Линейные модели (часть 2)

Артем Артемов

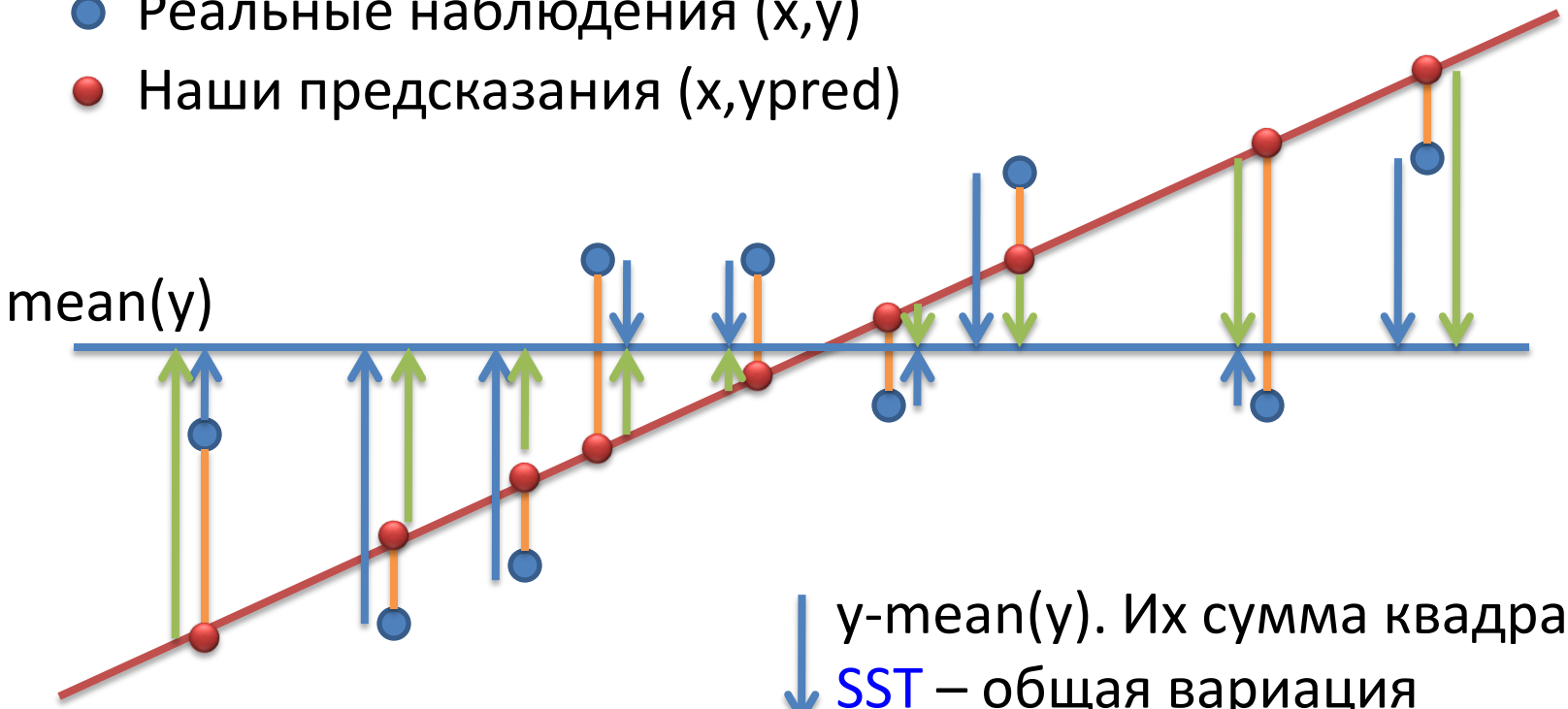
Какая из моделей лучше?

- Можно придумать разные модели, одна учитывает объем памяти, другая – ещё объем жесткого диска, третья дополнительно учитывает, является ли диск диском или твердотельным накопителем (SSD).
- Как сравнить, какая лучше?
- Наивный подход: насколько хорошо модель описывает данные ≈ насколько мала необъясненная дисперсия в y ≈ насколько R^2 близок к 1. Не работает, т.к. добавление параметров увеличивает R^2
- Скорректированный R^2 (*adjusted R^2*), информационные критерии (*AIC*, *BIC*)

http://ru.wikipedia.org/wiki/Коэффициент_детерминации

SST = SSX + SSE (повторение)

- Реальные наблюдения (x,y)
- Наши предсказания (x,ypred)



y-mean(y). Их сумма квадратов
SST – общая вариация

y-ypred. Ошибки.
SSE – необъясненная вариация

ypred-mean(y). Ошибки.
SSX – объясненная вариация

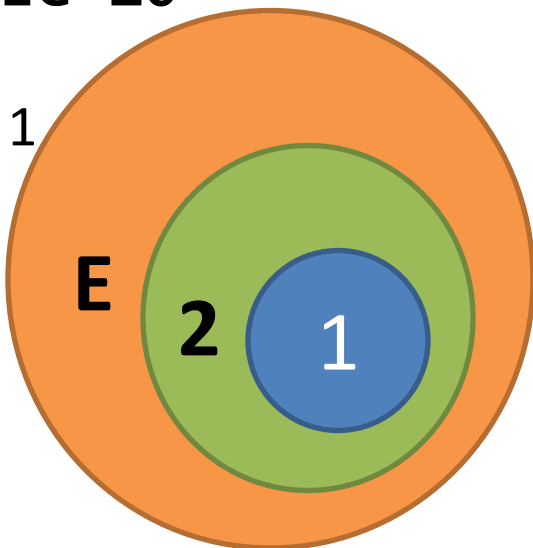
- R^2=SSX/SST
- F=((SSX)/(nparam-1))/
- /(SSE/(npoints-nparam))

ANOVA для сравнения моделей

```
> fit2=lm(Price_RUR ~ Memory_Gb+HDD_Gb+HDD_type, data=laptop)
> fit1=lm(Price_RUR ~ Memory_Gb, data=laptop)
> anova(fit1, fit2)
Analysis of Variance Table
Model 1: Price_RUR ~ Memory_Gb + HDD_Gb + HDD_type
Model 2: Price_RUR ~ Memory_Gb
  Res.Df      RSS Df Sum of Sq   F Pr(>F)
1    302 2.56e+10
2    304 4.61e+10 -2 -2.05e+10 121 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

$$F = \frac{\left(\frac{\mathrm{RSS}_1 - \mathrm{RSS}_2}{p_2 - p_1}\right)}{\left(\frac{\mathrm{RSS}_2}{n - p_2}\right)}$$

- F=((SSX**2**-SSX**1**)/(nparam**2**-nparam**1**))/
  /(SSE2/(npoints-nparam**2**))



http://www.statmethods.net/stats/regression.html
http://en.wikipedia.org/wiki/F_test

ANOVA для сравнения моделей

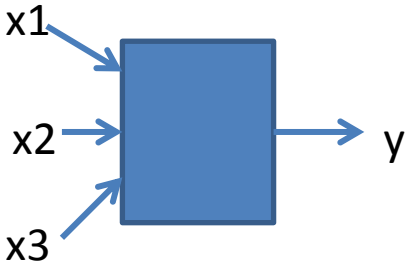
- Противоположный пример

```
> fit1=lm(Price_RUR ~ Memory_Gb+HDD_Gb+HDD_type+Color,
    data=laptop)
> fit2=lm(Price_RUR ~ Memory_Gb+HDD_Gb+HDD_type, data=laptop)
> anova(fit1, fit2)
Analysis of Variance Table

Model 1: Price_RUR ~ Memory_Gb + HDD_Gb + HDD_type + Color
Model 2: Price_RUR ~ Memory_Gb + HDD_Gb + HDD_type
Res.Df      RSS  Df Sum of Sq    F Pr(>F)
1    288 2.43e+10
2    302 2.56e+10 -14 -1.27e+09 1.08   0.38
```

Для чего нужны линейные модели?

Входные данные

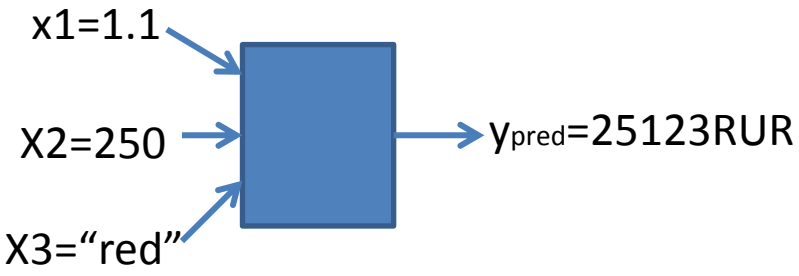
y	x1	x2	x3



Значимость каждой переменной:

Предсказание *y* по *x*

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   9503.178   1733.057   5.483 9.10e-08 ***
Memory_Gb     6232.204    421.098  14.800  < 2e-16 ***
HDD_Gb         -26.604      3.275  -8.123 1.34e-14 ***
Colorblue    -2496.492   4012.744  -0.622   0.5343
…
Colorred      1685.698   2736.167   0.616   0.5383
Colorsilver   8617.956   1679.963   5.130 5.33e-07 ***
…
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10330 on 289 degrees of freedom
Multiple R-squared: 0.578,      Adjusted R-squared: 0.5547
F-statistic: 24.74 on 16 and 289 DF,  p-value: < 2.2e-16
```



predict

```
> L3 = lm(formula = Price_RUR ~ HDD_Gb +
  HDD_type + HDD_Gb:HDD_type, data = laptop)
> newlaptops=data.frame( HDD_Gb=c(200, 1000,
  500), HDD_type=c("SSD", "HDD", "HDD"))
> newlaptops
  HDD_Gb HDD_type
1    200      SSD
2   1000      HDD
3    500      HDD

> predict( L3, newlaptops )
       1        2        3
50290.44 29700.06 21065.29
```

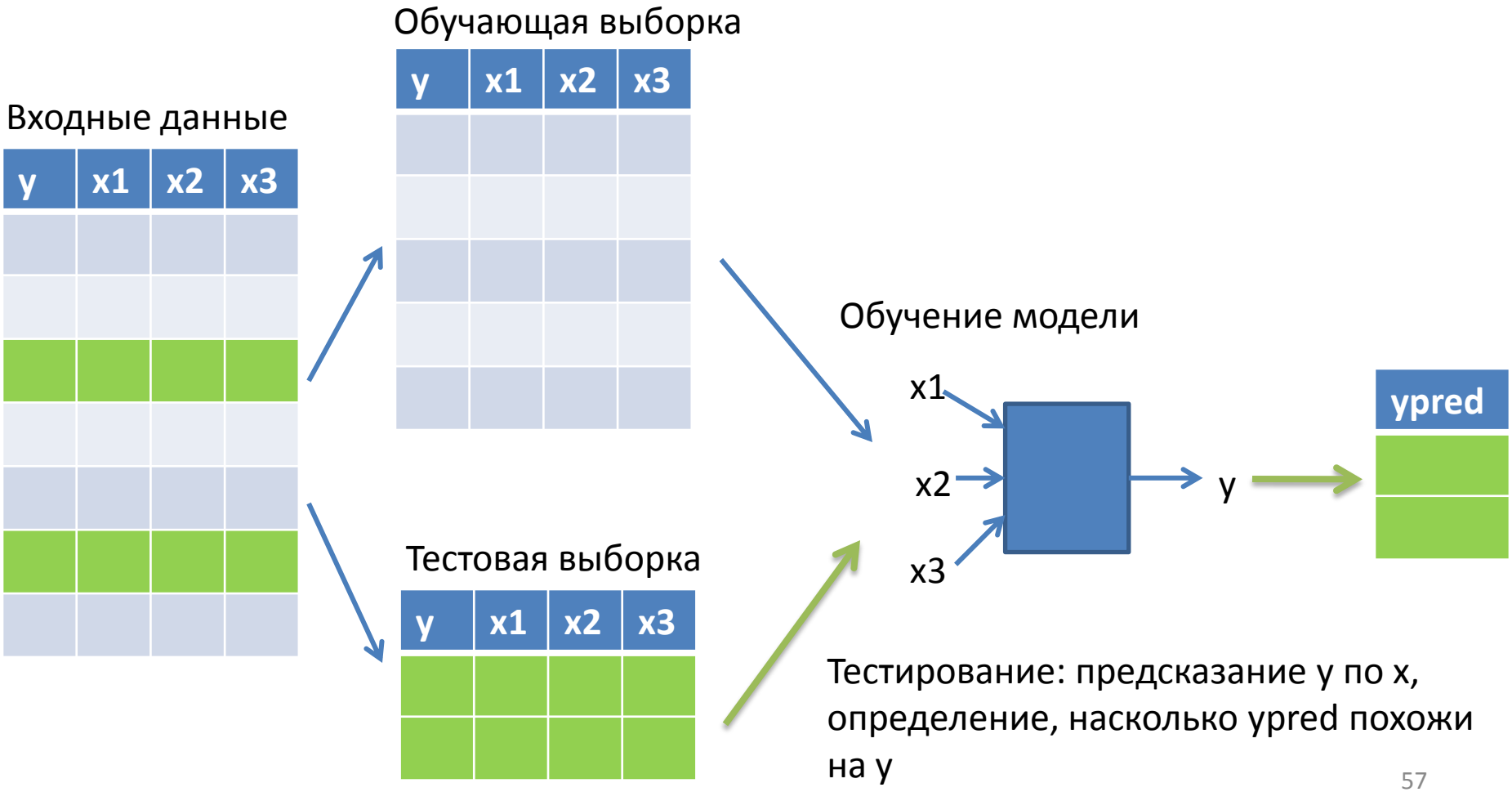
Модель

Dataframe с x-координатами новых точек, для которых делается предсказание y. Названия колонок должны соответствовать предикторам модели

Вектор предсказанных значений y

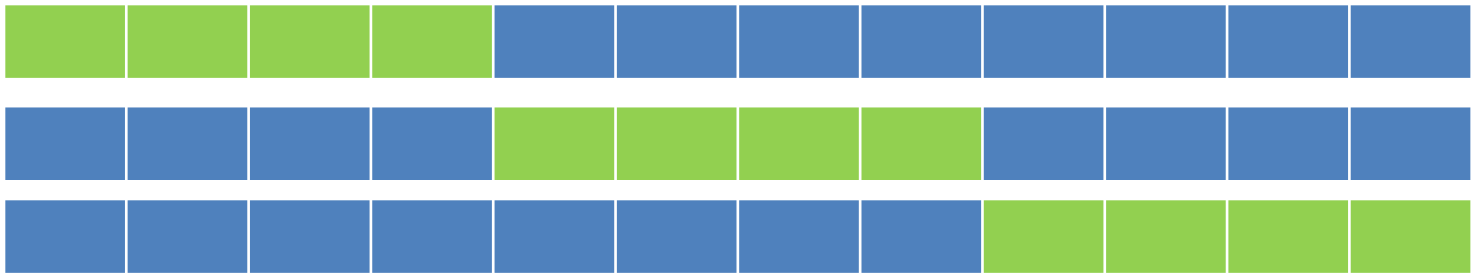
Кросс-валидация

- Для обучения модели и для её тестирования используются разные образцы (=строки в таблице).

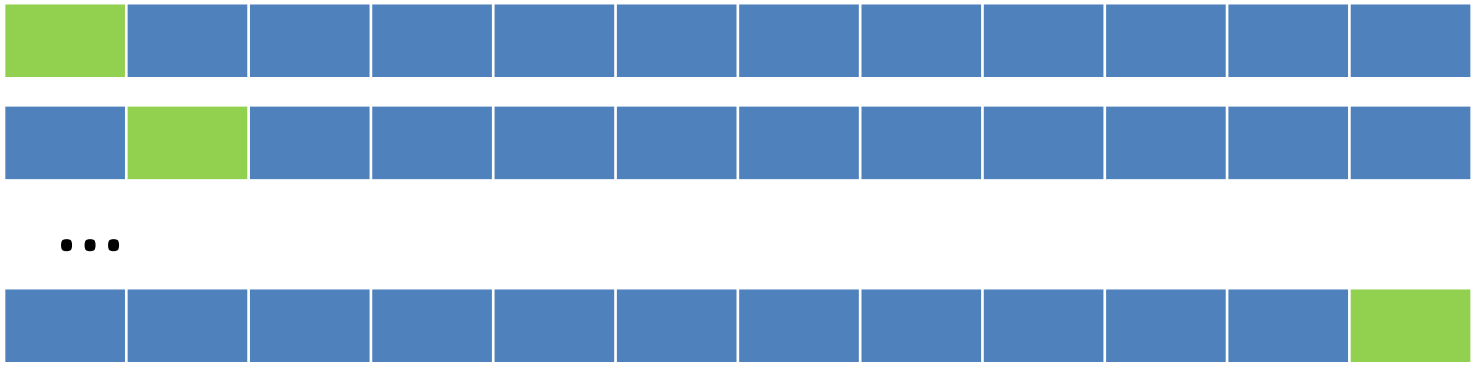


Методы кросс-валидации

- K-fold



- leave one out



Кросс-валидация. Пример

#(!) Устанавка дополнительного пакета

> install.packages("DAAG")

#Его подключение

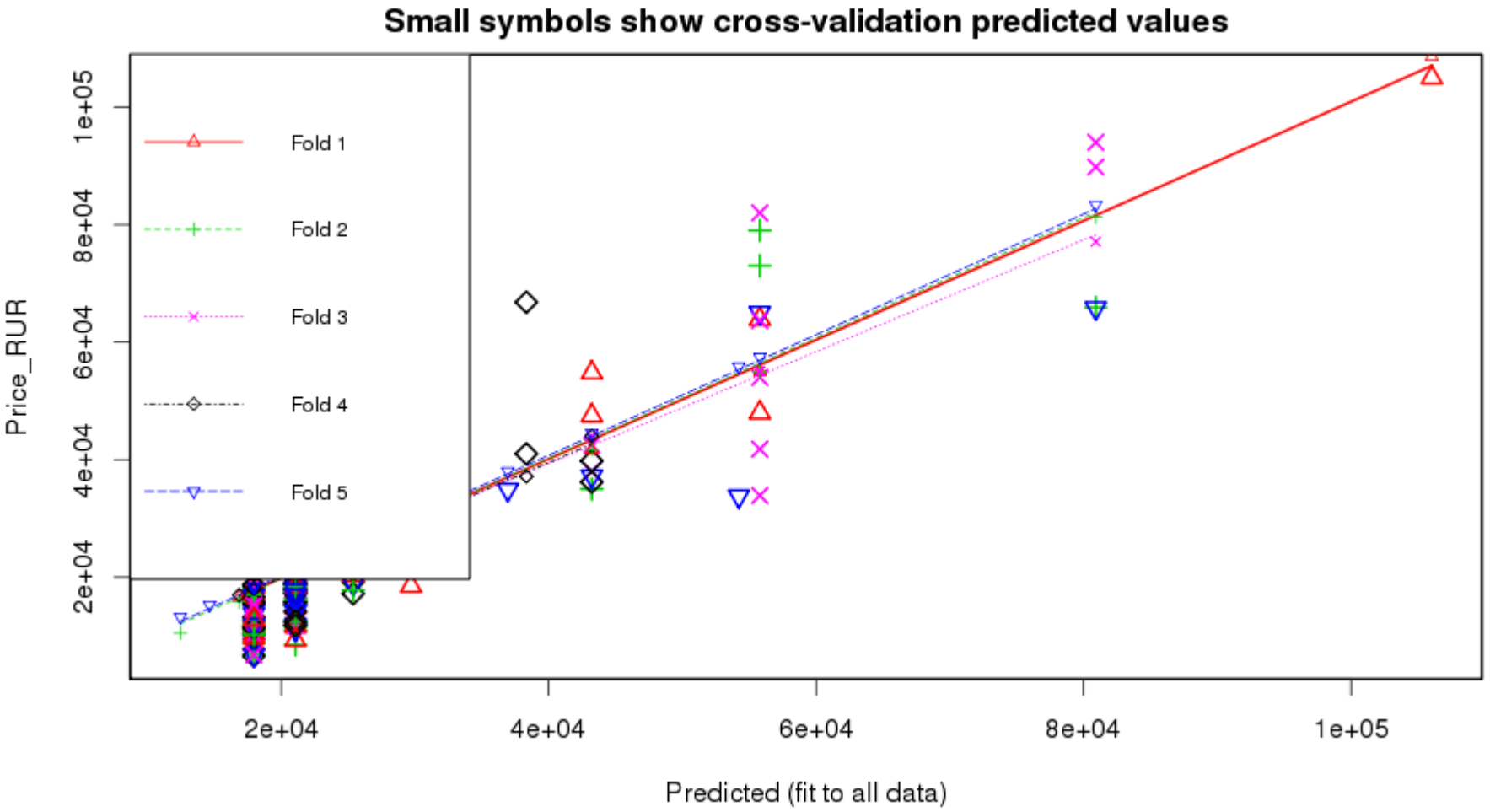
> library('DAAG‘) или > library(DAAG)

> cv.lm(laptop, L3, m=5)

Дополнительный аргумент – функция cost, по умолчанию:

cost=function(y, ypred) { mean ((y-ypred)^2) }

Кросс-валидация. Пример

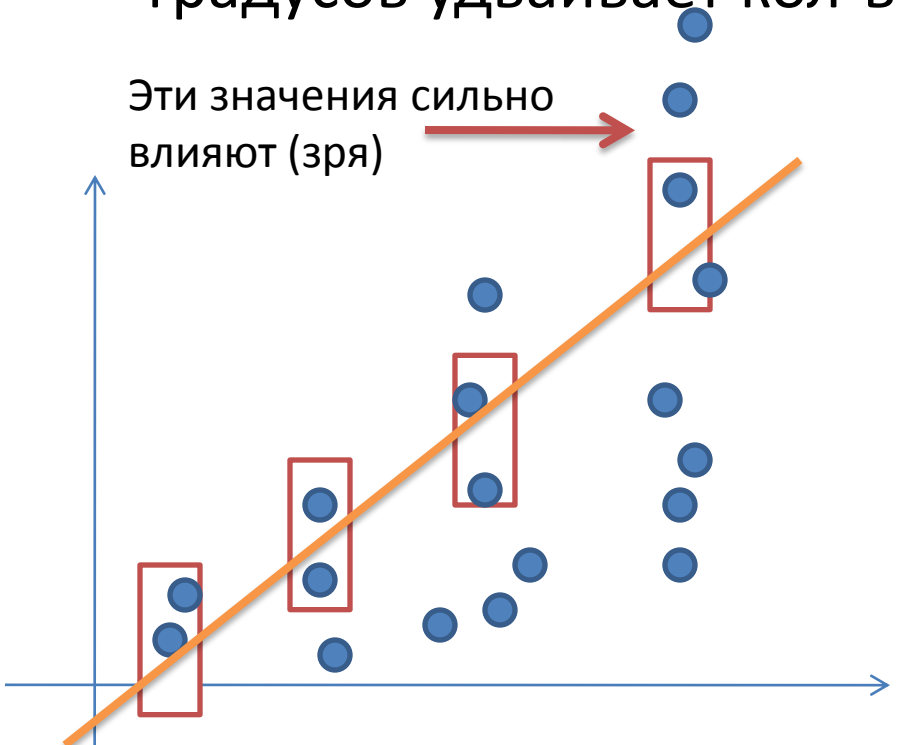


glm – обобщенные линейные модели

- Мотивация: иногда линейные модели не только не точны, но и по смыслу не подходят.
- Пример 1: как зависит количество людей на пляже от температуры
  - $f(-20)=-100$ человек?
- Пример 2: как зависит решение одного человека идти на пляж от температуры
  - p in [0,1]

glm: link function; var(mean)

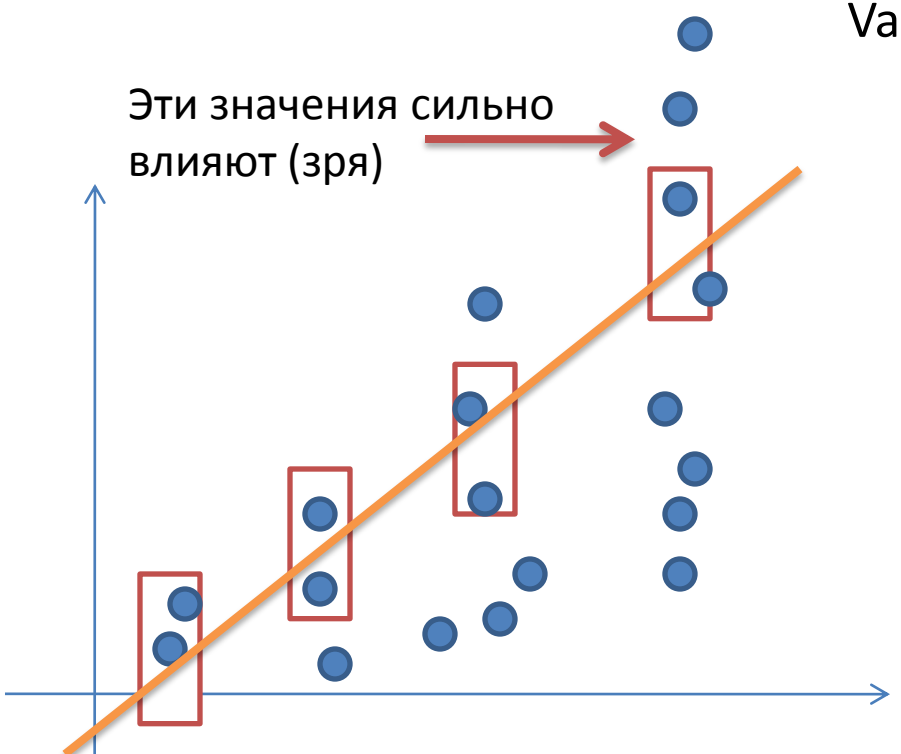
- 2 проблемы:
  - область определения y не соответствует области определения взвешенной суммы Xi
  - в разных областях y имеет разную дисперсию
- Например, пусть увеличение температуры на 5 градусов удваивает кол-во людей на пляже

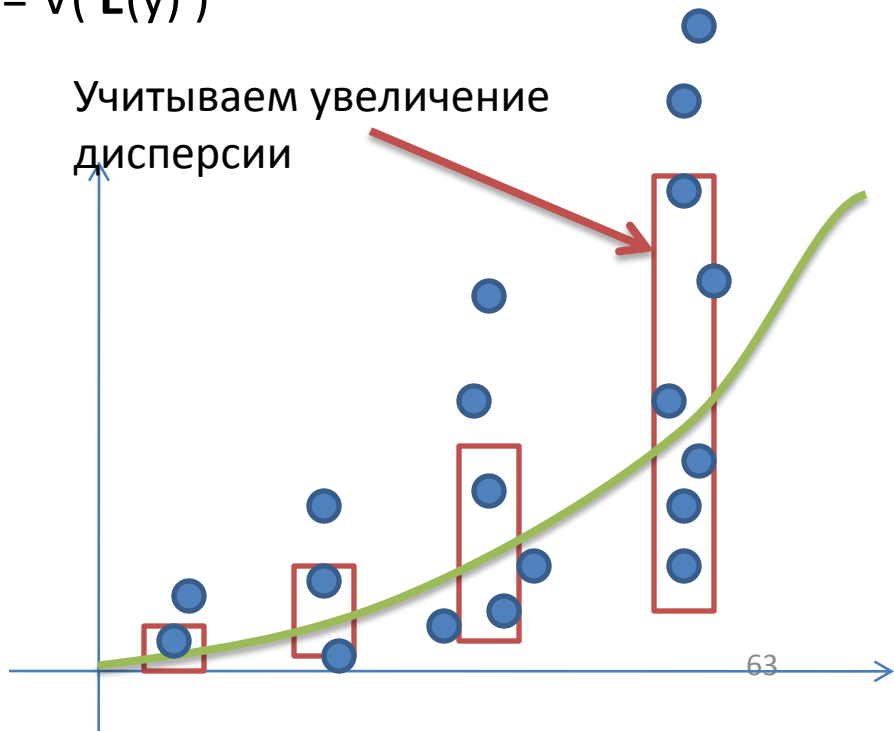


glm: link function; var(mean)

- Решения:
  - область определения ***y*** не соответствует области определения взвешенной суммы ***Xi***
    - link function g: $E(y) = g(\alpha + \beta x)$
  - в разных областях y имеет разную дисперсию
    - Предполагаем некоторую зависимость дисперсии от среднего

$$Var(y) = V(\mathbf{E}(y))$$



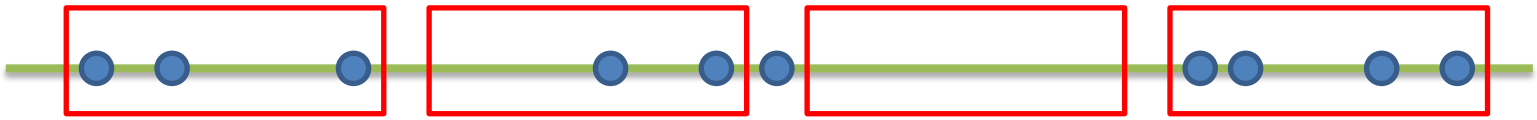


Правдоподобие

- Как учесть при определении параметров и при оценке качества модели разную дисперсию Var(*y*) = V(**E**(y))?
- Вместо суммы квадратов – правдоподобие: насколько при данном распределении и данных наблюдениях вероятно наблюдать такие параметры. Чем больше дисперсия, тем более правдоподобно удаление от среднего.
  - Максимизируем правдоподобие (max likelihood)
  - Упражнение: показать, что при нормальном распределении максимизация правдоподобия идентична минимизации суммы квадратов ошибок [на доске]
- log likelihood ratio = log ($L(M_1) / L(M_0)$) распределено как χ^2

Распределение Пуассона

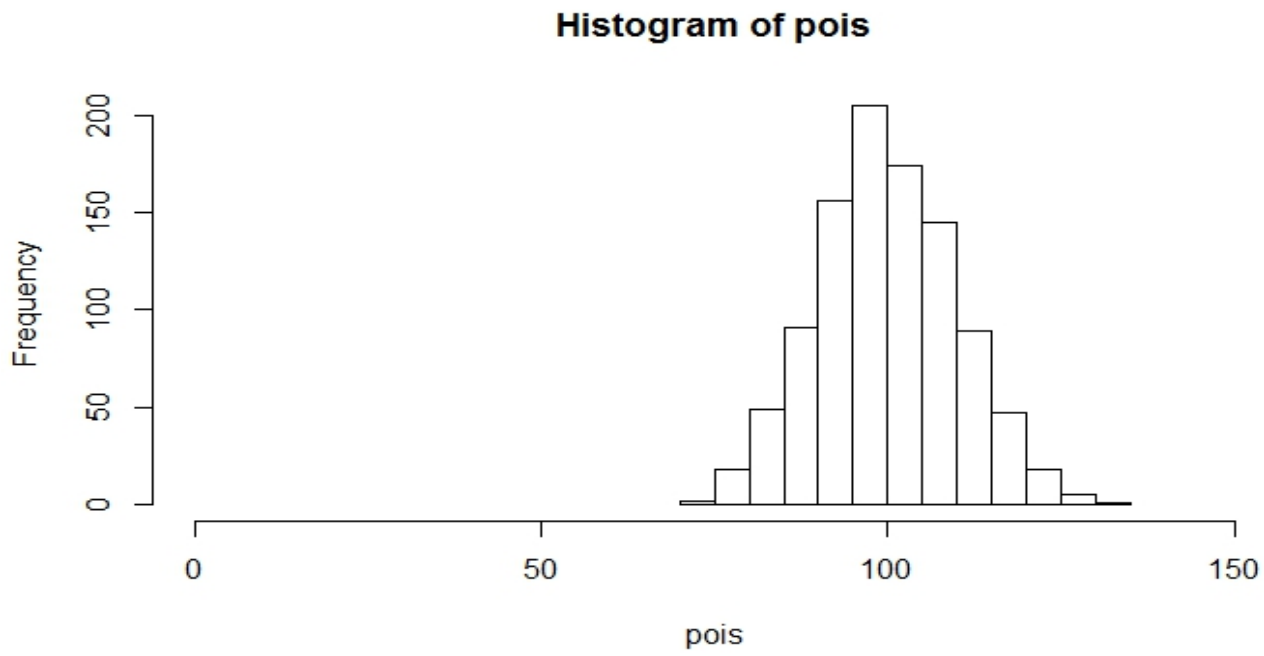
- Распределение количества редких событий в единицу времени (расстояния, объема) при ожидаемой интенсивности λ
  - сколько автобусов проехало мимо за единицу времени, если вы ожидаете увидеть λ автобусов
  - сколько человек проголосовало за единицу времени
  - сколько изюминок в булочке в единице объема



В среднем, в интервал попадает 3 точки, но могут быть и 2, и 0, и 4

Распределение Пуассона

```
> pois<-rpois(1000, lambda=100)
> hist(pois, xlim=c(0, 150))
```



glm: Регрессия Пуассона

- Используется для работы с **количественными данными**
- Предполагается, что зависимая переменная имеет распределение Пуассона (редкие события, например, появление автобусов на остановке за определенный промежуток времени, количество звонков на коммутатор за день и т.п.). События независимы, но происходят с некоторой фиксированной средней интенсивностью
- тогда логарифм ожидаемого значения зависимой переменной (например, количество автобусов) является линейной комбинацией независимых переменных (например, времени)

$$\log(E(Y \mid x)) = \alpha + \beta x$$

Пример: растет ли посещаемость сайта со временем

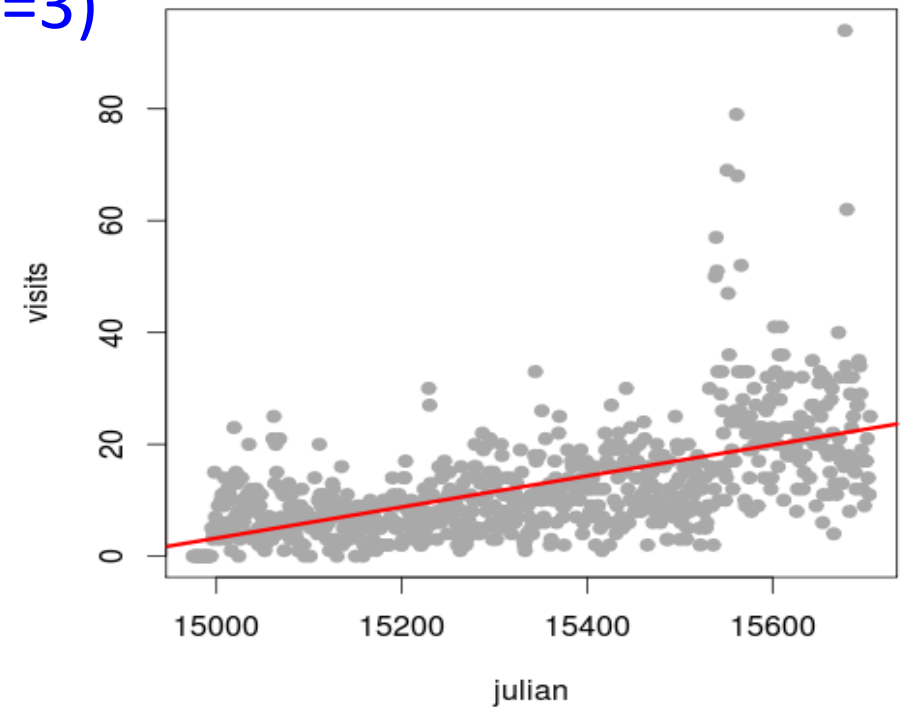
Данные: количество посещений сайта (для удобства превратим дату в единое число)

```
> load('gaData.rda')
> gaData$julian<-julian(gaData$date)
> head(gaData)
        date visits simplystats julian
1 2011-01-01      0           0  14975
2 2011-01-02      0           0  14976
3 2011-01-03      0           0  14977
4 2011-01-04      0           0  14978
5 2011-01-05      0           0  14979
6 2011-01-06      0           0  14980
```

Пример: посещаемость сайта

```
> plot(gaData$julian, gaData$visits, xlab="julian",
    ylab="visits", pch=19, col="darkgrey")
> lm1<-lm(gaData$visits~gaData$julian)
> abline(lm1, col='red', lwd=3)
```

Сначала построим линейную модель.
Не учитывает не-нормальное распределение y.
«Выбросы» сильно сдвигают прямую



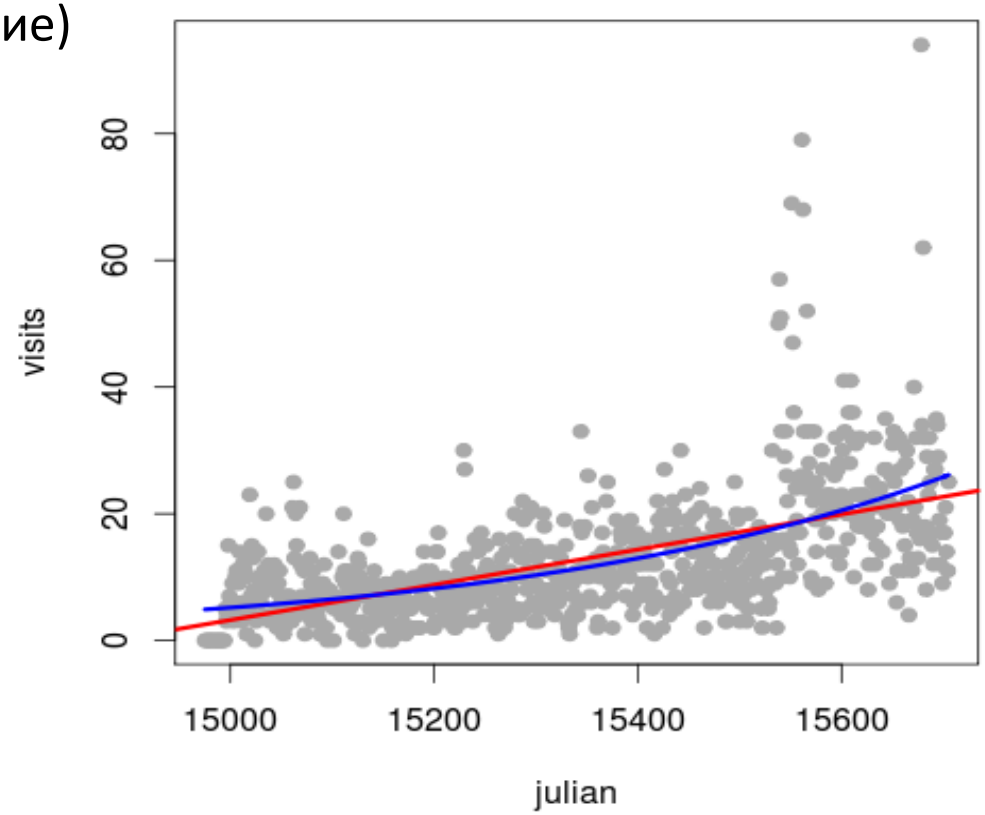
Регрессия Пуассона

```
> glm1<-glm(gaData$visits~gaData$julian, family='poisson')
> lines(gaData$julian, glm1$fitted, col='blue', lwd=3)
```

Наше предсказание (ожидание)

Можно представить. что наши наблюдения – результат генерации из распределения пуассона со средним

$\lambda = lg(a+bx)$



Логистическая регрессия

- Зависимая переменная принимает два значения (болен-здоров, жив-мертв, …)

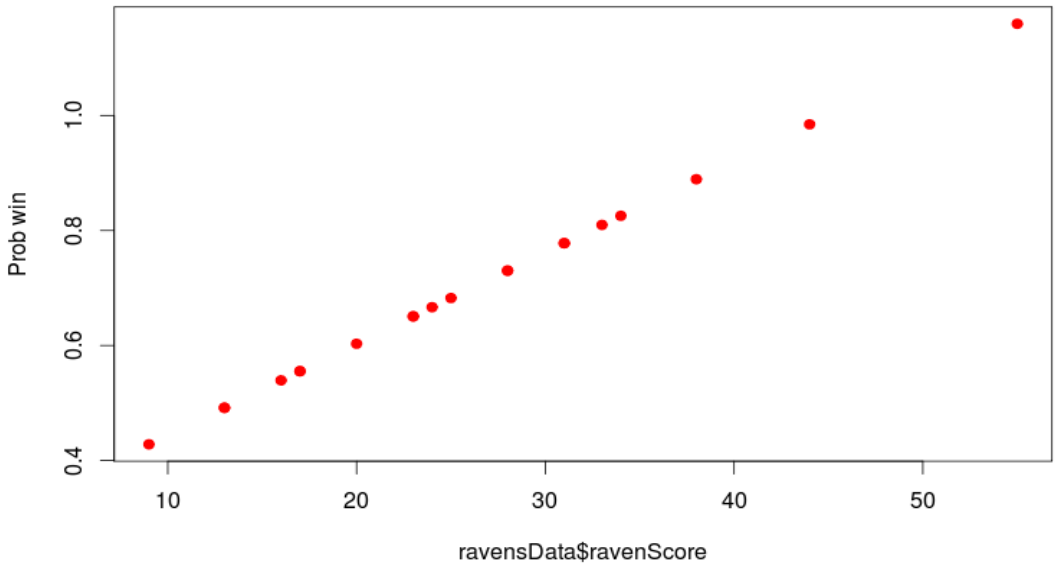
  Пример: таблица выигрышей команды. Хотим вычислять **вероятность** выигрыша в зависимости от количества очков

```
> load("ravensData.rda")
> head(ravensData)
  ravenWinNum ravenWin ravenScore opponentScore
1           1        W         24             9
2           1        W         38            35
3           1        W         28            13
4           1        W         34            31
5           1        W         44            13
6           0        L         23            24
```

Линейная регрессия не подходи

```
> lmRav<-
    lm(ravensData$ravenWinNum~ravensData$ravenScore)
> plot(ravensData$ravenScore, lmRav$fitted, pch=19, col='red',
    ylab="Prob win")
```

Для некоторых значений «предсказание» больше 1



Логистическая регрессия. link function

- Цель: превретить взвешенную сумму предикторов (любое число) в число из [0,1]

$$\log\left(\frac{Pr(RW_i|RS_i, b_0, b_1)}{1-Pr(RW_i|RS_i, b_{0,} b_1)}\right)=b_0+b_1 RS_i$$

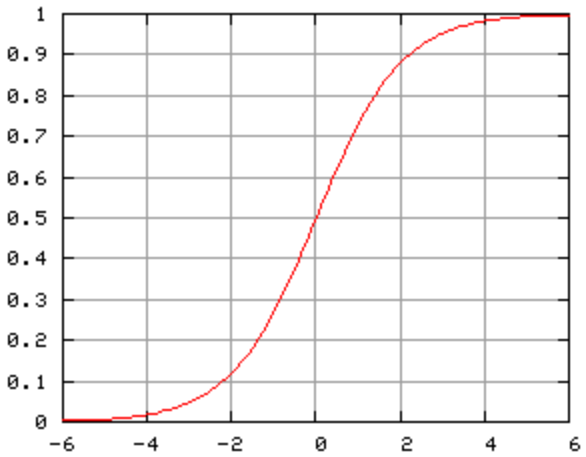
где RW_i- выигрыш (0 или 1)

RS_i- количесво очков

link function g:

g(p)=log(p / (1-p))

График g^-1

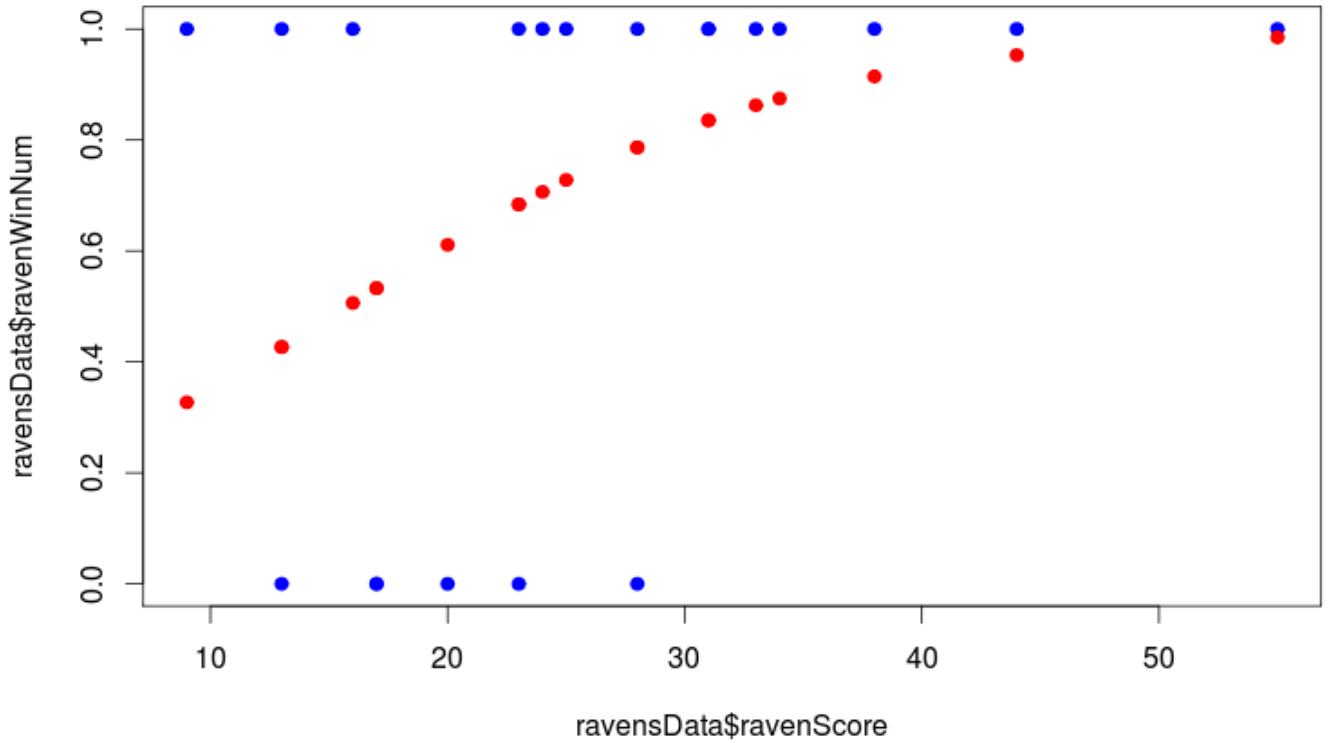


Логистическая регрессия

```
> logReg<-glm(
ravensData$ravenWinNum~ravensData$ravenScore,
family="binomial")
> logReg

Call:  glm(formula = ravensData$ravenWinNum ~
ravensData$ravenScore,
    family = "binomial")
Coefficients:
          (Intercept)  ravensData$ravenScore
              -1.6800                 0.1066
Degrees of Freedom: 19 Total (i.e. Null);  18
Residual
Null Deviance:      24.43
Residual Deviance: 20.89        AIC: 24.89
```

Предсказанные вероятности

```
> plot(ravensData$ravenScore, ravensData$ravenWinNum,
    pch=19, col='blue')
> points(ravensData$ravenScore, logReg$fitted, pch=19,
    col='red')
```

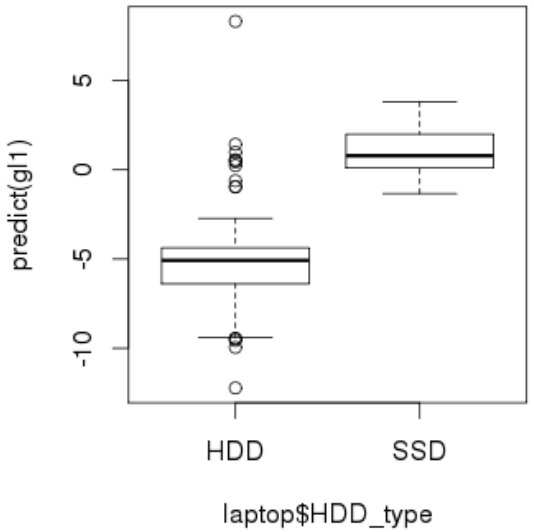


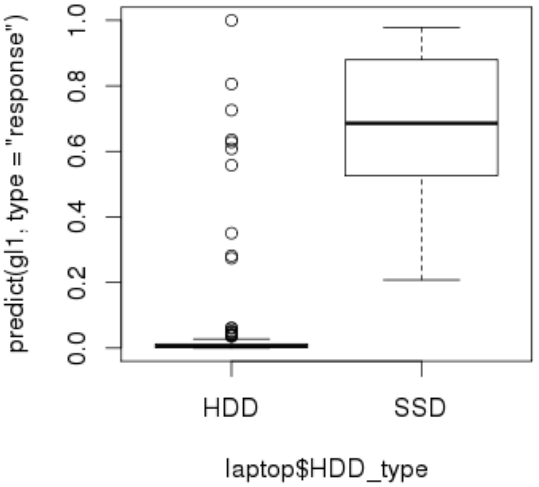
Кросс-валидация для категориальной зависимой переменной

- Пусть y – бинарная переменная (да-нет), например, больной – здоровый
- Пример: медицинский тест, измерены различные числовые и категориальные параметры x (температура, давление, есть ли кашель, …), необходимо предсказать y – болен ли пациент определенной болезнью
- Логистическая регрессия
- Предскажем наличие SSD в ноутбуке по его цене и объему диска

Кросс-валидация для glm

```
> gl1=glm(HDD_type ~ Price_RUR + HDD_Gb,
    family=binomial, data=laptop)
> gl1

Call:  glm(formula = HDD_type ~ Price_RUR + HDD_Gb,
    family = binomial,
        data = laptop)

Coefficients:
  (Intercept)    Price_RUR      HDD_Gb
-2.567427     0.000107    -0.009362

Degrees of Freedom: 305 Total (i.e. Null);  303 Residual
Null Deviance:      173
Residual Deviance: 60.3         AIC: 66.3

> plot(predict( gl1 ) ~ laptop$HDD_type)
> plot(predict( gl1, type="response" ) ~ laptop$HDD_type)
```





Рисуем вероятность, а не преобразованное значение

Кросс-валидация для glm

Как превратить вероятность в ответ (да-нет)? Выберем порог, например, 0.5

Определим функцию штрафов за отличия.

Два вектора одной длины: истинные **y** и предсказанные нами

```
> cost=function( y, ypred ) {
  mean( abs(y-ypred)>0.5 )
}
```

1 только там, где «не попали»

```
> cv1=cv.glm(laptop, gl1, cost, K=3)
> cv1$delta
[1] 0.0458 0.0436
```

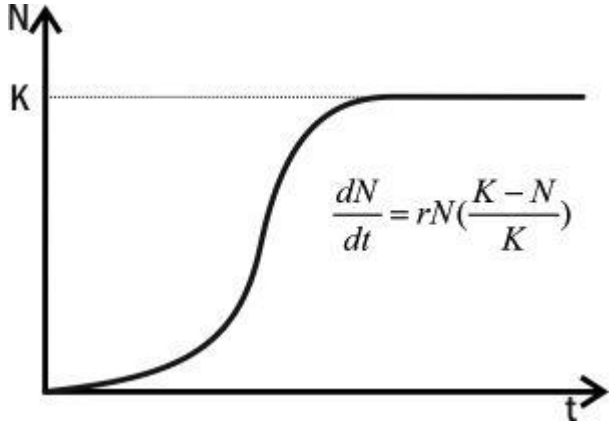
Non-linear least squares

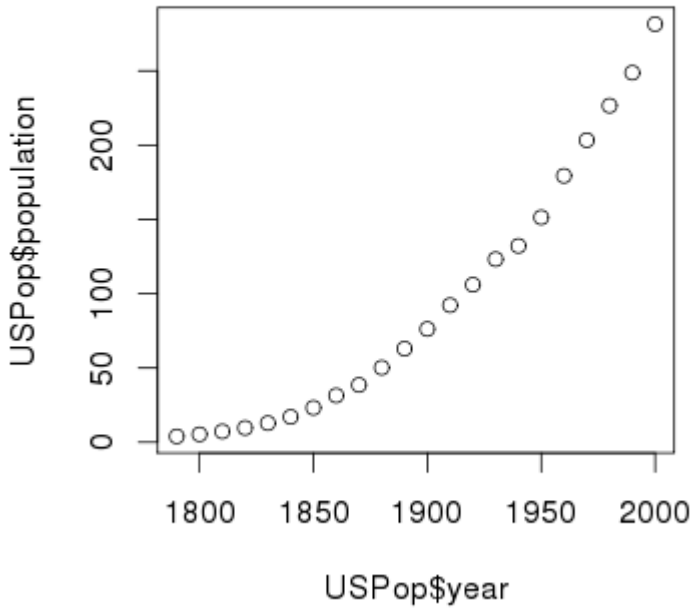
- Не всё можно выразить в виде суммы коэффициент*(что-то)+ коэффициент*(что-то)+…
- Не путайте glm и nls
- Вообще говоря, задача сводится к обходу пространства параметров и поиску такого набора параметров, при котором значение некоторой функции (напр., среднеквадратичное отклонение) минимально.
- Проблема локальных минимумов

Пример: логистический рост популяции

```
> install.packages("car")
> library(car)
> data(USPop)
> plot(USPop$year,
    USPop$population)
```

$$y_i = \frac{\beta_1}{1 + e^{\beta_2 + \beta_3 x_i}} + \varepsilon_i$$





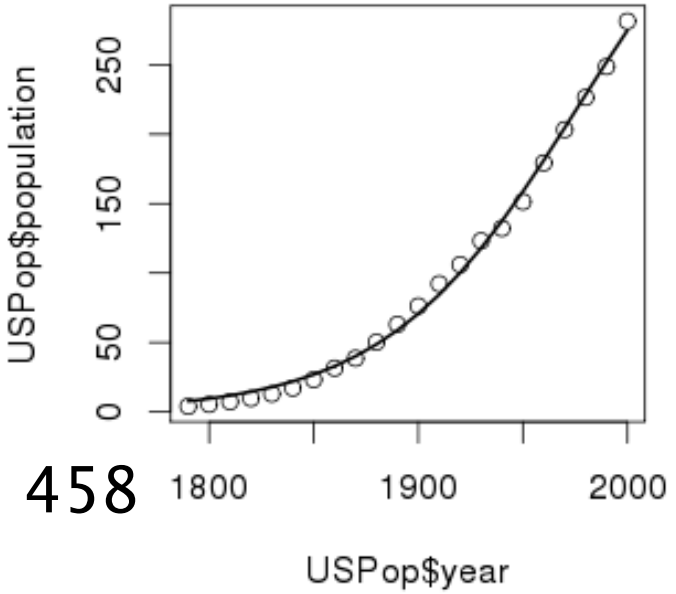
http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-nonlinear-regression.pdf

http://batrachos.com/Экспоненциальный_и_логистический

Пример: логистическая модель роста популяции

$$y_i = \frac{\beta_1}{1 + e^{\beta_2 + \beta_3 x_i}} + \varepsilon_i$$

```
> time <- 0:21
> pop.mod <- nls(
    population ~ beta1/(1 + exp(beta2 + beta3*time)),
    data=USPop,
    start=list(beta1 = 350, beta2 = 4.5, beta3 = -0.3),
    trace=T)
> pop.mod
…
   beta1    beta2    beta3
440.833    4.032   -0.216

residual sum-of-squares: 458
```



http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-nonlinear-regression.pdf