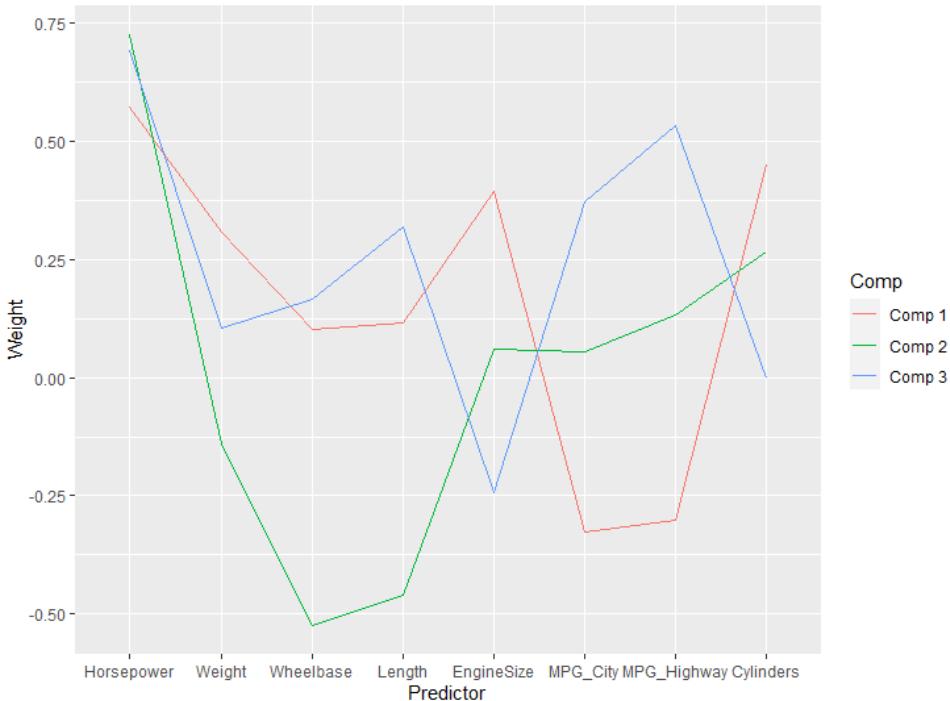


PLS регрессия

```
> model_pls <- plsr(Invoice ~ Horsepower + Weight +
+                     Wheelbase + Length + EngineSize +
+                     MPG_City + MPG_Highway + Cylinders,
+                     ncomp = 3, scale=TRUE, data = cars)
```



```
> df <- melt(model_pls$projection,
+             id.vars = 'Predictor',
+             variable.name = 'series')
> colnames(df) <- c("Predictor", "Comp", "Weight")
> ggplot(df, aes(Predictor, Weight,
+                  group=Comp, colour = Comp)) +
+   geom_line()
```

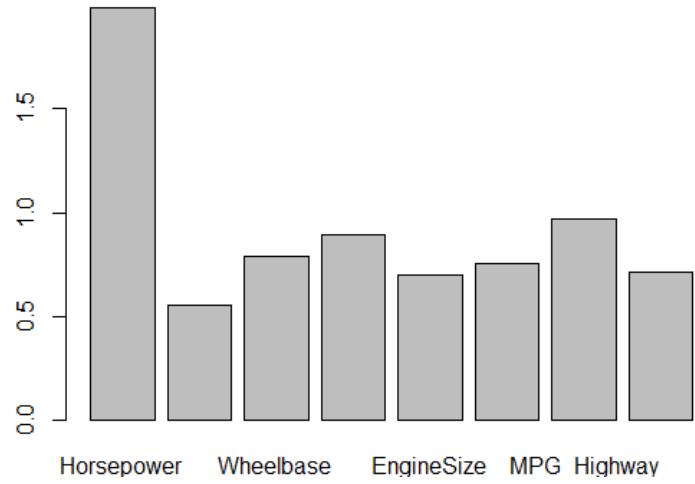


```
> summary(model_pls)
Data: X dimension: 426 8
Y dimension: 426 1
Fit method: kernelpls
Number of components considered: 3
TRAINING: % variance explained
      1 comps  2 comps  3 comps
X       69.87    83.86   91.10
Invoice 41.57    65.99   70.94
```

Вариация
предиктора

Вариация
отклика

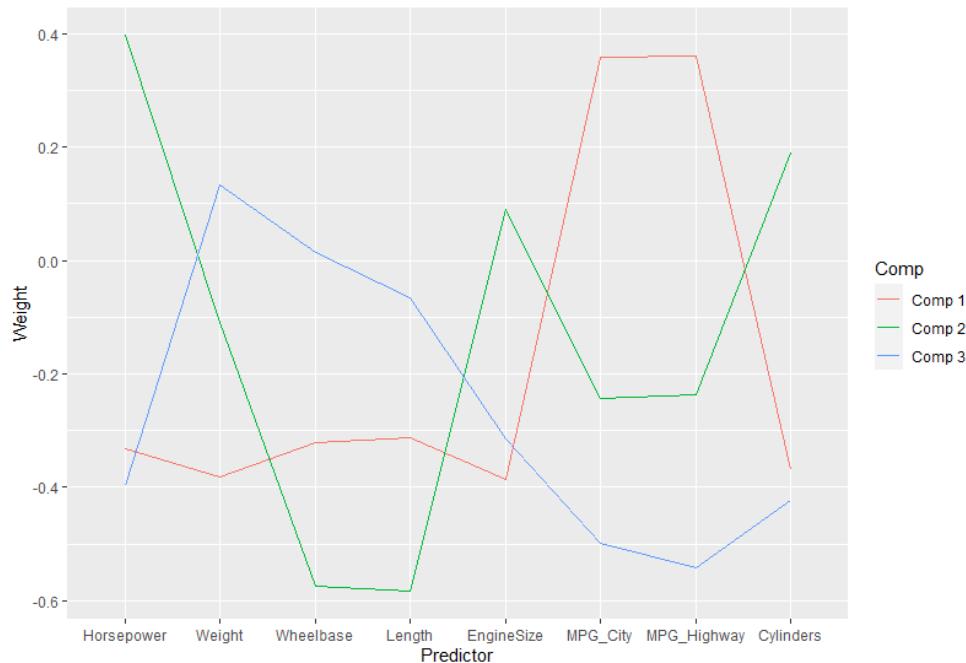
```
> barplot(rowSums(abs(model_pls$projection)),
+          main = "Variable Importance")
Variable Importance
```



PCR регрессия

```
> model_pcr <- pcr(Invoice ~ Horsepower + Weight +
+                     Wheelbase + Length + EngineSize +
+                     MPG_City + MPG_Highway + Cylinders,
+                     ncomp = 3, scale = TRUE, data = cars)

> df <- melt(model_pcr$projection,
+              id.vars = 'Predictor',
+              variable.name = 'series')
> colnames(df) <- c("Predictor", "Comp", "Weight")
>
> ggplot(df, aes(Predictor, Weight,
+                  group=Comp, colour = Comp)) +
+   geom_line()
```



Факторы и важность совсем другие!!!

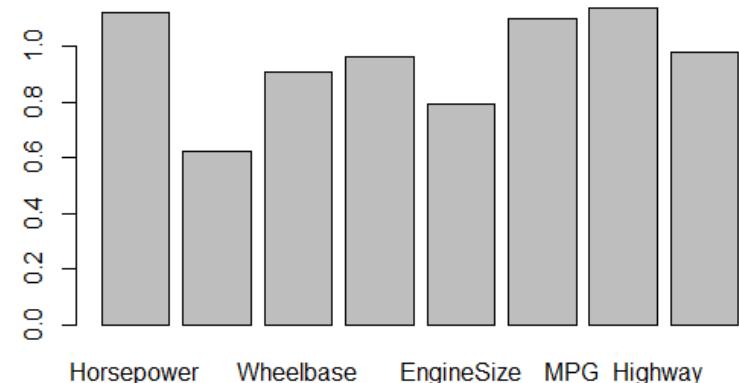
```
> summary(model_pcr)
Data: X dimension: 426 8
Y dimension: 426 1
Fit method: svdpc
Number of components considered: 3
TRAINING: % variance explained
  1 comps  2 comps  3 comps
X        71.56   84.94   92.38
Invoice 30.61   52.80   63.88
```

Вариация
предиктора
лучше чем у PLS

Вариация
отклика хуже
чем у PLS

```
> barplot(rowSums(abs(model_pcr$projection)),
+           main = "Variable Importance")
```

Variable Importance



Кластеризация переменных

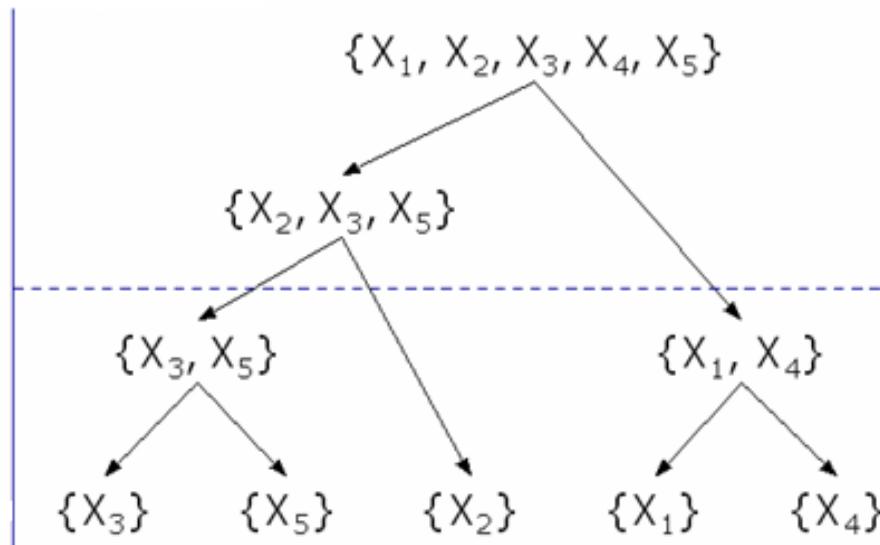
- У PCR и PLS регрессий существенный недостаток – не интерпретируемый результат
- Задачи процедуры кластеризации переменных:
 - группировка переменных в иерархические кластеры так, чтобы в одном кластере переменные были **максимально коррелированы**, а кластеры между собой нет
 - Затем выбирается либо первая гл. компонента кластера либо лучший представитель кластера



Суть алгоритма группировки переменных

- Восходящая иерархическая кластеризация с помощью стандартной процедуры hclust

```
varclus(x, similarity = c("spearman", "pearson", "hoeffding", "bothpos",  
"ccbothpos"), type=c("data.matrix", "similarity.matrix"),...)
```



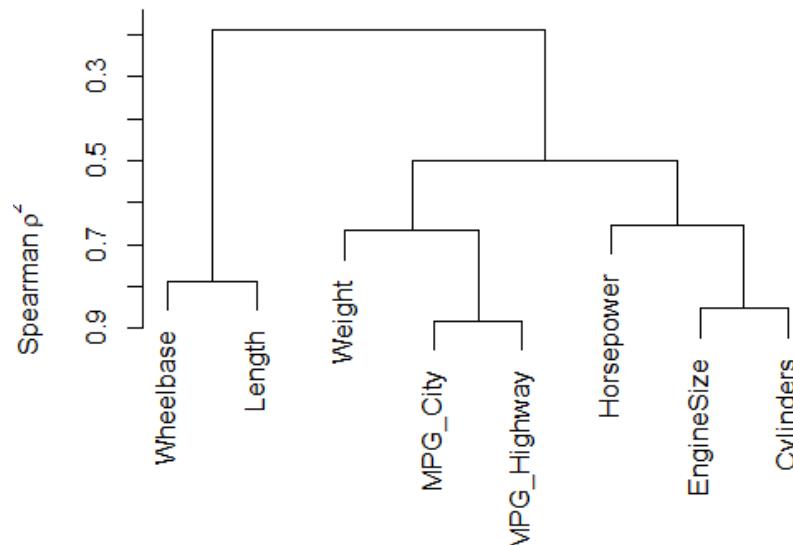
При менение VARCLUS

```
> vc <- varclus(~Horsepower + Weight + Wheelbase +
+                 Length + EngineSize + MPG_City +
+                 MPG_Highway + Cylinders, data=cars)
>
> print(vc)
```

Similarity matrix (Spearman rho^2)

	Horsepower	Weight	Wheelbase	Length	EngineSize	MPG_City	MPG_Highway	Cylinders
Horsepower	1.00	0.52	0.25	0.20	0.65	0.63	0.50	0.66
Weight	0.52	1.00	0.62	0.50	0.70	0.75	0.67	0.61
Wheelbase	0.25	0.62	1.00	0.79	0.46	0.38	0.29	0.37
Length	0.20	0.50	0.79	1.00	0.44	0.29	0.19	0.32
EngineSize	0.65	0.70	0.46	0.44	1.00	0.74	0.59	0.85
MPG_City	0.63	0.75	0.38	0.29	0.74	1.00	0.88	0.70
MPG_Highway	0.50	0.67	0.29	0.19	0.59	0.88	1.00	0.55
Cylinders	0.66	0.61	0.37	0.32	0.85	0.70	0.55	1.00

```
> plot(vc)
```



Graphical LASSO

$$\hat{\Theta} = \operatorname{argmin}_{\Theta \geq 0} \left(\operatorname{tr}(S\Theta) - \log \det(\Theta) + \lambda \sum_{j \neq k} |\Theta_{jk}| \right)$$

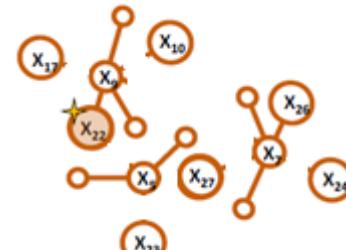
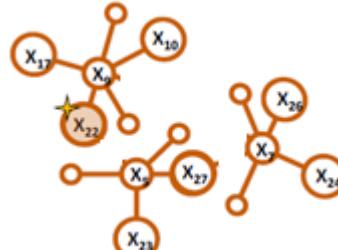
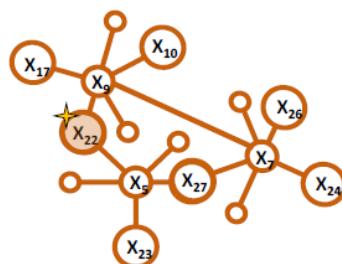
- Обратная ковариационная матрица Θ описывает частичные попарные корреляции переменных, S наблюдаемая ковариационная матрицы, λ - параметр регуляризации, чем больше, тем матрица Θ более разрежена:

$$\begin{pmatrix} \cdot & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\begin{pmatrix} \cdot & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & 0 & \cdot & 0 \\ \cdot & 0 & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & 0 & \cdot & \cdot \end{pmatrix}$$

$$\begin{pmatrix} \cdot & 0 & \cdot & 0 & 0 \\ 0 & \cdot & 0 & 0 & 0 \\ \cdot & 0 & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & \cdot \end{pmatrix}$$

тем меньше дуг в графе связей переменных:



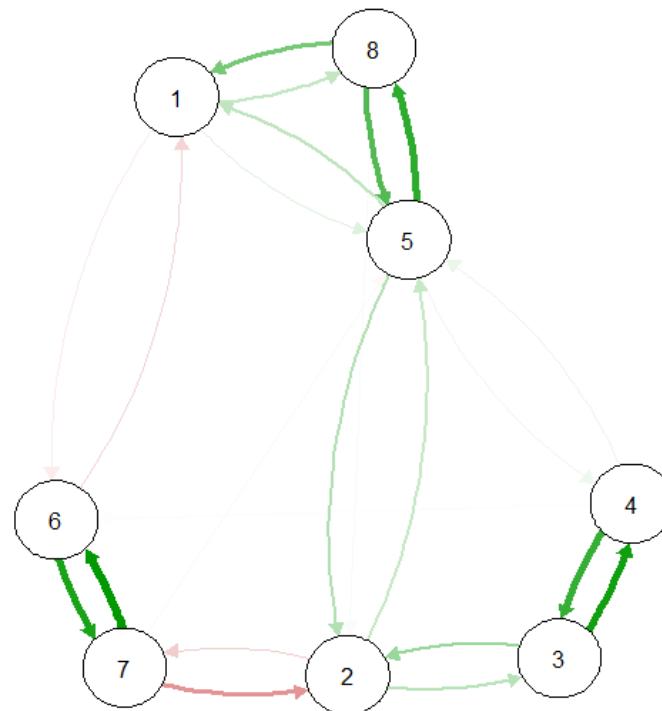
Применение glasso

```
> feats <- c("Horsepower", "Weight", "Wheelbase", "Length",
+           "EngineSize", "MPG_City", "MPG_Highway", "Cylinders")
> s<- var(scale(na.omit(cars[,feats])))
>
> a<-glasso(s, rho=0.1, approx = TRUE)
```

```
> a$wi
      [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.0000000 0.0000000 0.0000000
[2,] 0.0000000 0.0000000 0.2219857 0.0000000
[3,] 0.0000000 0.31756764 0.0000000 0.75602837
[4,] 0.0000000 0.0000000 0.6368529 0.0000000
[5,] 0.2012974 0.24302696 0.0000000 0.05295272
[6,] -0.1345076 0.0000000 0.0000000 0.0000000
[7,] 0.0000000 -0.33662554 0.0000000 0.0000000
[8,] 0.4355049 0.02032323 0.0000000 0.0000000

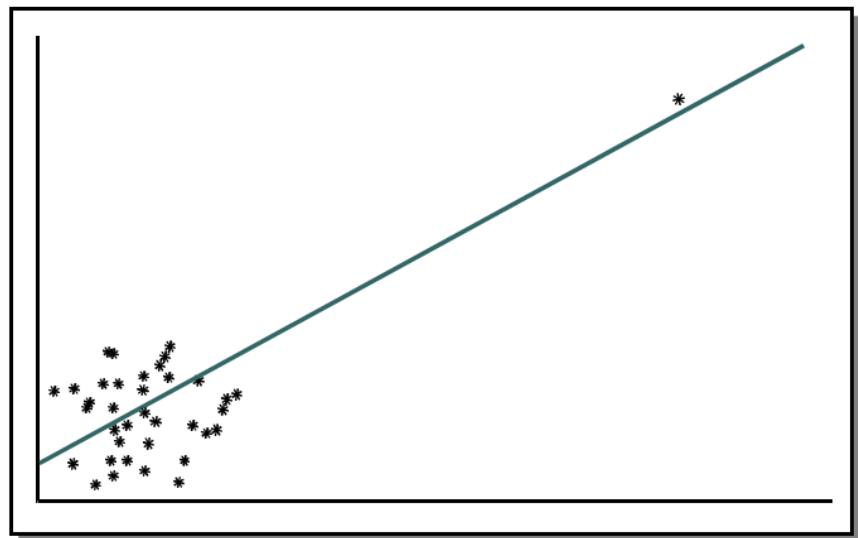
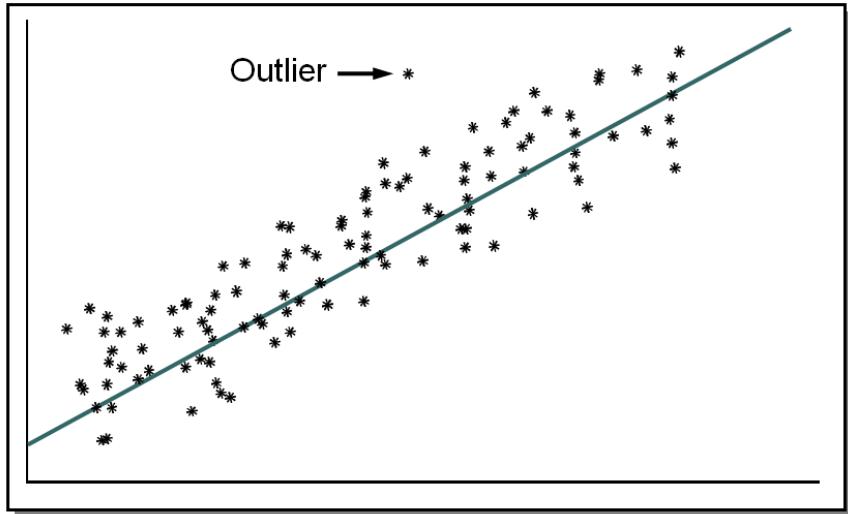
      [,5]      [,6]      [,7]      [,8]
[1,] .10275220 -0.055059945 0.0000000 0.1872723
[2,] .18254796 0.0000000000 -0.1570464 0.0000000
[3,] 0.0000000 0.0000000000 0.0000000 0.0000000
[4,] .06626767 -0.007185026 0.0000000 0.0000000
[5,] 0.0000000 0.0000000000 0.0000000 0.6594487
[6,] 0.0000000 0.0000000000 0.7247132 0.0000000
[7,] .01762976 0.801977897 0.0000000 0.0000000
[8,] .54102963 0.0000000000 0.0000000 0.0000000
```

```
> qgraph(a$wi, nodeNames = feats)
```



- 1: Horsepower
- 2: Weight
- 3: Wheelbase
- 4: Length
- 5: EngineSize
- 6: MPG_City
- 7: MPG_Highway
- 8: Cylinders

Выбросы и влиятельные наблюдения



Статистики для обнаружения выбросов и влиятельных наблюдений

RSTUDENT остатки

$$RSTUDENT = \frac{r_i}{s_{(i)}\sqrt{1 - h_i}}$$

оцениваются остатки после удаления наблюдения из выборки

Leverage

Оценка удаленности наблюдения от основного «облака»

Cook's D

Оценивает общее изменение в параметрах модели после удаления наблюдения.

$$D_i = \frac{\sum_{j=1}^n (\hat{Y}_j - \hat{Y}_{j(i)})^2}{p \text{ MSE}}$$

DFFITS

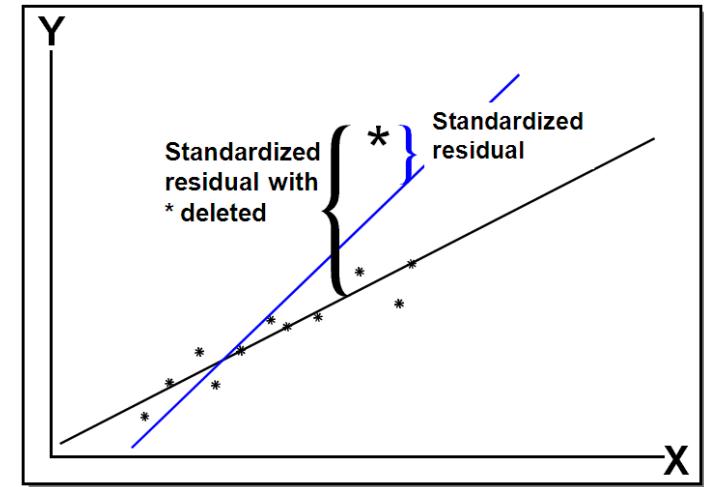
Оценивает изменение в прогнозе после удаления наблюдения.

$$DFFITS = \frac{\bar{y}_i - \bar{y}_{(i)}}{s_{(i)}\sqrt{h(i)}}$$

DFBETAs

оценивает изменение в каждом параметре после удаления наблюдения

$$DFBETA_{j(i)} = \frac{b_j - b_{j(i)}}{\hat{\sigma}(b_j)}$$



COVRATIO

оценивает изменение ковариационной матрицы после удаления наблюдения

$$COVRATIO_i = \frac{|s_{(i)}^2 (X'_{(i)} X_{(i)})^{-1}|}{|s^2 (XX)^{-1}|}$$

Пороговые значения статистик для обнаружения выбросов

Influential Statistics	Cutoff Values
RSTUDENT Residuals	$ RSTUDENT > 2$
LEVERAGE	$LEVERAGE > \frac{2p}{n}$
Cook's D	$CooksD > \frac{4}{n}$
DFFITS	$ DFFITS > 2\sqrt{\frac{p}{n}}$
DFBETAS	$ DFBETAS > \frac{2}{\sqrt{n}}$
COVRATIO	$COVRATIO < 1 - \frac{3p}{n}$ or $COVRATIO > 1 + \frac{3p}{n}$

Что делать?

1. Проверить данные на предмет ошибок и артифактов
2. Проверить адекватность модели
3. Оценить робастность модели, построив ее с участием и без участия выбросов
4. Использовать робастные регрессии и функции потерь

Поиск выбросов и влиятельных наблюдений

```
> lm_model <- lm(Invoice ~ Horsepower + Weight + Wheelbase, cars)
> summary(lm_model)

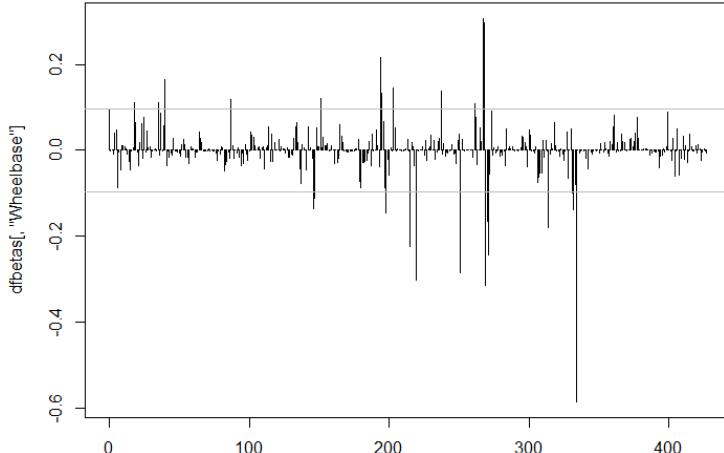
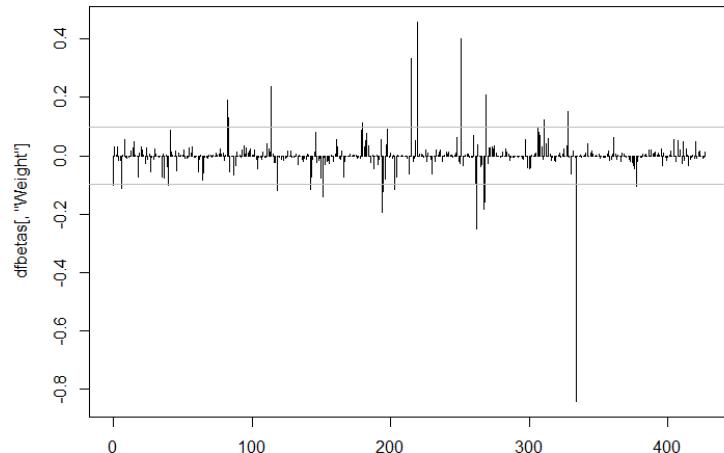
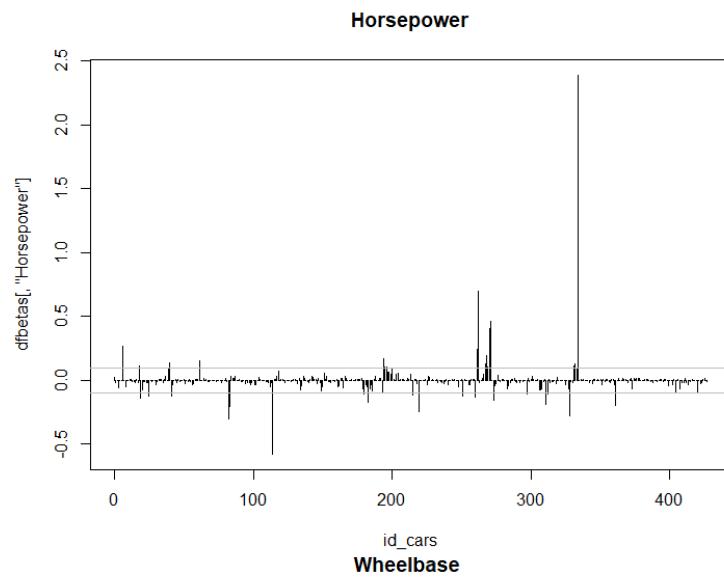
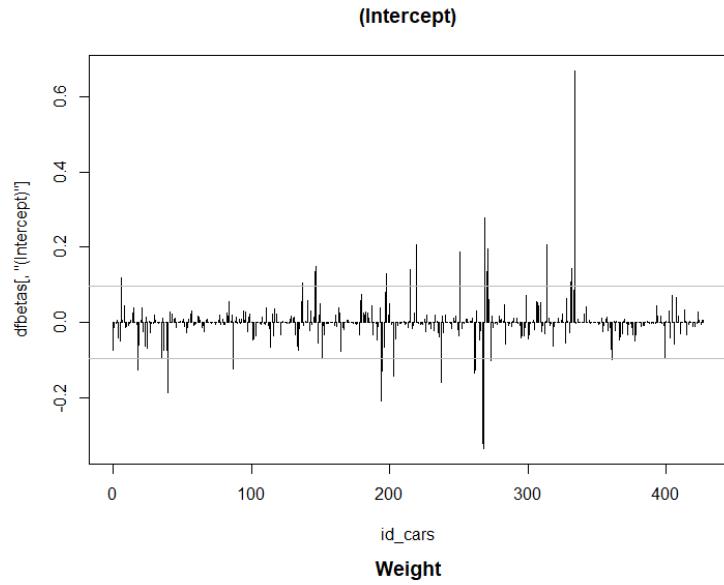
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 35434.018    7106.827   4.986 9.01e-07 ***
Horsepower    212.507      8.343   25.471 < 2e-16 ***
Weight        2.133       1.122    1.902   0.0578 .
Wheelbase     -544.870     86.197   -6.321 6.58e-10 ***
---
```

```
> dfbetas <- as.data.frame(dfbetas(lm_model))
> dfbetas$Residual <- lm_model$residuals
> dfbetas$Rstudent <- rstudent(lm_model)
> dfbetas$HatDiagH <- lm.influence(lm_model)$hat
> dfbetas$DFFITS <- dffits(lm_model)
> dfbetas$CooksD <- cooks.distance(lm_model)
> head(dfbetas)

(Intercept) Horsepower      Weight     Wheelbase Residual Rstudent HatDiagH DFFITS CooksD
1 -0.074568269  0.0200873357 -0.101787142  0.0941997756 -10151.267 -1.0818403 0.012925816 -0.12379903 0.0038300108
2 -0.013419513 -0.0205595682  0.030072459 -0.0005231153  -7069.350 -0.7501156 0.005704294 -0.05681609 0.0008078504
3 -0.003552372 -0.0012243044  0.003984042  0.0004844926  -2968.199 -0.3143266 0.002849039 -0.01680157 0.0000707235
4  0.005051827 -0.0573570954  0.031457134 -0.0091549292 -11293.141 -1.1988749 0.004616168 -0.08164308 0.0016646813
5 -0.041973444 -0.0006748431 -0.014834282  0.0410677930  10148.066  1.0767451 0.004215882  0.07006071 0.0012266646
6 -0.049464237 -0.0019379098 -0.015848925  0.0479789006  12206.331  1.2959017 0.004175827  0.08391739 0.0017577156
```

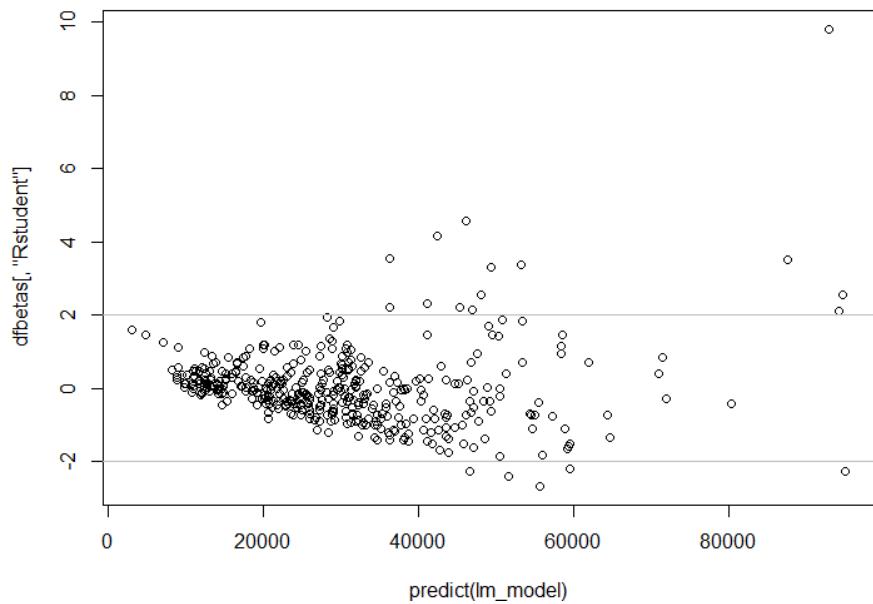
Поиск выбросов и влиятельных наблюдений

```
> plot(id_cars, dfbetas[,"(Intercept)"], type = "h", main = "(Intercept)")  
> abline(h = 2/sqrt(n), col = "gray")  
> abline(h = -2/sqrt(n), col = "gray")
```

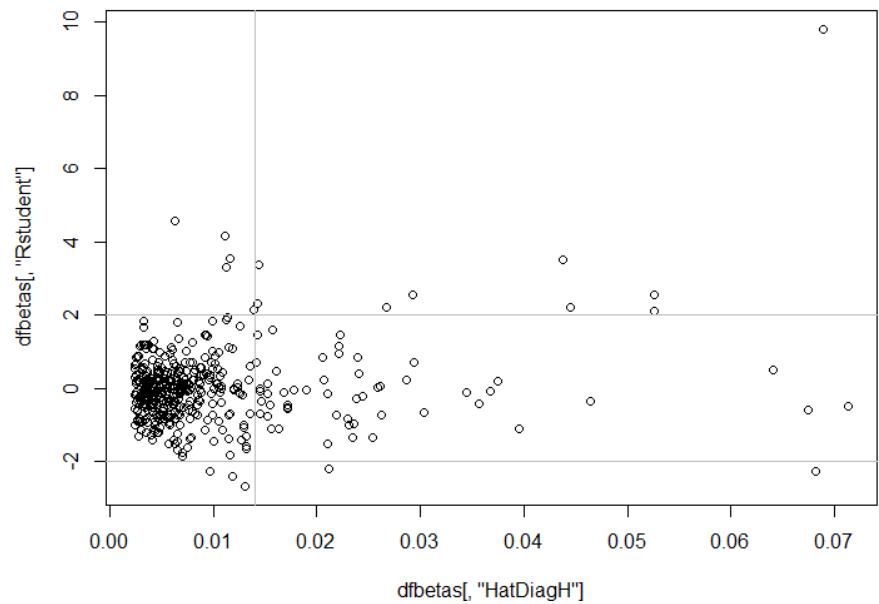


Поиск выбросов и влиятельных наблюдений

```
> plot(predict(lm_model), dfbetas[, "Rstudent"])
> abline(h = 2, col = "gray")
> abline(h = -2, col = "gray")
```

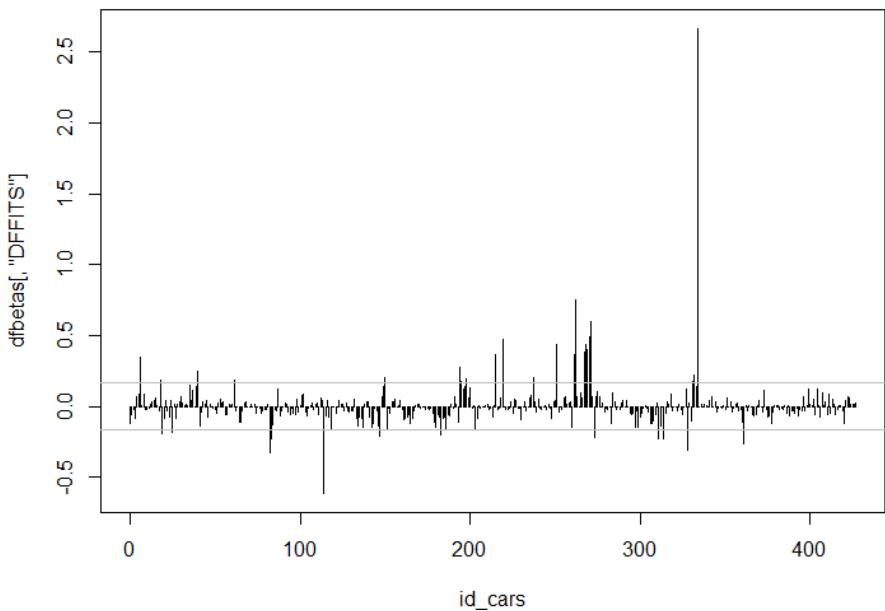


```
> plot(dfbetas[, "HatDiagH"], dfbetas[, "Rstudent"])
> abline(h = 2, col = "gray")
> abline(h = -2, col = "gray")
> abline(v = 2*p/n, col = "gray")
```

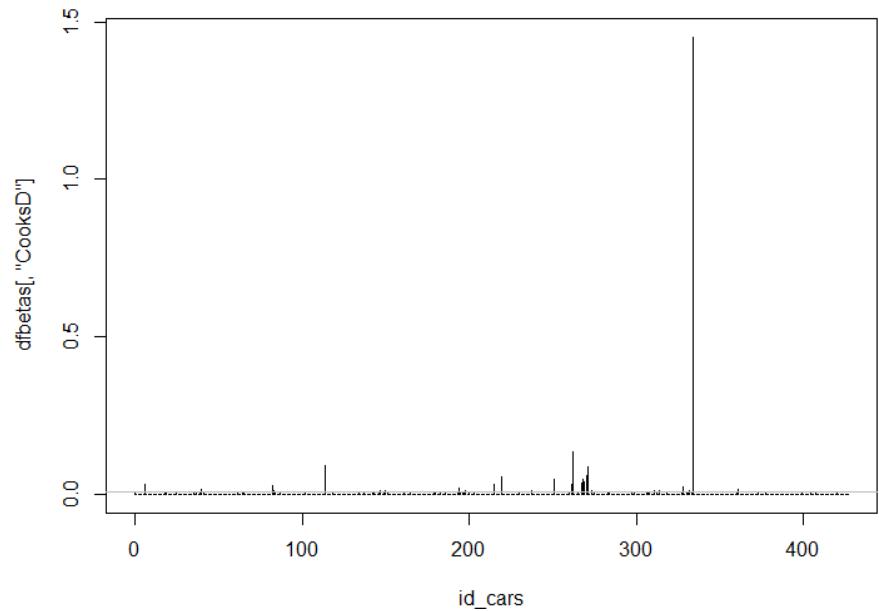


Поиск выбросов и влиятельных наблюдений

```
> plot(id_cars, dfbetas[, "DFFITS"], type = "h")
> abline(h = 2*sqrt(p/n), col = "gray")
> abline(h = -2*sqrt(p/n), col = "gray")
```



```
> plot(id_cars, dfbetas[, "CooksD"], type = "h")
> abline(h = 4/n, col = "gray")
```



Рассматриваемые модели

Предиктор Отклик	Категориальный	Непрерывный	Непрерывный и категориальный
Непрерывный	Дисперсионный анализ (ANOVA)	Регрессия наименьших квадратов (OLS Regression)	Ковариационный анализ (ANCOVA)
Категориальный	Логистическая регрессия и таблицы частот	Логистическая регрессия	Логистическая регрессия

ANCOVA = ANOVA+Регрессия

- Комбинация регрессии и дисперсионной модели
- Сравнивает групповые средние с учетом «ковариатов» - числовых предикторов, влияющих на отклик

GROUP 1		GROUP 2		...	GROUP k	
X	y	x	y	...	x	y
X_{11}	y_{11}	X_{21}	y_{21}	...	X_{k1}	y_{k1}
X_{12}	y_{12}	X_{22}	y_{22}	...	X_{k2}	y_{k2}
X_{13}	y_{13}	X_{23}	y_{23}	...	X_{k3}	y_{k3}
...
X_{1n_1}	y_{1n_1}	X_{2n_2}	y_{2n_2}	...	X_{kn_k}	y_{kn_k}

- Модель среднего для каждой группы: $\mu_i = a_i + \beta x$
- Групповые смещения разные, а регрессионные коэффициенты одинаковые:

$$\hat{y} = a_i + b x$$

$$b = \frac{\sum_i S_{xy}(i)}{\sum_i S_{xx}(i)}$$

$$a_i = \bar{y}_i - b \bar{x}$$

ANCOVA – анализ вариации

- Квадраты остатков: вся вариация (S_{YY}) = неописанная (SSE)+групповая (SSG)+ «регрессионная» (SSX)

$$TOT(SS) = S_{yy}$$

$$SSE = \frac{\left(\sum_i S_{xx(i)} \right) \cdot \left(\sum_i S_{yy(i)} \right) - \left(\sum_i S_{xy(i)} \right)^2}{\left(\sum_i S_{xx(i)} \right)}$$
$$SSG = \frac{S_{xx} \cdot S_{yy} - S_{xy}^2}{S_{xx}} - SSE$$
$$SSX = \sum_i S_{yy(i)} - SSE$$

- Критерий Фишера:

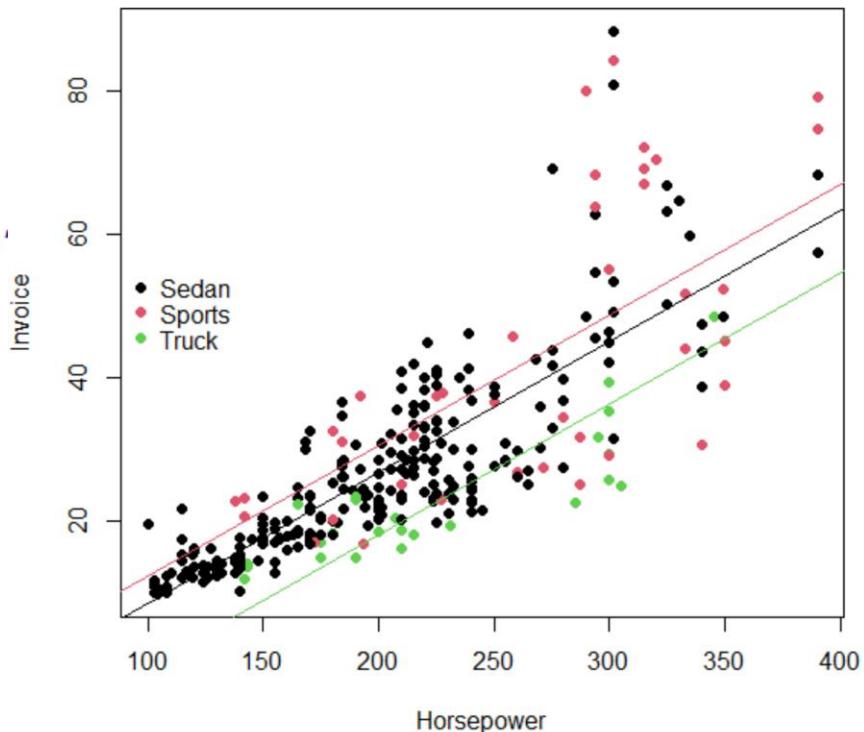
SOURCE	df	SS	MS	F
GROUP	k-1	SSG	MSG	$F_G = MSG/MSE$
X (covariate)	1	SSX	MSX	$F_X = MSX/MSE$
Error	N-k-1	SSE	MSE	
Total	N-1	TOT(SS)		

- Базовые гипотезы две:

- Все групповые средние равны
- Коэффициенты регрессии равен 0

Пример

```
> aov_model<-aov(Invoice~Type+Horsepower-1,df)
> summary(aov_model)
      Df Sum Sq Mean Sq F value Pr(>F)
Type        3 267638   89213 1287.1 <2e-16 ***
Horsepower   1 38502    38502  555.5 <2e-16 ***
Residuals  316 21904       69
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 
> aov_model$coefficients
  TypeSedan  TypeSports  TypeTruck  Horsepower
-9.6198161 -5.9532235 -18.3992439   0.1824284
```

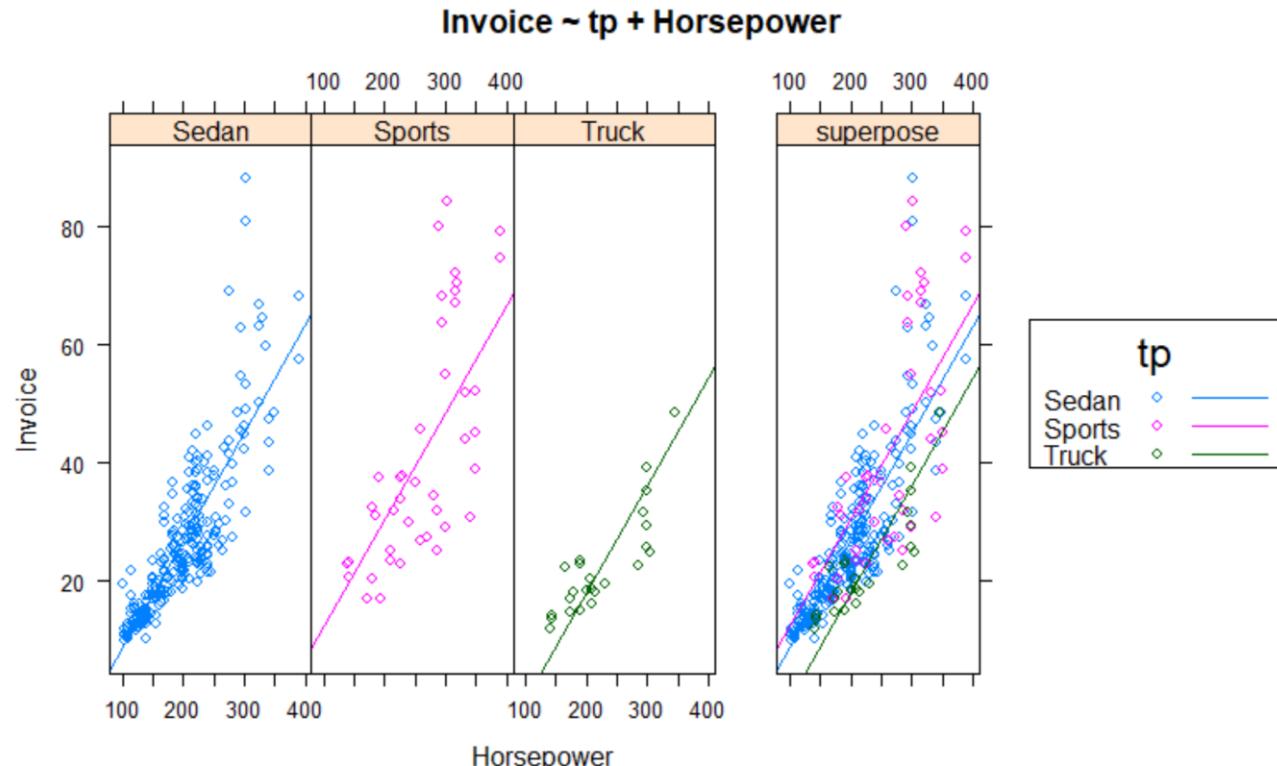


```
> cols=factor(df>Type)
> with(df,plot(x=Horsepower,y=Invoice,col=cols,pch=19))
> legend("left",legend=levels(cols),pch=19, col=factor(levels(cols)),box.lty=0)
> for (i in c(1,2,3))
+ abline(a=aov_model$coefficients[i],b=aov_model$coefficients[4],col=factor(levels(cols))[i])
```

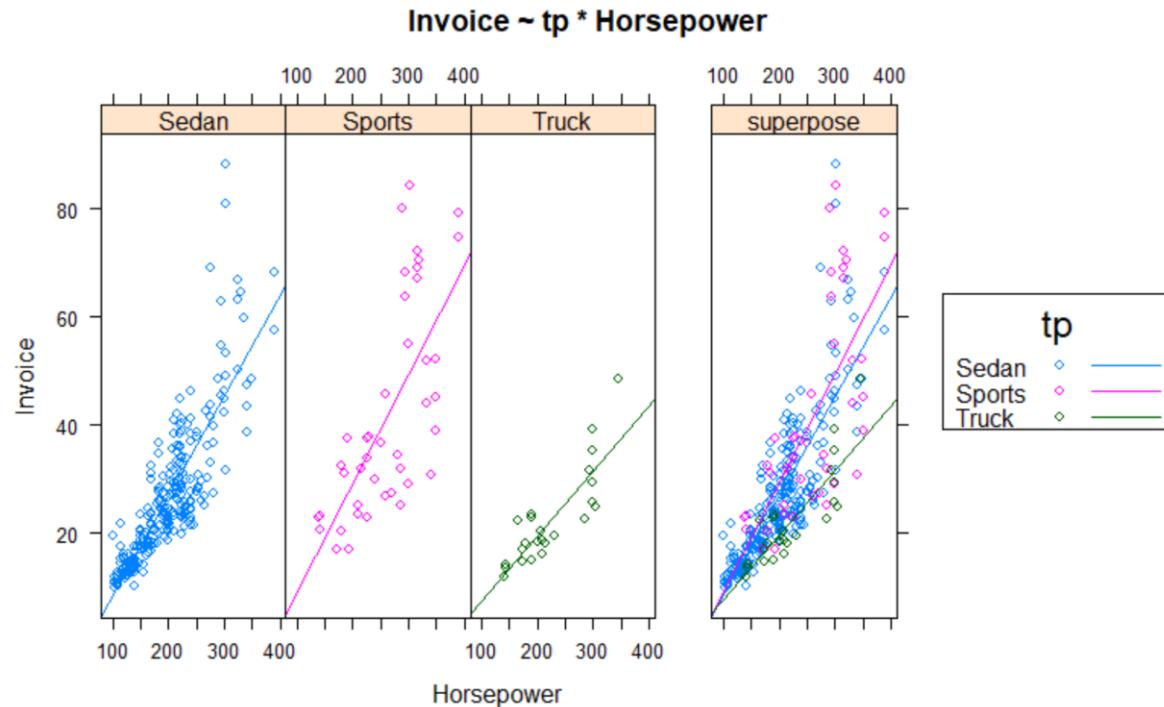
Приимер на ancova (пакет HH)

```
> df$tp<-factor(df>Type)
> ancova(Invoice~tp+Horsepower,df)
Analysis of Variance Table

Response: Invoice
          Df Sum Sq Mean Sq F value    Pr(>F)
tp          2   9507   4753  68.575 < 2.2e-16 ***
Horsepower  1  38502   38502 555.457 < 2.2e-16 ***
Residuals  316  21904      69
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```



Взаимодействующие предикторы ANCOVA



```
> ancova(Invoice~tp*Horsepower,df)
Analysis of Variance Table

Response: Invoice
          Df Sum Sq Mean Sq F value    Pr(>F)
tp          2   9507   4753  69.453 < 2e-16 ***
Horsepower  1  38502  38502 562.569 < 2e-16 ***
tp:Horsepower 2    414    207   3.023  0.05007 .
```

Анализ вариации квадратов остатков (SS) типа I, II, III

А как считается SS?

Тип I (последовательный): SS(A), SS(B|A), SS(AB|A,B) Порядок важен!

```
> anova(aov_model)
Analysis of Variance Table
```

```
Response: Invoice
          Df Sum Sq Mean Sq F value Pr(>F)
Type          3 267638   89213 1303.535 < 2e-16
Horsepower    1  38502   38502  562.569 < 2e-16
Type:Horsepower 2    414     207    3.023 0.05007 .
Residuals    314  21490      68
```

```
> anova(aov(Invoice~Horsepower*Type-1,df))
Analysis of Variance Table
```

```
Response: Invoice
          Df Sum Sq Mean Sq F value Pr(>F)
Horsepower      1 300669   300669 4393.237 < 2.2e-16 ***
Type            3    5471     1824   26.646  2.22e-15 ***
Horsepower:Type 2    414     207    3.023 0.05007 .
Residuals      314  21490      68
```

Тип II (без взаимодействий):
SS(A|B), SS(B|A)

```
> Anova(aov_model,type='II')
Anova Table (Type II tests)
```

```
Response: Invoice
          Sum Sq Df F value Pr(>F)
Type          5471  3  26.646 2.22e-15 ***
Horsepower    38502  1  562.569 < 2.2e-16 ***
Type:Horsepower 414  2   3.023  0.05007 .
Residuals    21490 314
```

Тип III (каждый против остальных):
SS(B|A,AB), SS(A|B,AB)

```
> Anova(aov_model,type='III')
Anova Table (Type III tests)
```

```
Response: Invoice
          Sum Sq Df F value Pr(>F)
Type          2398.3  3  11.681 2.819e-07 ***
Horsepower    29927.0  1  437.280 < 2.2e-16 ***
Type:Horsepower 413.8  2   3.023  0.05007 .
Residuals    21489.8 314
```