У меня есть набор данных с целевой переменной y и 1000 признаков. Предсказываю y по признакам. Далее отбираю 5 самых значимых. Есть ли подвох?

У меня есть набор данных с целевой переменной y и 1000 признаков. Предсказываю y по признакам. Далее отбираю самые 5 самых значимых. Есть ли подвох?

```
N <- 10000
M <- 1000
X <- matrix(rnorm(N * M, mean=0, sd = 1), nrow = N, ncol = M)
y <- rnorm(N, mean=0, sd=1)

model <- lm(y ~ X)
as.data.frame(summary(model)$coefficients) %>%  slice_min(order_by = `Pr(>|t|)`, n=5)
```

```
##          Estimate Std. Error   t value     Pr(>|t|)
## X426  0.03710488 0.01043125  3.557090 0.0003769144
## X159  0.03538670 0.01068585  3.311548 0.0009314676
## X609  0.03204934 0.01060356  3.022508 0.0025139492
## X144 -0.03177972 0.01061069 -2.995066 0.0027513184
## X761 -0.03116078 0.01071727 -2.907528 0.0036518726
```

У меня есть набор данных с целевой переменной y и 1000 признаков. Предсказываю y по признакам. Далее отбираю самые 5 самых значимых. Есть ли подвох?

```
N <- 10000
M <- 1000
X <- matrix(rnorm(N * M, mean=0, sd = 1), nrow = N, ncol = M)
y <- rnorm(N, mean=0, sd=1)

model <- lm(y ~ X)
as.data.frame(summary(model)$coefficients) %>%
    mutate(FDR=p.adjust(`Pr(>|t|)`, method="BH")) %>%
    slice_min(order_by = `Pr(>|t|)`, n=5)
```

```
##      Estimate Std. Error  t value    Pr(>|t|)       FDR
## 1 0.03204437 0.01066638 3.004240 0.002669777 0.8713444
## 2 0.03079443 0.01062565 2.898121 0.003763123 0.8713444
## 3 0.03068196 0.01061686 2.889928 0.003862508 0.8713444
## 4 0.02905613 0.01064368 2.729895 0.006347776 0.8713444
## 5 0.02896787 0.01075808 2.692663 0.007101514 0.8713444
```

У меня есть набор данных с целевой переменной y и 1000 признаков. Предсказываю y по признакам. Далее отбираю самые 5 самых значимых. Есть ли подвох?

```r
N <- 500
M <- 1000
X <- matrix(rnorm(N * M, mean=0, sd = 1), nrow = N, ncol = M)
y <- rnorm(N, mean=0, sd=1)

model <- lm(y ~ X)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
## ALL 500 residuals are 0: no residual degrees of freedom!
##
## Coefficients: (501 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.5132783         NA      NA       NA
## X1           0.3027199         NA      NA       NA
## X2          -0.9950539         NA      NA       NA
## X3          -0.7236734         NA      NA       NA
## X4           0.3946682         NA      NA       NA
## X5           0.0773044         NA      NA       NA
```

У меня есть набор данных с целевой переменной y и 1000 признаков. Предсказываю y по признакам. Далее отбираю самые 5 самых значимых. Есть ли подвох?

```r
N <- 500
M <- 1000
X <- matrix(rnorm(N * M, mean=0, sd = 1), nrow = N, ncol = M)
y <- rnorm(N, mean=0, sd=1)

model <- lm(y ~ X)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
## ALL 500 residuals are 0: no residual degrees of freedom!
##
## Coefficients: (501 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.5132783         NA      NA       NA
## X1           0.3027199         NA      NA       NA
## X2          -0.9950539         NA      NA       NA
## X3          -0.7236734         NA      NA       NA
## X4           0.3946682         NA      NA       NA
## X5           0.0773044         NA      NA       NA
```

Обычная линейная регрессия не может работать в случае, когда признаков больше, чем объектов

У меня есть набор данных с целевой переменной y и 1000 признаков. Выбираю 5 признаков, для которых максимальна разница между средним значением для y < 0 и y >= 0. Предсказываю y по признакам. Есть ли подвох?

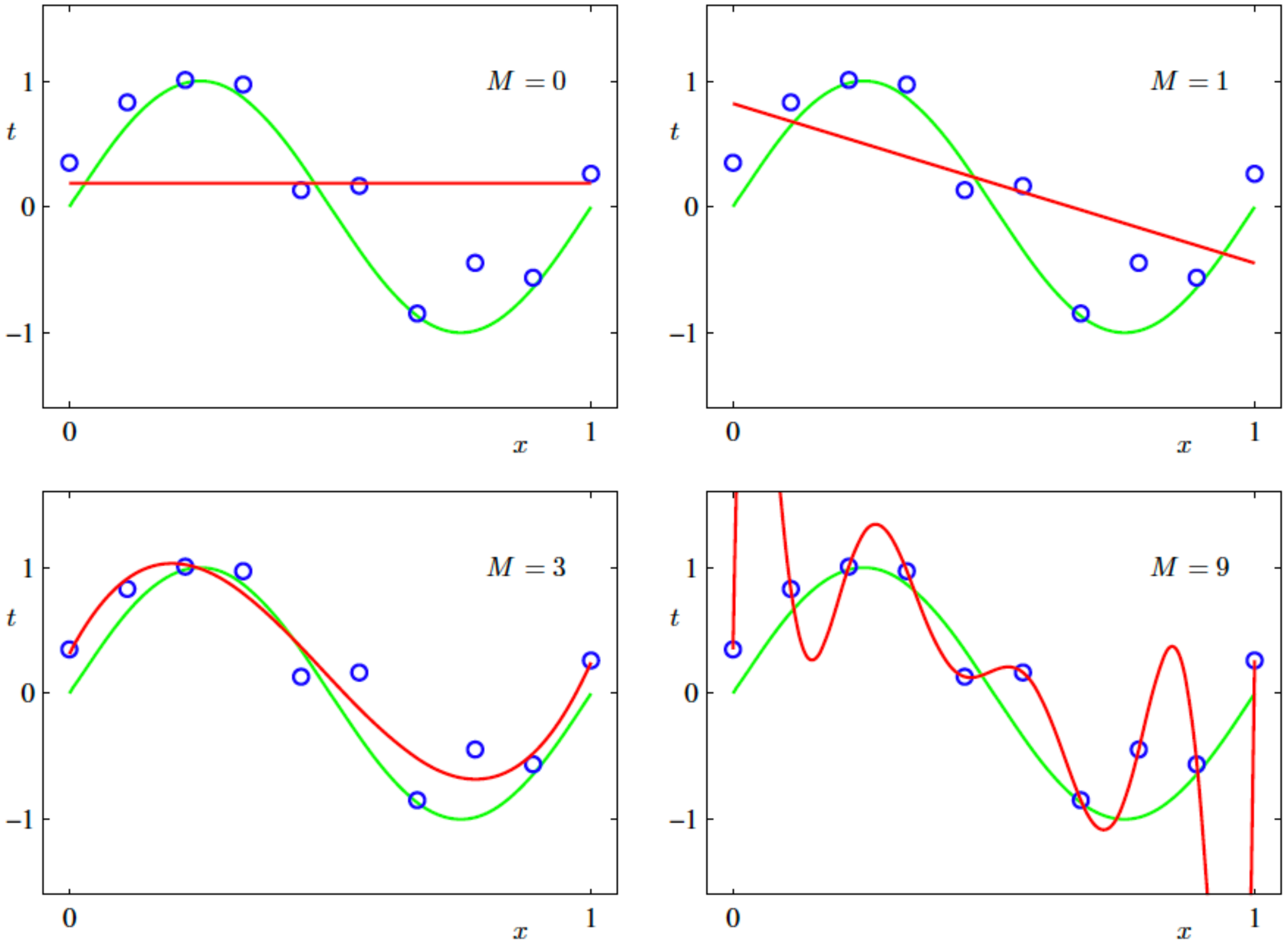
```r
N <- 500
M <- 1000
X <- matrix(rnorm(N * M, mean=0, sd = 1), nrow = N, ncol = M)
y <- rnorm(N, mean=0, sd=1)

H <- y < 0
IL <- y >= 0

abs_mean_diff <- abs(colMeans(X[H, ] ) - colMeans(X[IL, ]))
ranks <- rank(abs_mean_diff)
X_sel <- X[, ranks > 995]

model <- lm(y ~ X_sel)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ X_sel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7599 -0.6104 -0.0160  0.6325  2.9766
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03453    0.04582   0.754  0.45137
## X_sel1       0.12253    0.04573   2.680  0.00762 **
## X_sel2       0.13128    0.04455   2.947  0.00336 **
## X_sel3      -0.04323    0.04290  -1.008  0.31411
## X_sel4      -0.07578    0.04551  -1.665  0.09654 .
## X_sel5      -0.08723    0.04382  -1.991  0.04708 *
```

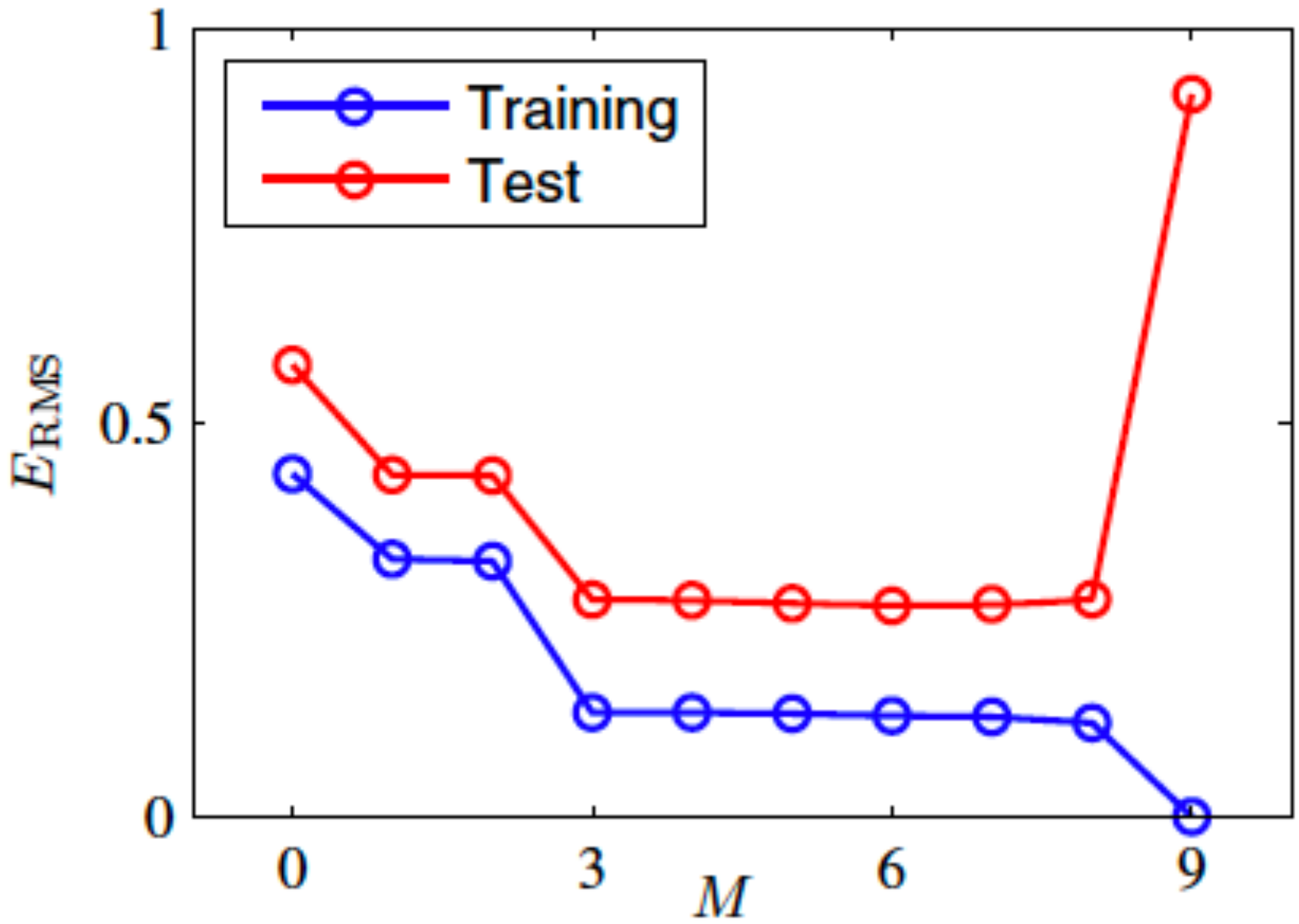
Неявно все равно устроили себе множественное тестирование

Можно попытаться отобрать признаки на основе их корреляции с предсказываемой величиной (и делать проверку значимости корреляции), но тогда слабое влияние, взаимодействие признаков и тд - пропускаем

Переобучение vs недообучение



Переобучение



Переобучение

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Регуляризация

Добавляем штраф за большие веса

$$MSE + penalty(w)$$

Виды штрафов:

$$L_1 = \alpha \sum_i |w_i|$$

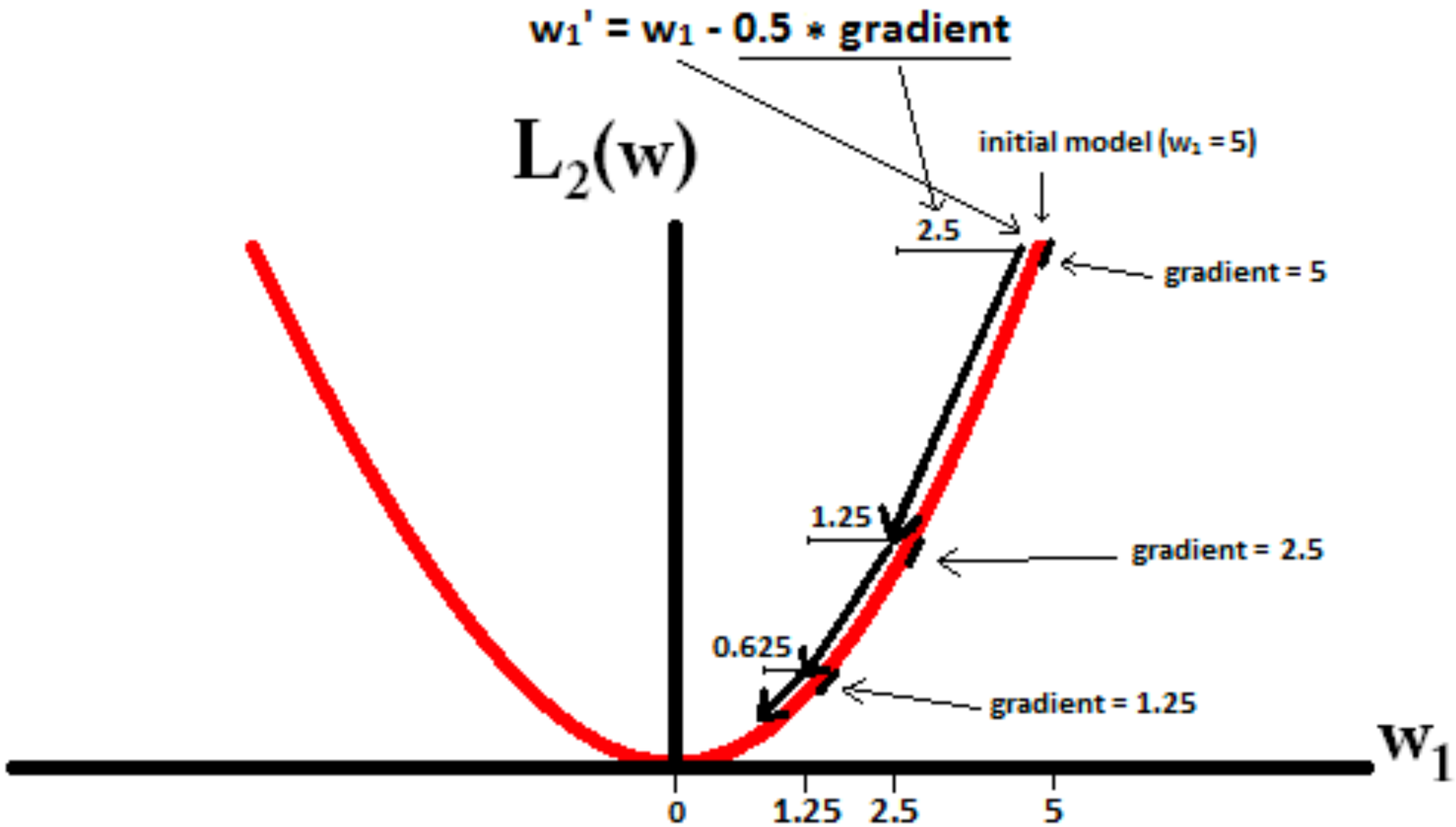
$$L_2 = \beta \sum_i w_i^2$$

$$L_{elastic} = \alpha \sum_i |w_i| + (1 - \alpha) \sum_i w_i^2$$

В чем отличие L1 от L2

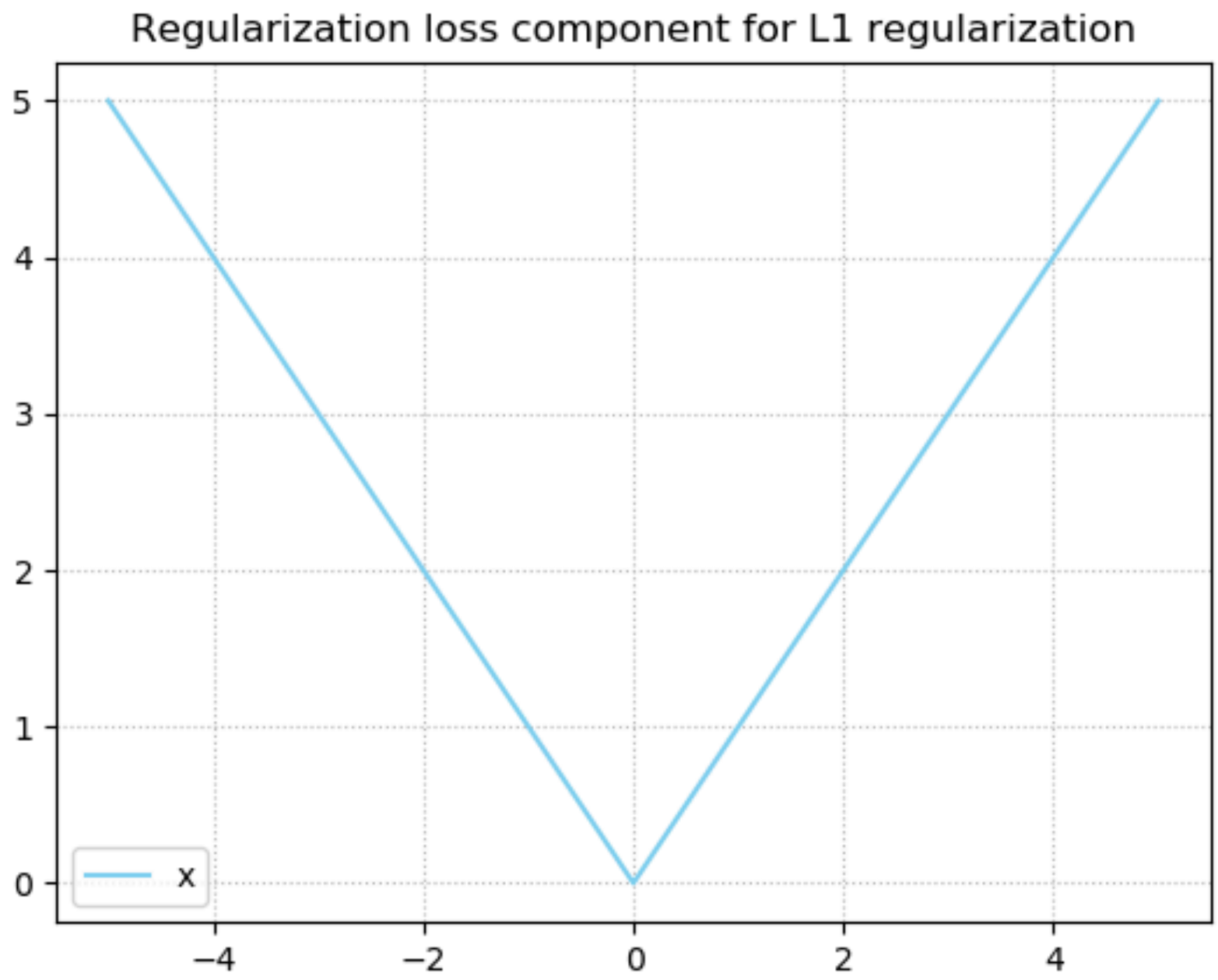
Как штрафуется модель за наличие больших весов?

L2-регуляризация



Штрафуем веса за то, что они большие

L1-регуляризация



Штрафуем веса за сам факт их существования, но независимо от величины веса

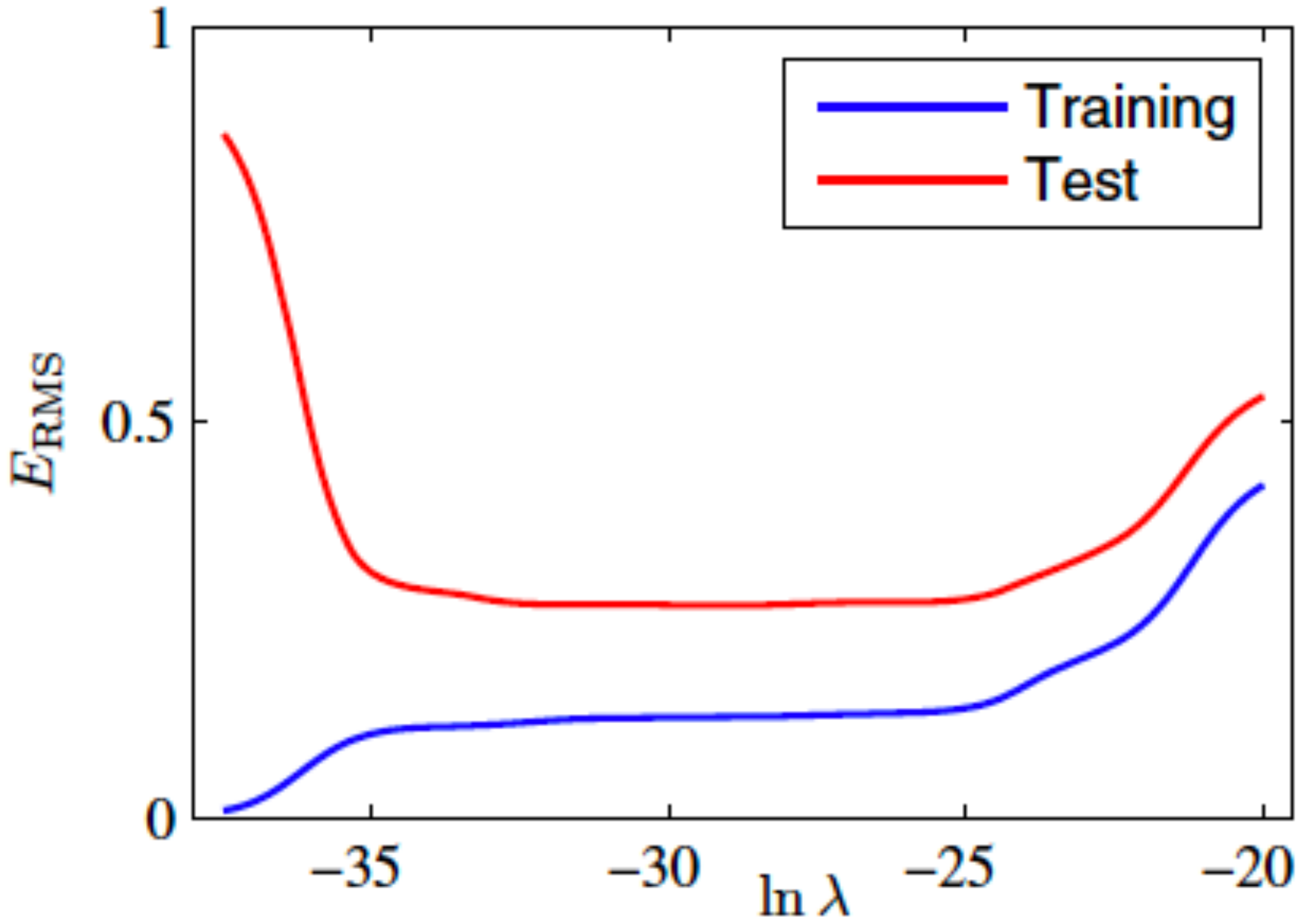
В чем отличие L1 от L2

Как штрафуется модель за наличие больших весов?

L1-регуляризация оставляет значимые весов, остальные зануляет

L2-регуляризация делает незнакомые веса близкими к 0

Действие регуляризации



Регуляризация

Регуляризация - это сообщение некоторой информации о весах, которую мы знаем без данных. Регуляризация - введение априорной вероятности.

Регуляризация

Добавляем штраф за большие веса, внезапно - это помогает и искать решение для случаев, когда число переменных > числа объектов

$$MSE + penalty(w)$$

Виды штрафов:

$$L_1 = \alpha \sum_i |w_i|$$

$$L_2 = \beta \sum_i w_i^2$$

$$L_{elastic} = \alpha \sum_i |w_i| + (1 - \alpha) \sum_i w_i^2$$

```r
# l1 loss
model <- glmnet(X, y,
                alpha = 1,
                lambda = 0.1)
coef(model)[1:5]
```

```
## [1] -0.018667980  0.000000000  0.005796671  0.000000000  0.000000000
```

```r
# l2 loss
model <- glmnet(X, y,
                alpha = 0,
                lambda = 0.1)
coef(model)[1:5]
```

```
## [1]  0.087247680 -0.001800099  0.036130725  0.037990325 -0.010083735
```

```r
# elastic loss
model <- glmnet(X, y,
                alpha = 0.5,
                lambda = 0.1)
coef(model)[1:5]
```

```
## [1] 0.007732438 0.000000000 0.020444167 0.000000000 0.000000000
```

Как подсчитать значимость?

Bootstrap

Изначальные объекты выборки

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Bootstrap-выборки

13	9	10	14	15	15	13	6	8	10	14	6	7	7	15

11	4	7	1	7	9	6	7	11	7	2	3	5	12	11

...

14	4	13	1	13	5	8	9	5	14	11	5	3	7	6

Bootstrap

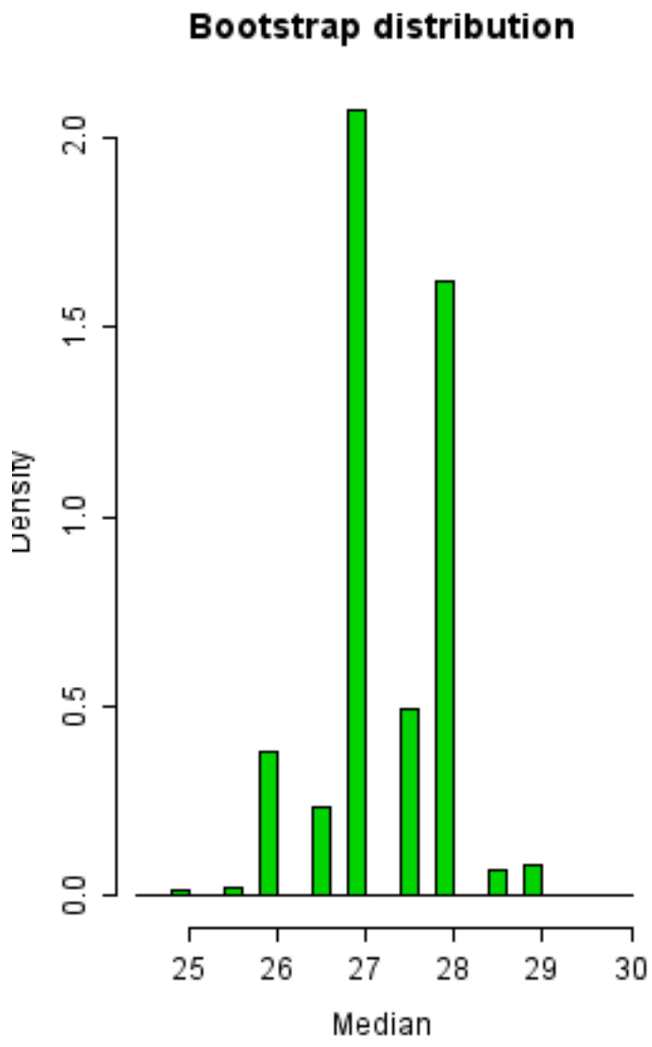
Для каждой bootstrap-выборки оцениваем значения коэффициентов, строим распределение для каждого коэффициента.

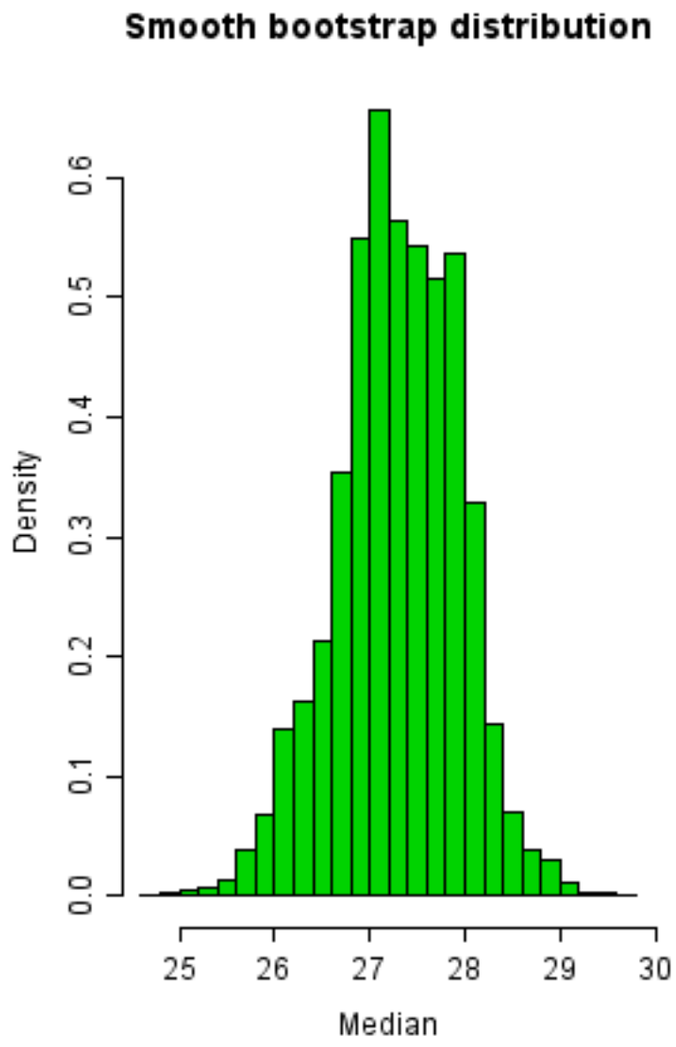
Bootstrap-выборки

13	9	10	14	15	15	13	6	8	10	14	6	7	7	15

Значения коэффициентов при признаках для данной bootstrap-выборки

Bootstrap


Bootstrap distribution
Density
0.0
0.5
1.0
1.5
2.0
25
26
27
28
29
30
Median


Smooth bootstrap distribution
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
25
26
27
28
29
30
Median

Кросс-валидация

Гиперпараметры

- У модели есть параметры и гиперпараметры
- Параметры модели учатся на основе выборки самой моделью (алгоритмом ее обучения)
- Гиперпараметры - это параметры, которые задаем мы и которые влияют на то, как модель учит параметры

Примеры гиперпараметров?

Примеры гиперпараметров?

1. Регуляризация - какая регуляризация и с каким коэффициентом
2. Степень полинома, которым мы аппроксимировали функцию
3. Что-то еще?

Примеры гиперпараметров?

1. Регуляризация - какая регуляризация и с каким коэффициентом

2. Степень полинома, которым мы аппроксимировали функцию

3. Признаки, которые мы даем модели - тоже гиперпараметры!

Train-test split

Обучение (train) **Тест (test)**

Обучаем модель на train, проверяем качество модели на test.

Train-test split?

Обучение (train) **Тест (test)**

Обучаем модель на train, проверяем качество модели на test.

Как подбирать гиперпараметры модели? - Никак

Train-validation-test split!

Обучение (train) **Валидация (validation)** **Тест (test)**

1) **Выбираем некоторые значения гиперпараметров**
2) **Обучаем модель с такими гиперпараметрами на train**
3) **Смотрим качество на validation**
4) **Пробуем таким образом много разных значений гиперпараметров и выбираем то, которое дает наилучшее**

Train-validation-test split!

Обучение (train) **Валидация (validation)** **Тест (test)**

1) **Выбираем некоторые значения гиперпараметров**
2) **Обучаем модель с такими гиперпараметрами на train**
3) **Смотрим качество на validation**
4) **Пробуем таким образом много разных значений гиперпараметров и выбираем то, которое дает наилучшее**

Какие минусы подхода?

Train-validation-test split?

Обучение (train) **Валидация (validation)** **Тест (test)**

Какие минусы подхода?

1) **Существенно уменьшаем объем данных, на которых учится модель**
2) **Большая нестабильность оценки качества при сравнении моделей из-за малого размера выборки**

Кросс-валидация

Обучение (train)

Тест (test)

...

Много разбиений на train и вариацию. На каждом разбиении выбираем лучшие гиперпараметры. Потом смотрим, какие значения гиперпараметров встречаются чаще всего, на основании чего делаем вывод об итоговых значениях гиперпараметров

Что еще можно оценить?

Кросс-валидация

Обучение (train)

Тест (test)

...

Что еще можно оценить?

Для данного набора значений гиперпараметров можем оценить среднее качество модели и дисперсию по разным разбиениям

Кросс-валидация. Как разбивать?

Обучение (train)

Тест (test)

Кросс-валидация. Leave-one-out cross-validation

...

На каждой итерации в валидацию попадает ровно один объект. По остальным учимся

Кросс-валидация. Leave-one-out cross-validation

...

На каждой итерации в валидацию попадает ровно один объект. По остальным учимся

Какие минусы?

Кросс-валидация. Leave-one-out cross-validation

...

Какие минусы?

1) **Невозможно оценить некоторые метрики, подразумевающие, например, что в валидации у нас есть оба класса**
2) **Склонна завышать качество, так как хотя бы один похожий объект в обучении найдется**
3) **Есть формула для оценки качества, получаемого на leave-one-out кросс-валидации**

Кросс-валидация. Монте-карло кросс-валидация

...

На каждой итерации случайно выбираем какой-то процент объектов в валидацию

Кросс-валидация. Монте-карло кросс-валидация

...

На каждой итерации случайно выбираем какой-то процент объектов в валидацию

Какие минусы?

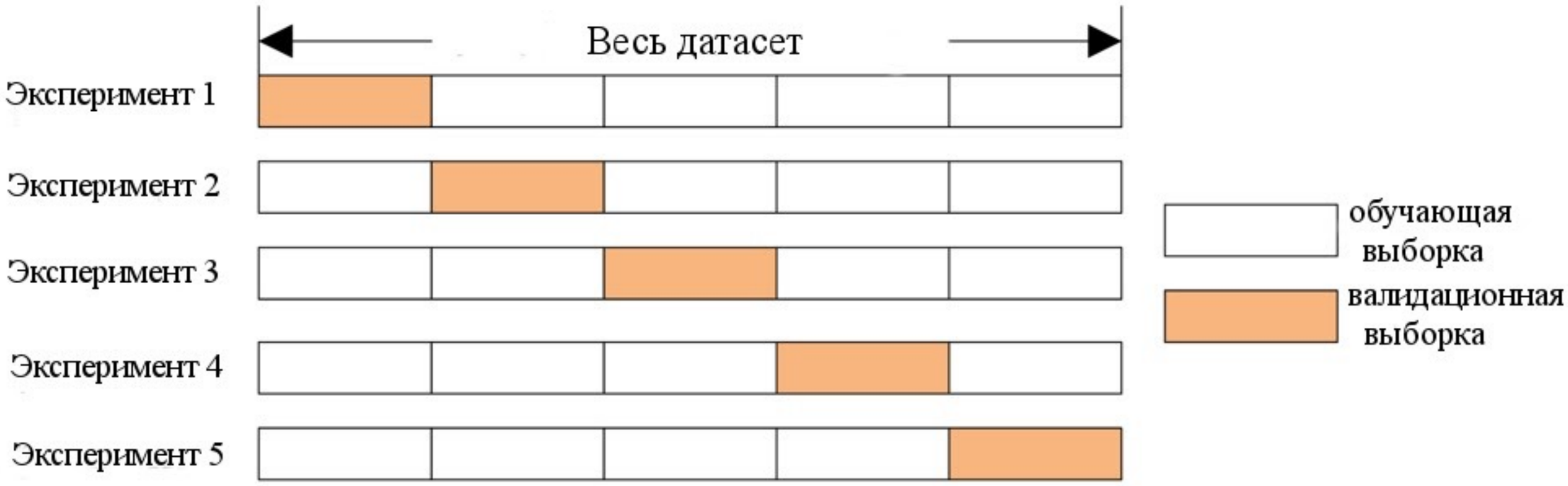
Кросс-валидация. Монте-карло кросс-валидация

Какие минусы?

1) **Нет гарантий, что все объекты побывают и в обучении, и в валидации**

Кросс-валидация. K-fold кросс-валидация

Почему картинка неверна?



Кросс-валидация. K-fold кросс-валидация

Тест отдельно должен быть

Обучающая выборка

Тест (test)

Кросс-валидация. K-fold кросс-валидация

Вся обучающая выборка

Какие минусы?

Тест (test)

Кросс-валидация. K-fold кросс-валидация

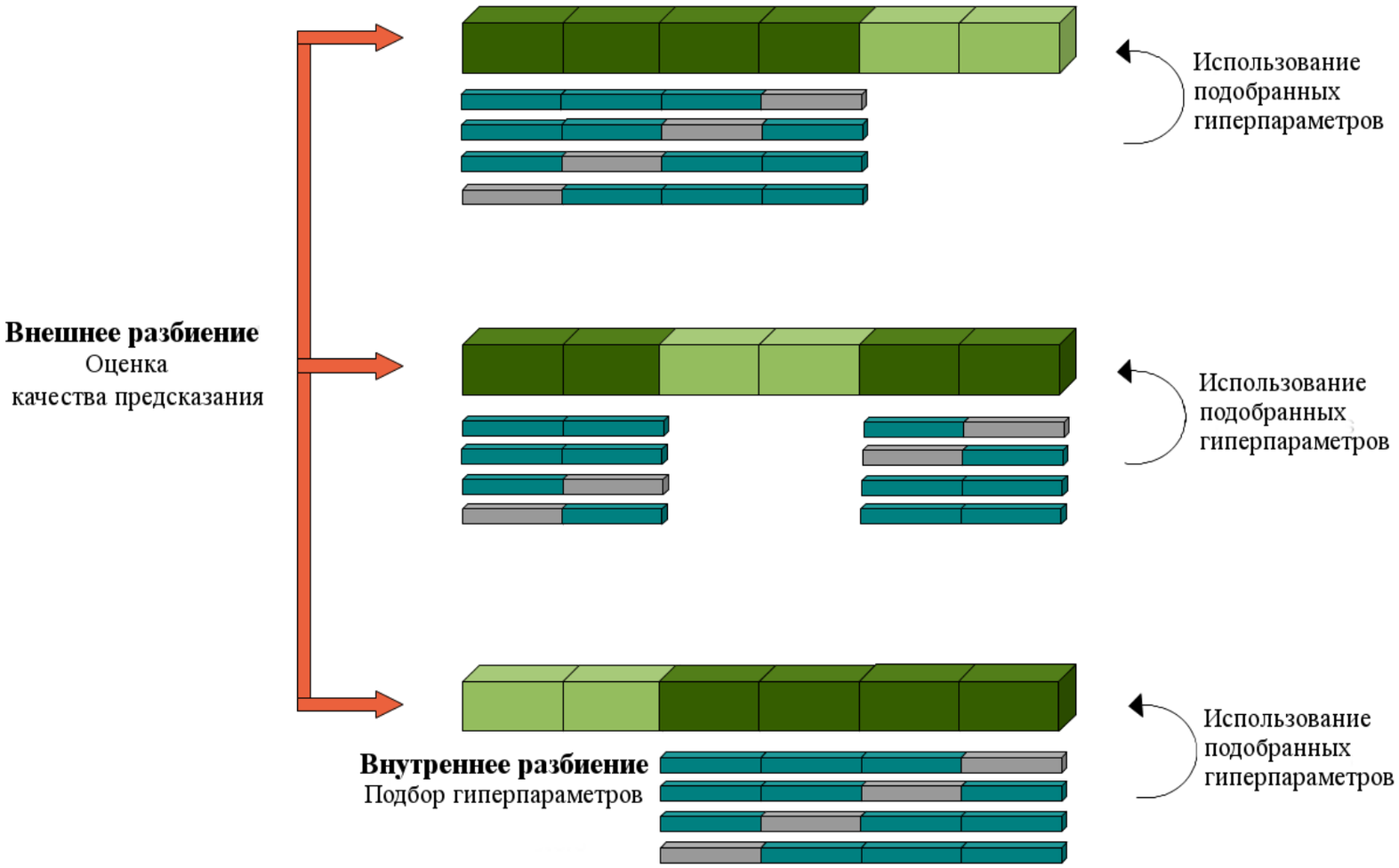
Вся обучающая выборка

Какие минусы?

1) **Не совсем понятно, сколько блоков брать**
2) ***хотелось бы не откусывать тест**

Тест (test)

Вложенная кросс-валидация



Все равно лучше иметь независимые данные для тестирования финальной модели

Кросс-валидация в биологии

Какие проблемы у любой предложенной валидации?

Кросс-валидация в биологии

Какие проблемы у любой предложенной валидации?

Она не учитывает домена, в котором мы работаем.

Для каждой задачи надо отдельно думать, как правильно сделать валидацию.

Кросс-валидация в биологии

Какие проблемы у любой предложенной валидации?

Она не учитывает домена, в котором мы работаем.

Для каждой задачи надо отдельно думать, как правильно сделать валидацию.

Например, если у вас есть данные пациентов из разных больниц, то правильно делать так, чтобы пациенты из одной больницы были либо все в обучении, либо все в валидации

Предсказываем цены на ноутбуки

```
laptop <- read.csv("laptop_price.csv")
head(laptop)
```

```
##   Manufacturer  Model Processor Memory_Gb HDD_Gb HDD_type Price_RUR
## 1        Acer Aspire  i3-3110M         4    500      HDD     16400
## 2        Acer Aspire  i3-3120M         4    500      HDD     16500
## 3        Acer Aspire  i5-3230M         4    500      HDD     18500
## 4        Acer Aspire   \xd1-70         2    500      HDD     12000
## 5        Acer Aspire   \xd1-70         2    500      HDD     12000
## 6        Acer Aspire     1007U         2    500      HDD     11300
##   Screen_size_inch Battery_capacity_mAh   OS      Color
## 1             15.6                4400 Win8      black
## 2             15.6                4400 Win8      black
## 3             15.6                4400 Win8      black
## 4             11.6                2500 Win8 turquoise
## 5             11.6                2500 Win8      black
## 6             11.6                5000 Win8 turquoise
```

Просто память

```
model <- lm(Price_RUR ~ Memory_Gb, data=laptop)
summary(model)
```

```
##
## Call:
## lm(formula = Price_RUR ~ Memory_Gb, data = laptop)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -16662  -6292  -2558     790   64438
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5022.8     1651.2   3.042  0.00256 **
## Memory_Gb     4442.4      333.1  13.335  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12310 on 304 degrees of freedom
## Multiple R-squared:  0.3691, Adjusted R-squared:  0.367
## F-statistic: 177.8 on 1 and 304 DF,  p-value: < 2.2e-16
```

Несколько переменных

```
model <- lm(Price_RUR ~ Memory_Gb +  Screen_size_inch + HDD_Gb, data=laptop)
summary(model)
```

```
##
## Call:
## lm(formula = Price_RUR ~ Memory_Gb + Screen_size_inch + HDD_Gb,
##     data = laptop)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -24879  -5552  -1505   2670  61021
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      16475.481   4437.238   3.713 0.000244 ***
## Memory_Gb         7266.167    406.667  17.868  < 2e-16 ***
## Screen_size_inch  -511.390    350.186  -1.460 0.145237
## HDD_Gb             -31.022      3.305  -9.387  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10650 on 302 degrees of freedom
## Multiple R-squared:  0.5308, Adjusted R-squared:  0.5262
## F-statistic: 113.9 on 3 and 302 DF,  p-value: < 2.2e-16
```

Факторная переменная

```
model <- lm(Price_RUR ~ Manufacturer, data=laptop)
summary(model)
```

```
##
## Call:
## lm(formula = Price_RUR ~ Manufacturer, data = laptop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32377  -8255  -2499    3490  73174
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        21198.6     1415.0  14.981  < 2e-16 ***
## ManufacturerApple  46078.5     3527.5  13.063  < 2e-16 ***
## ManufacturerAsus     206.2     1714.4   0.120 0.904357
## ManufacturerDell    7427.6     2079.1   3.573 0.000411 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12090 on 302 degrees of freedom
## Multiple R-squared:  0.3958, Adjusted R-squared:  0.3898
## F-statistic: 65.95 on 3 and 302 DF,  p-value: < 2.2e-16
```

Кодирование меток

A/G -> 0, T/C -> 1, ...

Какой минус?

Кодирование меток

A/G -> 0, T/C -> 1, ...

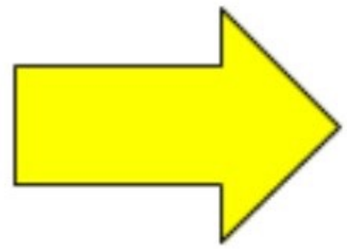
Какой минус?

Задаем неявную информацию о том, что A/G > T/C и тд

Использовать только вместе с сортировкой по предсказываемой величине

One-hot encoding

Цвет
Красный
Красный
Желтый
Зеленый
Желтый



Красный	Желтый	Зеленый
1	0	0
1	0	0
0	1	0
0	0	1

Факторная переменная

- Влияет ли цвет ноутбука на его цену?
- Модель, если x – число: $y_i = \alpha x_{1i} + \beta x_{2i} + \varepsilon_i$
- Если x – фактор, то такая запись не подходит. Вместо этого:

$$y_i = \alpha_1 I(x_{1i}==black) + \alpha_2 I(x_{1i}==white) + \ldots + \varepsilon_i$$

Коэффициент (подбираются при построении модели)

Индикатор (равен 1, если x – черный цвет, иначе 0)

Если две факторные переменные?

$$y_i = \alpha_1 I(x_{1i}==black) + \alpha_2 I(x_{1i}==white) + \ldots +$$

$$+ \beta_1 I(x_{2i}==Apple) + \beta_2 I(x_{2i}==ASUS) + \ldots + \varepsilon_i$$

Факторная переменная

```
model <- lm(Price_RUR ~ Manufacturer, data=laptop)
summary(model)
```

```
##
## Call:
## lm(formula = Price_RUR ~ Manufacturer, data = laptop)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32377  -8255  -2499   3490  73174
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        21198.6     1415.0  14.981  < 2e-16 ***
## ManufacturerApple  46078.5     3527.5  13.063  < 2e-16 ***
## ManufacturerAsus     206.2     1714.4   0.120 0.904357
## ManufacturerDell    7427.6     2079.1   3.573 0.000411 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12090 on 302 degrees of freedom
## Multiple R-squared:  0.3958, Adjusted R-squared:  0.3898
## F-statistic: 65.95 on 3 and 302 DF,  p-value: < 2.2e-16
```

Почему на одно меньше, чем производителей?

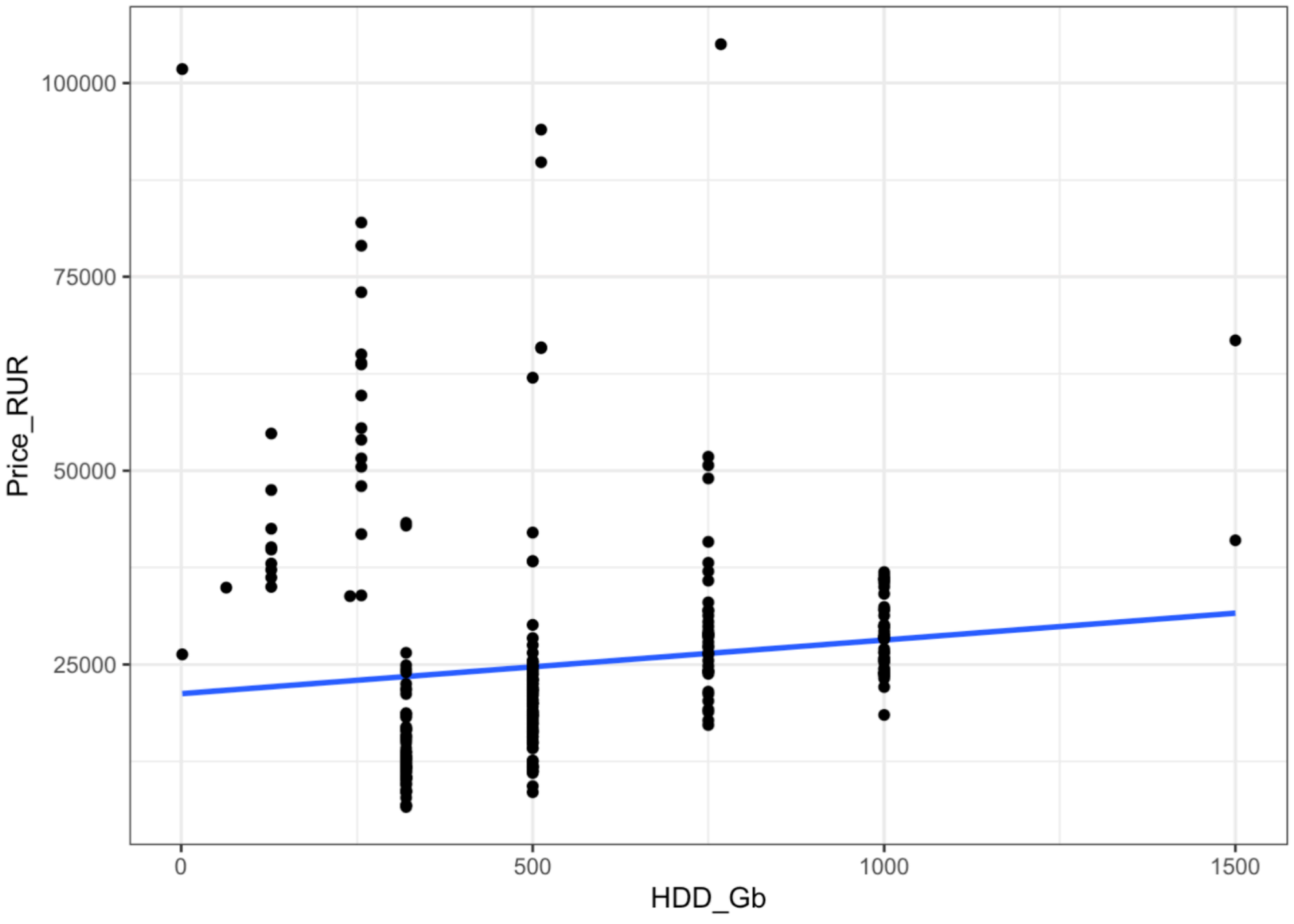
Числа и факторы

```
model <- lm(Price_RUR ~ HDD_Gb, data=laptop)
summary(model)
```

```
##
## Call:
## lm(formula = Price_RUR ~ HDD_Gb, data = laptop)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -16851  -9244  -3445   1912  80551
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21238.584   2027.553  10.475   <2e-16 ***
## HDD_Gb          6.913      3.410   2.027   0.0435 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15400 on 304 degrees of freedom
## Multiple R-squared:  0.01334,    Adjusted R-squared:  0.01009
## F-statistic: 4.109 on 1 and 304 DF,  p-value: 0.04352
```

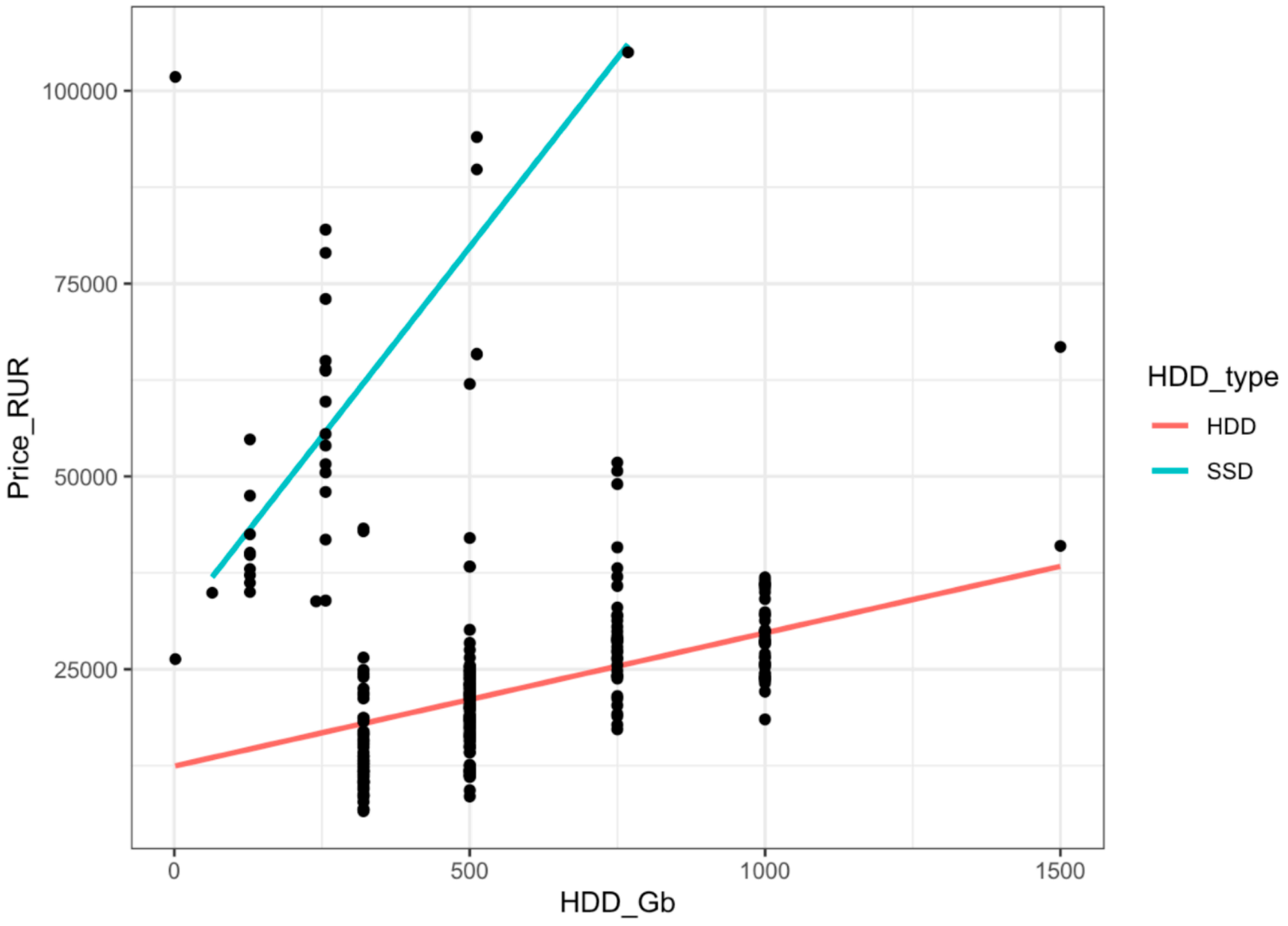
```
laptop %>% ggplot(aes(y=Price_RUR, x=HDD_Gb)) + geom_smooth(method="lm", se=F) + geom_point() + theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
laptop %>% ggplot(aes(y=Price_RUR, x=HDD_Gb)) + geom_smooth(aes(color=HDD_type), method="lm", se=F) + geom_point
() + theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Числа и факторы

```
model <- lm(Price_RUR ~ HDD_Gb + HDD_type, data=laptop)
summary(model)
```

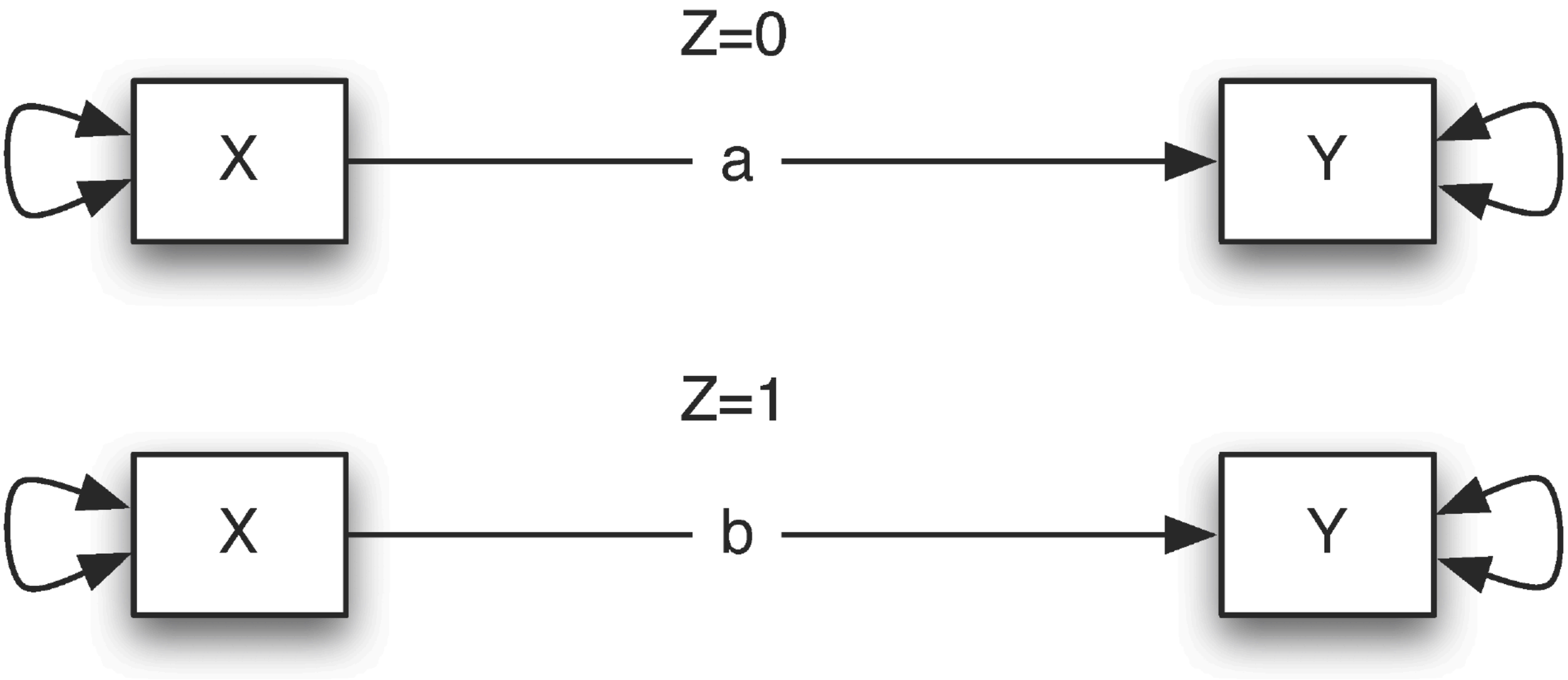
```
##
## Call:
## lm(formula = Price_RUR ~ HDD_Gb + HDD_type, data = laptop)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -22833  -5948  -1886   2889  91028
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10741.160   1594.347   6.737 8.14e-11 ***
## HDD_Gb         20.290      2.591   7.830 8.27e-14 ***
## HDD_typeSSD 40797.575   2442.199  16.705  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11130 on 303 degrees of freedom
## Multiple R-squared:  0.4864, Adjusted R-squared:  0.483
## F-statistic: 143.5 on 2 and 303 DF,  p-value: < 2.2e-16
```

Числа и факторы

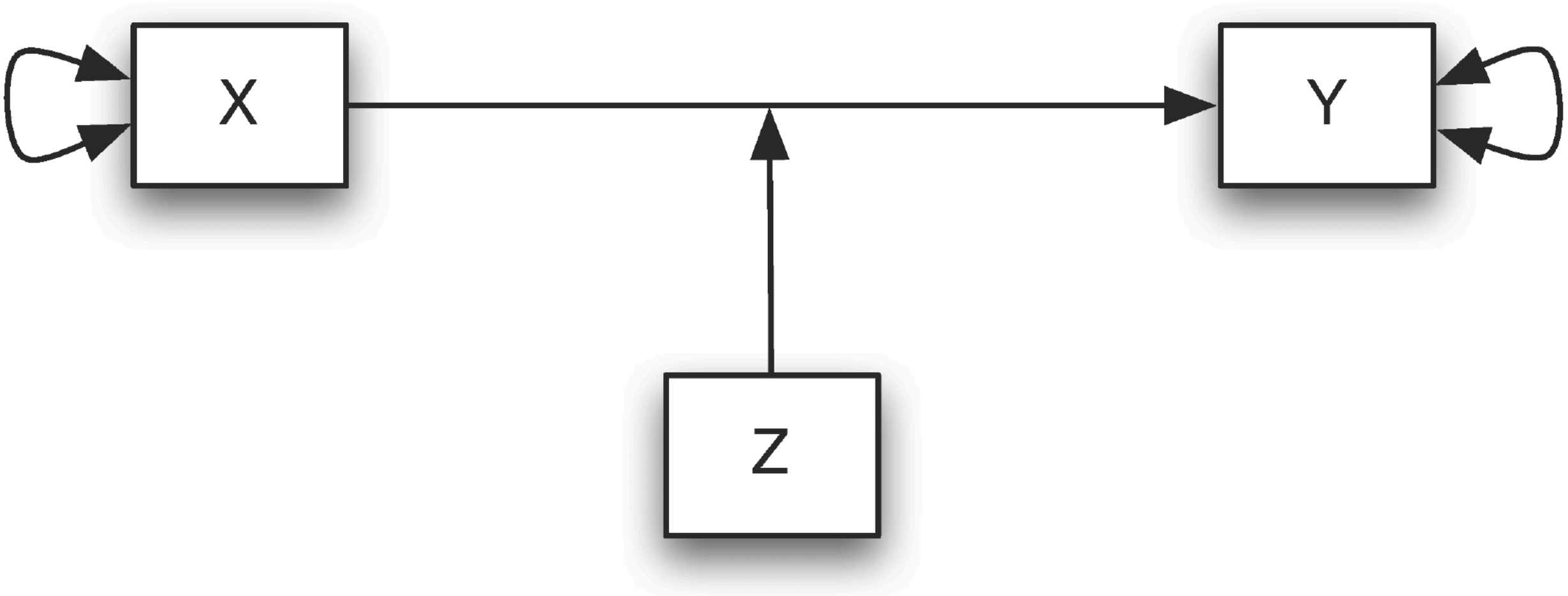
```
model <- lm(Price_RUR ~ HDD_Gb + HDD_type + HDD_Gb:HDD_type, data=laptop)
summary(model)
```

```
##
## Call:
## lm(formula = Price_RUR ~ HDD_Gb + HDD_type + HDD_Gb:HDD_type,
##     data = laptop)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21886  -6049  -1461   2885  89344
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         12430.529   1525.776   8.147 9.97e-15 ***
## HDD_Gb                 17.270      2.488   6.941 2.38e-11 ***
## HDD_typeSSD         18232.081   4265.934   4.274 2.58e-05 ***
## HDD_Gb:HDD_typeSSD     80.870     12.874   6.281 1.17e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10480 on 302 degrees of freedom
## Multiple R-squared:  0.5457, Adjusted R-squared:  0.5412
## F-statistic: 120.9 on 3 and 302 DF,  p-value: < 2.2e-16
```

Moderation effect



Moderation effect



Как выглядит линейная модель

$$Y = \beta_0 + \beta_1 \cdot X + \beta_2 \cdot Z + \beta_3 \cdot X \cdot Z + \epsilon$$

Вклад X **Вклад Z** **moderation effect Z**

Как выглядит линейная модель

$$Y = \beta_0 + \beta_1 \cdot X + \beta_2 \cdot Z + \beta_3 \cdot X \cdot Z + \epsilon$$

Вклад X **Вклад Z** **moderation effect Z**

Если Z либо 0, либо 1, то как выглядит?

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon, \;\; Z = 0$$

$$Y = \beta_0 + \beta_2 + (\beta_1 + \beta_3) \cdot X + \epsilon, \;\; Z = 1$$

Числа и факторы

```
model <- lm(Price_RUR ~ HDD_Gb * HDD_type, data=laptop)
summary(model)
```

```
##
## Call:
## lm(formula = Price_RUR ~ HDD_Gb * HDD_type, data = laptop)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21886  -6049  -1461   2885  89344
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         12430.529   1525.776   8.147 9.97e-15 ***
## HDD_Gb                 17.270      2.488   6.941 2.38e-11 ***
## HDD_typeSSD         18232.081   4265.934   4.274 2.58e-05 ***
## HDD_Gb:HDD_typeSSD     80.870     12.874   6.281 1.17e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10480 on 302 degrees of freedom
## Multiple R-squared:  0.5457, Adjusted R-squared:  0.5412
## F-statistic: 120.9 on 3 and 302 DF,  p-value: < 2.2e-16
```

```r
library(psych)
```

```r
example <- lm(bdi ~ stateanx*epiNeur, data=epi.bfi)
example
```

```
##
## Call:
## lm(formula = bdi ~ stateanx * epiNeur, data = epi.bfi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0493  -2.2513  -0.4707   2.1135  11.9949
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.06367    2.18559   0.029   0.9768
## stateanx           0.03750    0.06062   0.619   0.5368
## epiNeur           -0.14765    0.18869  -0.782   0.4347
## stateanx:epiNeur   0.01528    0.00466   3.279   0.0012 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.12 on 227 degrees of freedom
## Multiple R-squared:  0.4978, Adjusted R-squared:  0.4912
## F-statistic: 75.02 on 3 and 227 DF,  p-value: < 2.2e-16
```

Обобщенные линейные модели

Иногда линейные модели не просто неточны - а неверны. Пример:

1) Как зависит число людей на пляже от температуры? f(-20) = -100?

2) Как зависит вероятность одного человека пойти на пляж от температуры (вероятность находится в пределах [0,1])

Обобщенные линейные модели

Можем искать ответ в виде нелинейной функции от линейной (берем линейную регрессию, считаем у и берем от него некоторую функцию)

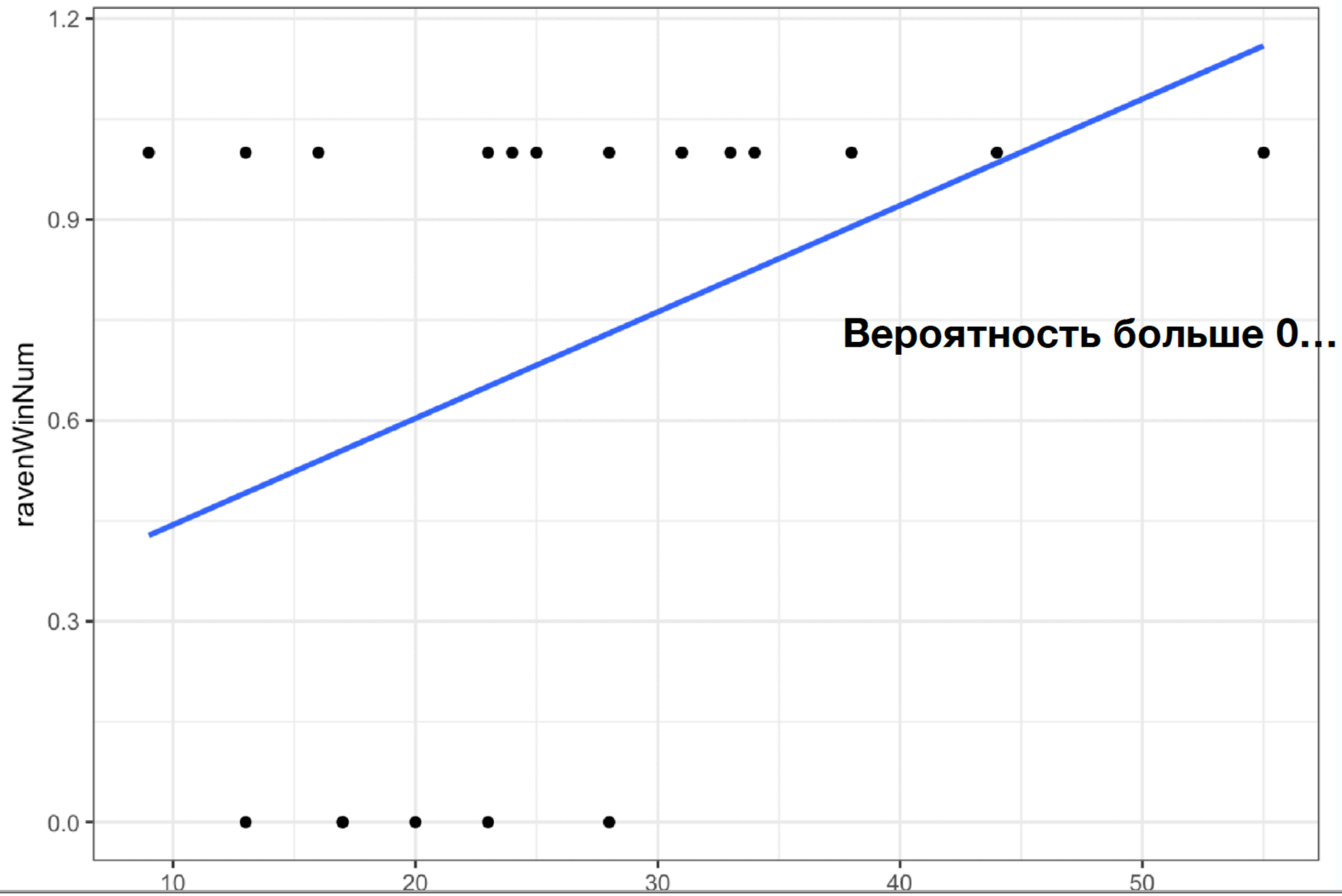
Таблица выигрышей команды

```
# https://github.com/jtleek/dataanalysis/blob/master/week5/003countOutcomes/data/ravensData.rda
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
```

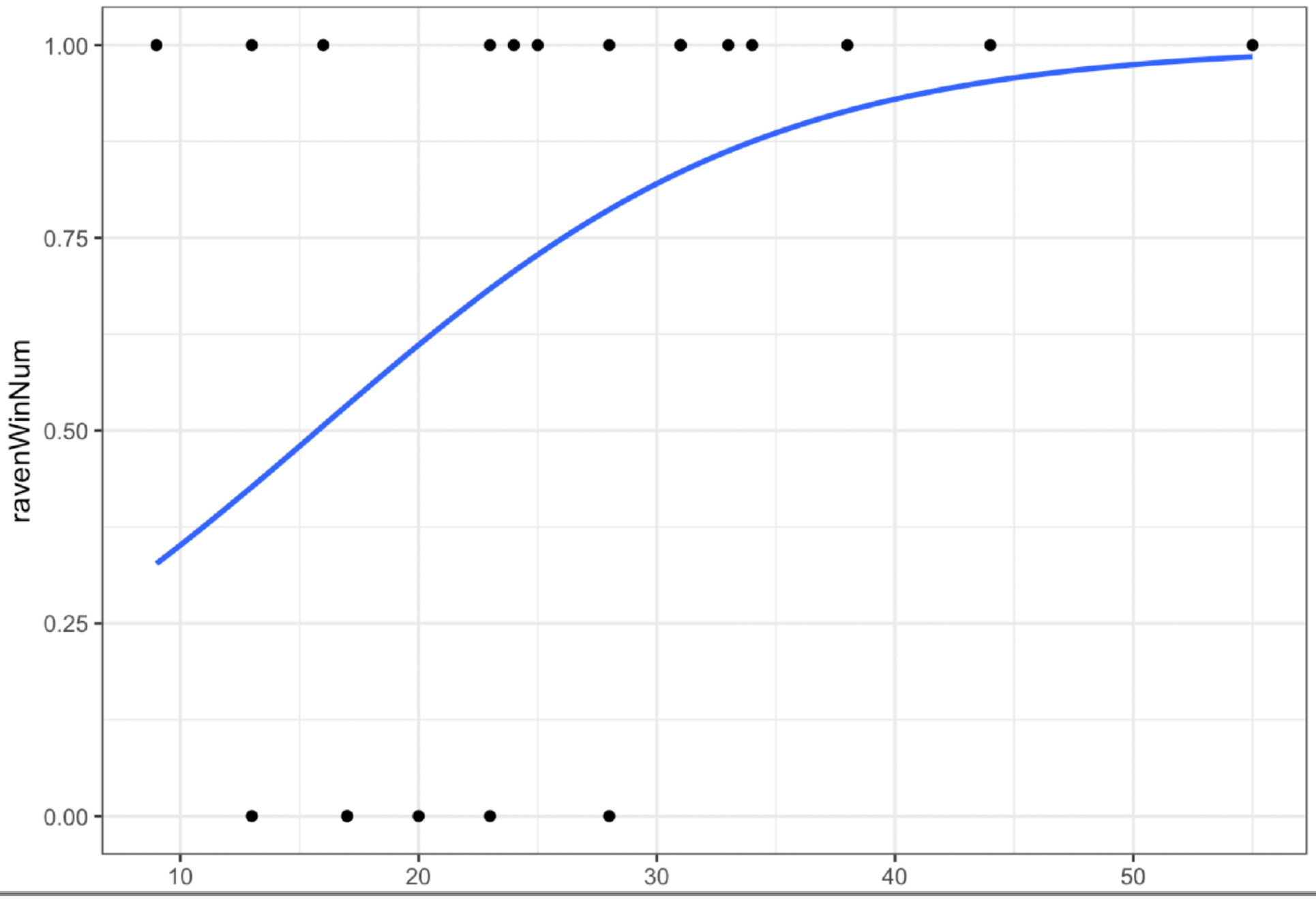
Линейная регрессия((

```
ravensData %>% ggplot(
    aes(y=ravenWinNum, x=ravenScore)) +
    geom_smooth(method='lm', se=F) +
    geom_point() + theme_bw()
```



Логистическая регрессия

```
ravensData %>% ggplot(
    aes(y=ravenWinNum, x=ravenScore)) +
    geom_smooth(method="glm",
                method.args = list(family = "binomial"),
                se=F) +
    geom_point() + theme_bw()
```



Логистическая регрессия

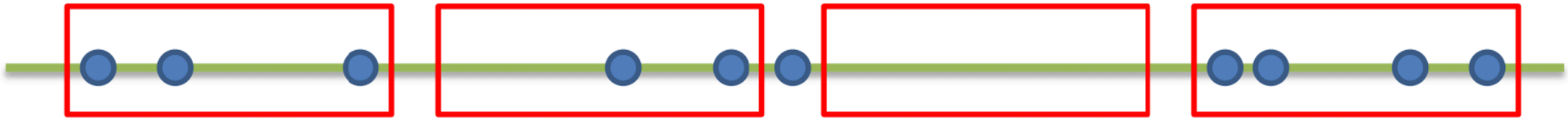
```
glm2 <- glm(ravenWinNum ~ ravenScore,
            family = "binomial",
            data=ravensData)
summary(glm2)
```

```
##
## Call:
## glm(formula = ravenWinNum ~ ravenScore, family = "binomial",
##     data = ravensData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7575  -1.0999   0.5305   0.8060   1.4947
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.68001    1.55412  -1.081     0.28
## ravenScore   0.10658    0.06674   1.597     0.11
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24.435  on 19  degrees of freedom
## Residual deviance: 20.895  on 18  degrees of freedom
## AIC: 24.895
##
## Number of Fisher Scoring iterations: 5
```

Используется другая метрика качества модели - AIC
Чем меньше- тем лучше

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

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

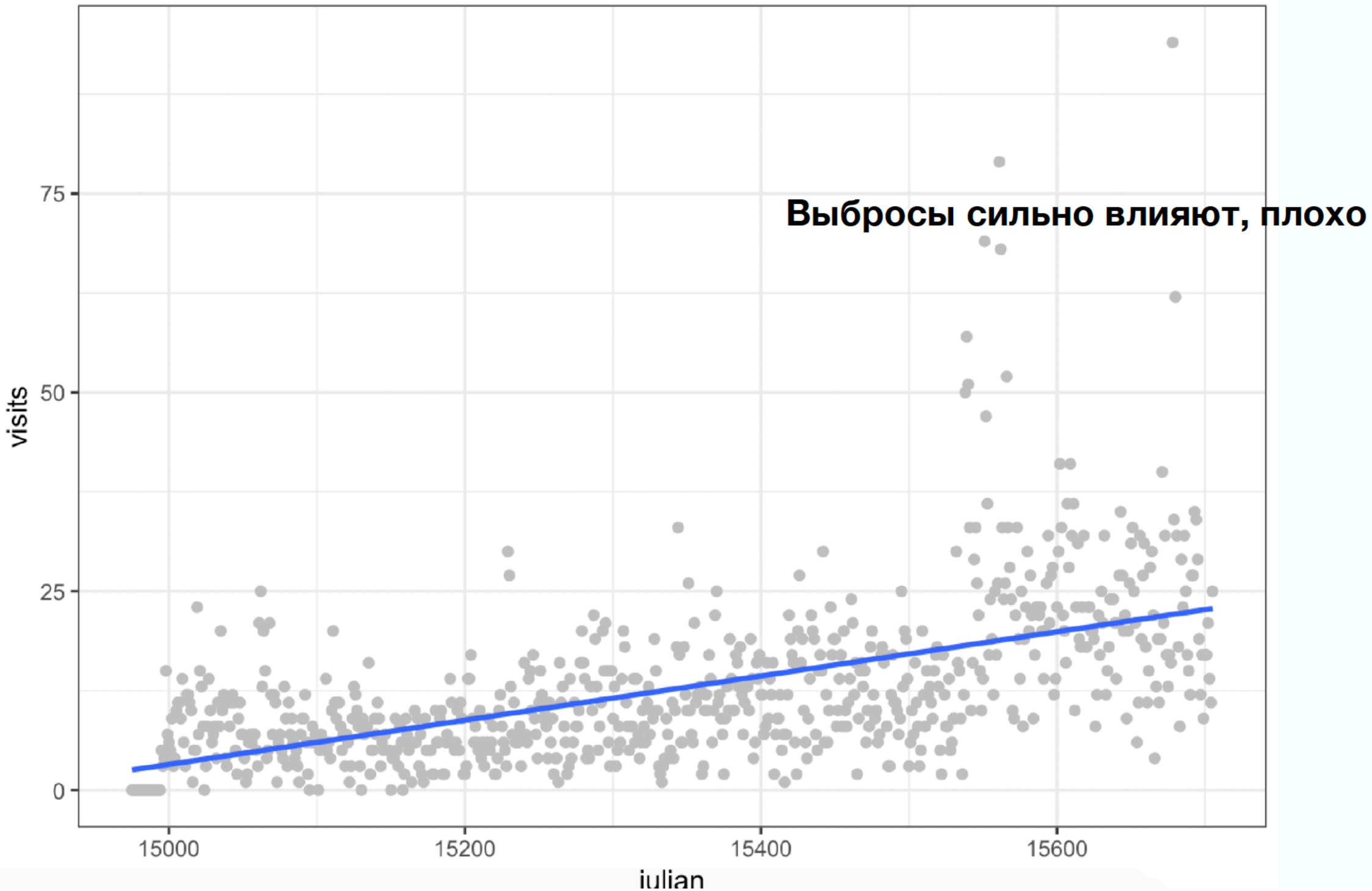
Число визитов в театр

```
# https://github.com/jtleek/dataanalysis/blob/master/week5/003countOutcomes/data/gaData.rda
load("gaData.rda")
head(gaData)
```

```
##         date visits simplystats
## 1 2011-01-01      0           0
## 2 2011-01-02      0           0
## 3 2011-01-03      0           0
## 4 2011-01-04      0           0
## 5 2011-01-05      0           0
## 6 2011-01-06      0           0
```

```
gaData$julian <- julian(gaData$date)
gaData %>% ggplot(aes(x=julian, y=visits)) +
    geom_point(color='grey') +
    geom_smooth(method="lm", se=F) +
    theme_bw()
```

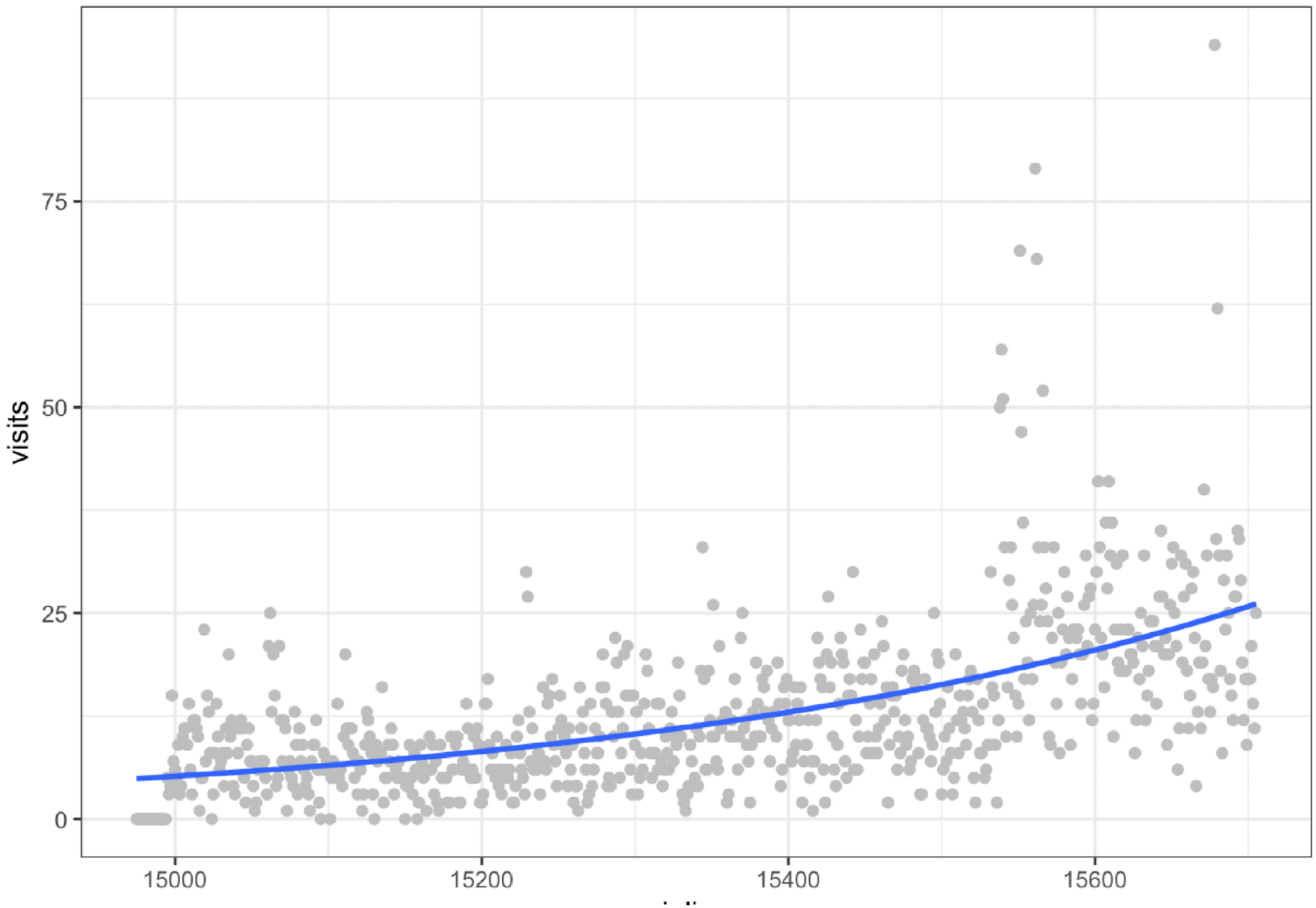
Простая линейная модель



Выбросы сильно влияют, плохо

Пуассон

```
gaData %>% ggplot(aes(x=julian, y=visits)) +
    geom_point(color='grey') +
    geom_smooth(method="glm",
                method.args = list(family = "poisson"),
                se=F) +
    theme_bw()
```



```
glm1 <- glm(visits ~ julian, family = "poisson", data=gaData)
summary(glm1)
```

```
##
## Call:
## glm(formula = visits ~ julian, family = "poisson", data = gaData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0466  -1.5908  -0.3198   0.9128  10.6545
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.275e+01  8.130e-01  -40.28   <2e-16 ***
## julian       2.293e-03  5.266e-05   43.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5150.0  on 730  degrees of freedom
## Residual deviance: 3121.6  on 729  degrees of freedom
## AIC: 6069.6
##
## Number of Fisher Scoring iterations: 5
```

Используется другая метрика качества модели - AIC
Чем меньше- тем лучше