Логистическая регрессия

Г. Мороз

Логистическая регрессия

Логистическая регрессия или логит-регрессия (logistic regression, logit regression) была описана в работе [Cox 1958] и применяется в случаях, когда зависимая переменная принимает два значения а предикторы могут быть как числовыми, так и категориальными.

шансы, натуральный логарифм

Мы хотим чего-то такого:

$$\underbrace{y}_{[-\infty,+\infty]} = \underbrace{\beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_k \cdot x_k + \varepsilon_i}_{[-\infty,+\infty]}$$

Вероятность — (в классической статистике) отношение количества успехов к общему числу событий:

$$p = \frac{\#\text{ успехов}}{\#\text{ неудач} + \#\text{ успехов}}, \text{ область значений: } [0, 1]$$

Шансы — отношение количества успехов к количеству неудач:

$$odds = \frac{p}{1-p} = \frac{\text{p(успеха)}}{\text{p(неудачи)}}, \text{ область значений: } [0, +\infty]$$

Натуральный логарифм шансов:

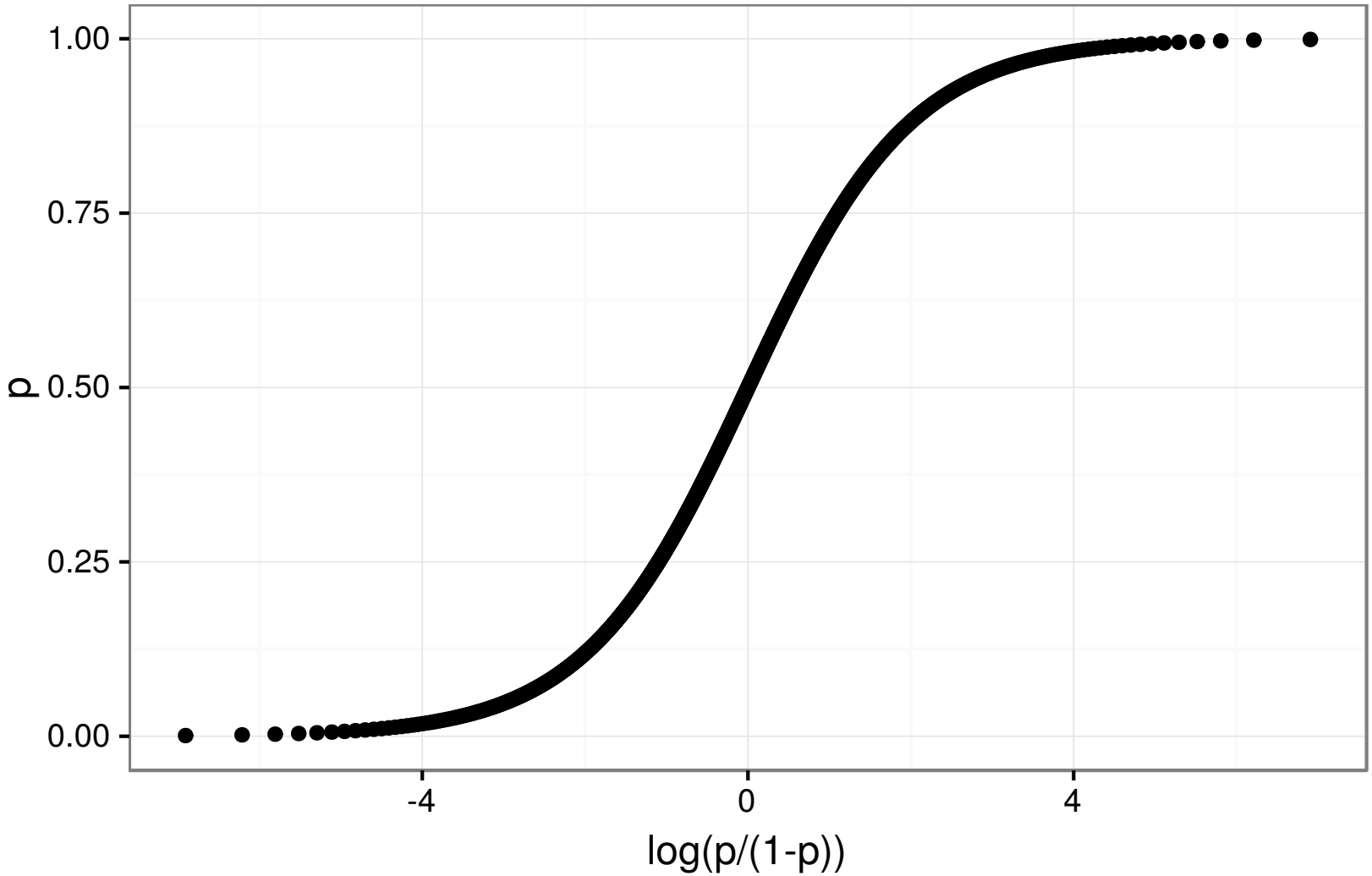
$$log(odds), \text{ область значений: } [-\infty, +\infty]$$

вероятность $\longleftrightarrow$ логарифм шансов

# у	# н	p	$odds$	$log(odds)$
1	9	0.1	0.11	-2.20
2	8	0.2	0.25	-1.39
3	7	0.3	0.43	-0.85
4	6	0.4	0.67	-0.41
5	5	0.5	1	0
6	4	0.6	1.5	0.41
7	3	0.7	2.33	0.85
8	2	0.8	4	1.39
9	1	0.9	9	2.20

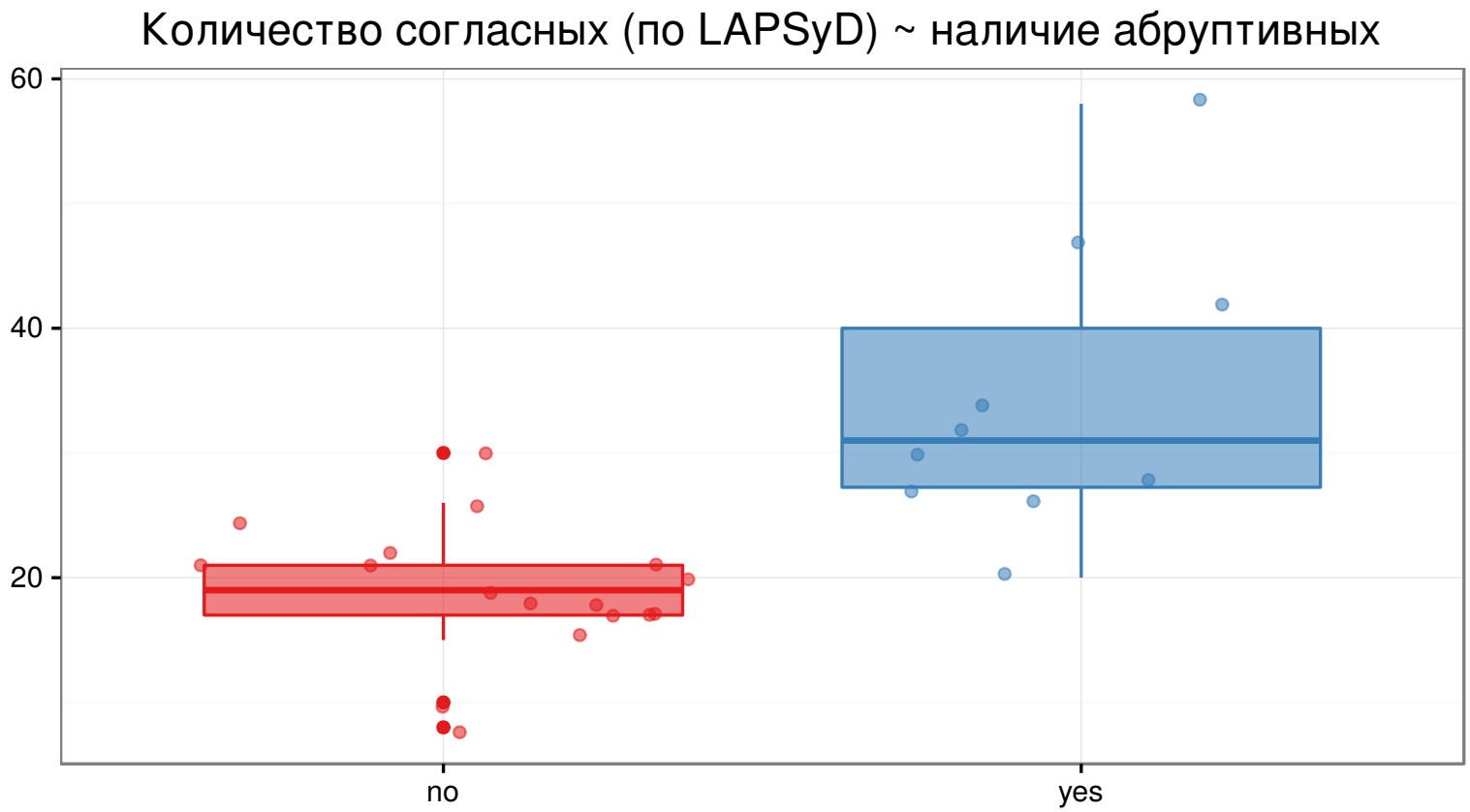
```
a <- 1:9
b <- 9:1
p <- a/(b+a)
lo <- log(p/(1-p))

p <- exp(lo)/(1 + exp(lo))
```

сигмоида



Задача 1

Проанализируем данные, содержащих выборку языков с указанием количества согласных и наличия в данном языке абруптивных согласных. На графике представлен результат (можно посмотреть более интерактивный вариант):



Модель без предикторов

```
df <- read.csv("http://goo.gl/0btfKa")
fit1 <- glm(ejectives ~1, data = df, family = "binomial") # логит-регрессия
summary(fit1)

Call:
glm(formula = ejectives ~1, family = "binomial", data = df)      # формула

Deviance Residuals:                                  # распределение остатков
      Min         1Q     Median         3Q        Max
  -0.9619    -0.9619    -0.9619     1.4094     1.4094

Coefficients:                                        # коэфициенты модели
             Estimate   Std. Error   z value   Pr(>|z|)
(Intercept)   -0.5306       0.3985    -1.331      0.183    # β0

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 35.594 on 26 degrees of freedom
Residual deviance: 35.594 on 26 degrees of freedom
AIC: 37.594                                          # критерий Акаике

Number of Fisher Scoring iterations: 4
```

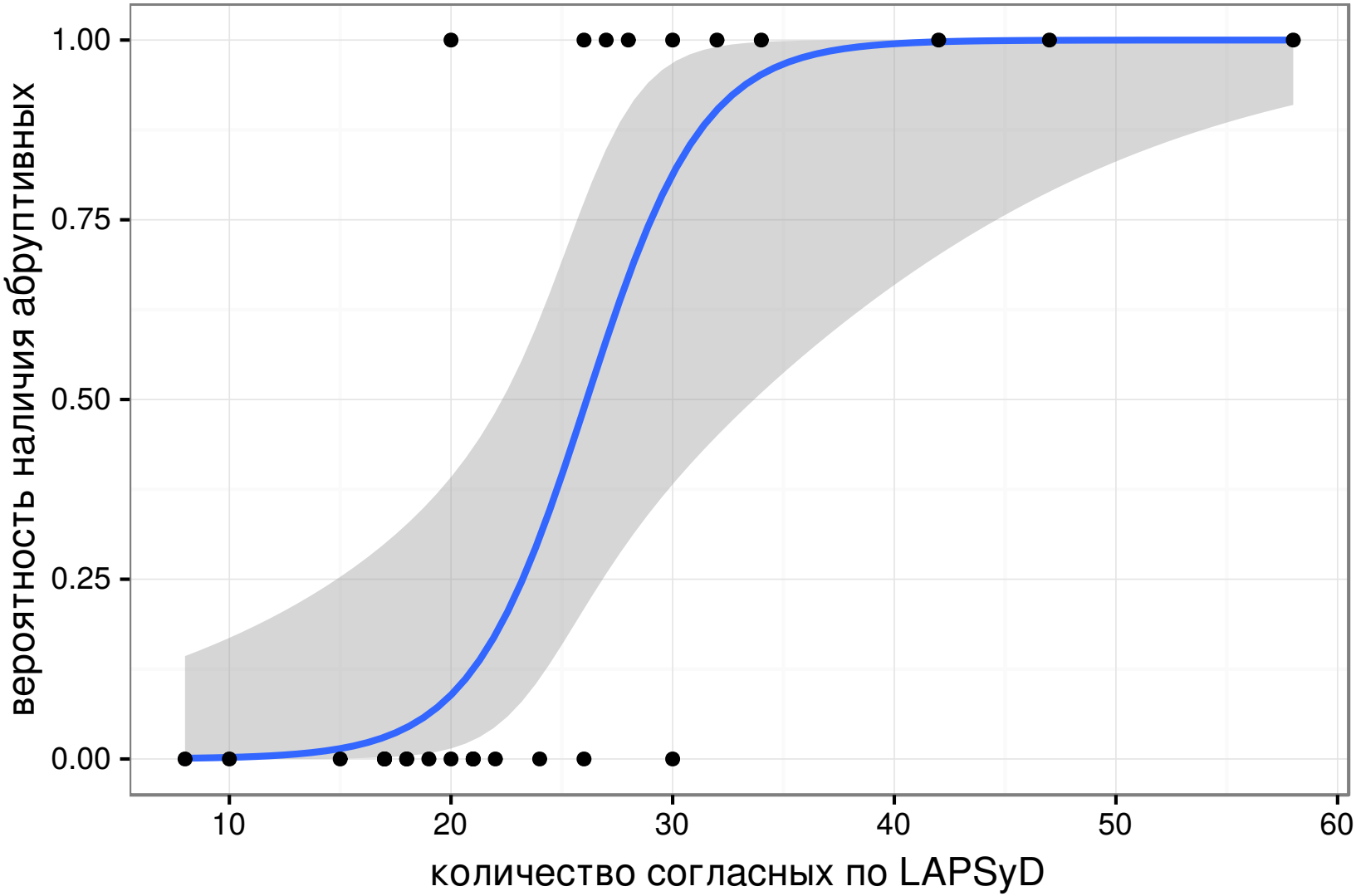
```
table(df$ejectives)                  # а сколько у нас языков с абруптивами?
 no yes
 17  10
log(10/17)
-0.5306283                           # так вот как получен коэфициент β0...
```

Модель с числовым предиктором

```
df <- read.csv("http://goo.gl/0btfKa")
fit2 <- glm(ejectives ~n.cons.lapsyd, data = df, family = "binomial")
summary(fit2)
```

```
Call:
glm(formula = ejectives ~n.cons.lapsyd, family = "binomial", data = df)

Deviance Residuals:                                  # распределение остатков
      Min         1Q     Median         3Q        Max
  -1.8317    -0.4742    -0.2481     0.1914     2.1997

Coefficients:                                        # коэфициенты модели
                Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)      -9.9204      3.7699   -2.631    0.0085  **    # β0
n.cons.lapsyd     0.3797      0.1495    2.540    0.0111   *    # β1

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 35.594 on 26 degrees of freedom
Residual deviance: 16.202 on 25 degrees of freedom
AIC: 20.202                                          # критерий Акаике

Number of Fisher Scoring iterations: 6
```

Визуализация: ggplot2



Визуализация: ggplot2

```
df <- read.csv("http://goo.gl/0btfKa")

str(df)
'data.frame': 27 obs. of 3 variables:
$ name: Factor w/ 27 levels "Abkhaz","Amharic",..: 25 15 22 16 24 19 14 7 ...
$ n.cons.lapsyd: int 24 21 21 22 21 20 19 18 15 17 ...
$ ejectives: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

# Нужно переделать значения переменной ejectives в 0 и 1
df$ejectives.value <- as.numeric(df$ejectives) - 1

library(ggplot2)
ggplot(data = df, aes(x = n.cons.lapsyd, y = ejectives.value))+
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  geom_point()
```

Чтобы убрать доверительный интервал, можно добавить аргумент `ls = F` в функцию geom_smooth().

Интерпретация

Какова вероятность по нашим данным, что в языке с 29 согласными есть абруптивные звуки?

$$\log(odds) \text{ или } \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \times \text{n.cons.lapsyd}$$

$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$\beta_0 + \beta_1 \times \text{n.cons.lapsyd} = -9.9204 + 0.3797 \times 29 = 1.0909$$

$$\frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = \frac{e^{1.0909}}{1 + e^{1.0909}} = 0.7485512$$

Т. е. в соответствии с нашими данными, вероятность, что в языке с 29 согласными есть абруптивные звуки примерно 3 к 1 или 0.75.

Предсказания, на основе модели

Неужели необходимо помнить все эти формулы?

```
new <- data.frame(n.cons.lapsyd = 29)
predict(fit2, new, type="response")
         1
 0.7485964
```

Функция predict() принимает на вход построенную модель (не обязательно логистическую) и датафрейм со столбцами, использованными для построения модели. Естественно, значений может быть несколько.

```
new <- data.frame(n.cons.lapsyd = 27:31)
predict(fit2, new, type="response")
         1          2          3          4          5
 0.5821783  0.6707173  0.7485964  0.8131865  0.8641927
```

Чтобы получить не вероятности, а значение шансов (*odds*), следует в аргументе type указать значение link (это значение по умолчанию).

```
new <- data.frame(n.cons.lapsyd = 27:31)
predict(fit2, new, type="link")
         1          2          3          4          5
 0.3317220  0.7114312  1.0911404  1.4708496  1.8505588
```

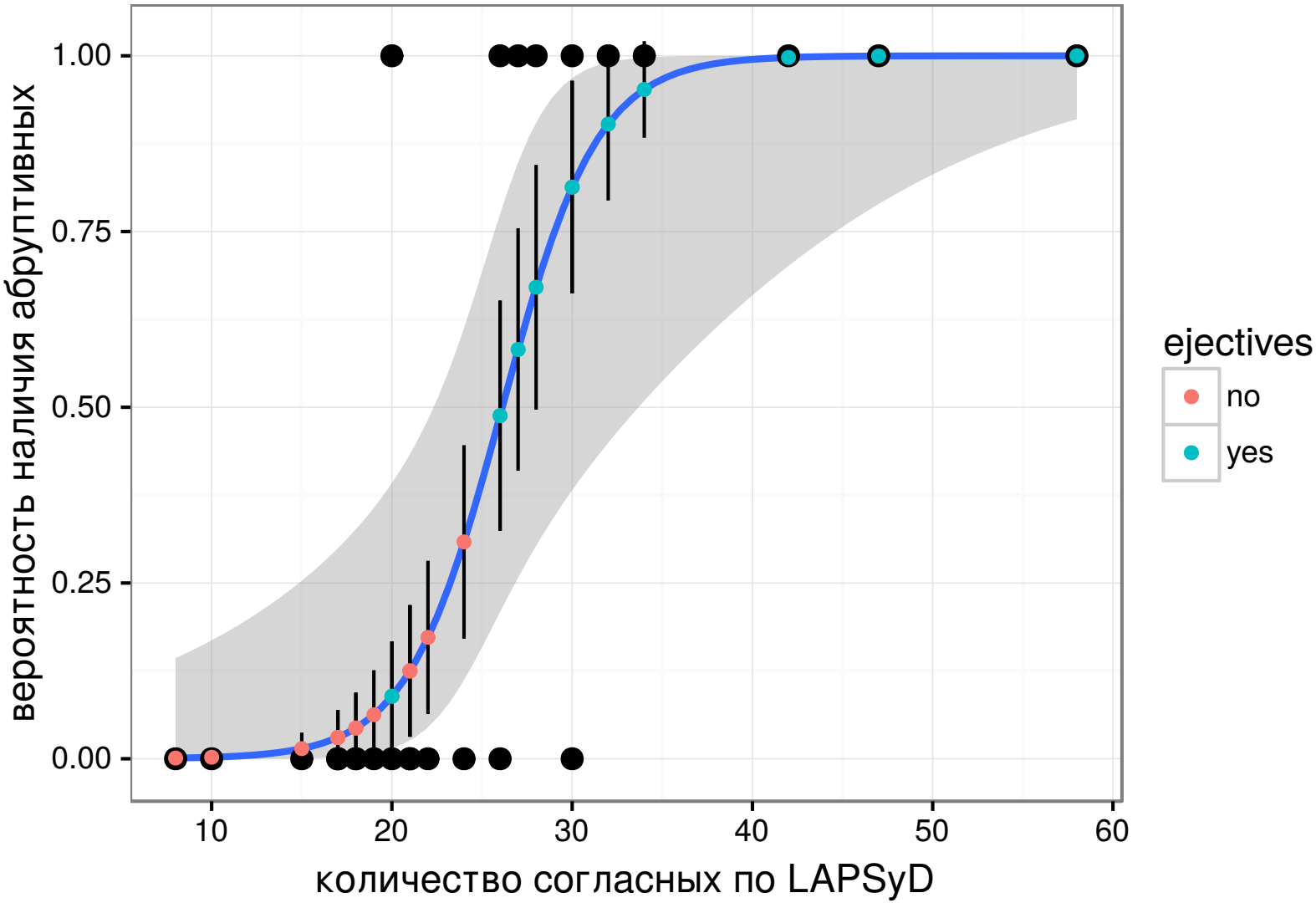
Доверительные интервалы для наблюдений

Можно построить доверительный интервал для каждого наблюдения:

```
df <- read.csv("http://goo.gl/0btfKa")
fit2 <- glm(ejectives ~n.cons.lapsyd, data = df, family = "binomial")

pred <- predict(fit2, type="response", se.fit = T)
df <- cbind.data.frame(df, fit = pred$fit, se.fit = pred$se.fit)
head(df)
```

	name	n.cons.lapsyd	ejectives	fit	se.fit
1	Turkish	24	no	0.30844363	0.13769764
2	Korean	21	no	0.12493187	0.09358363
3	Tiwi	21	no	0.12493187	0.09358363
4	Kpelle	22	no	0.17266963	0.10904915
5	Tulu	21	no	0.12493187	0.09358363
				↑	↑
				вероятности	довер. инт.

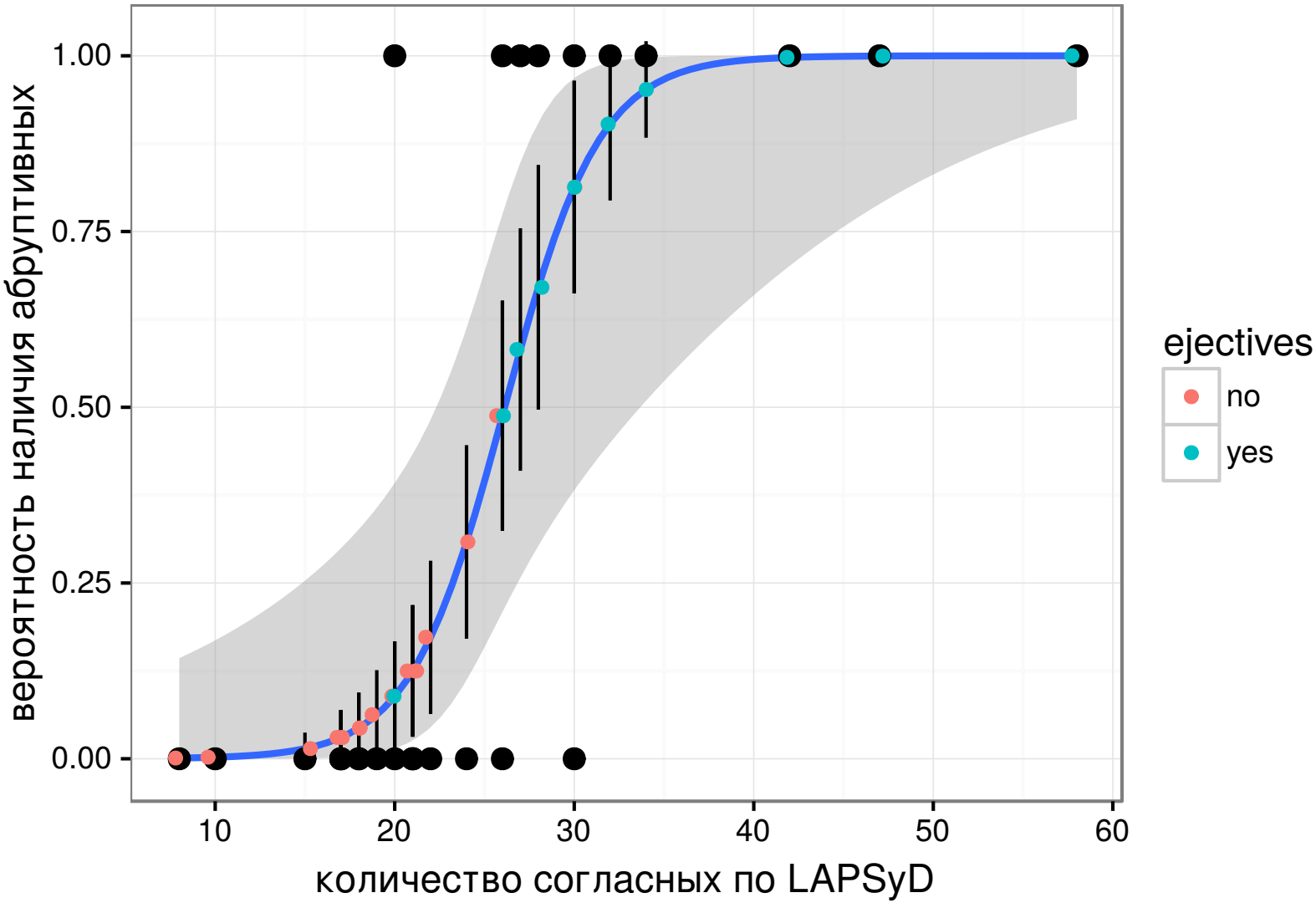
Визуализация: ggplot2



Визуализация: ggplot2

```
df <- read.csv("http://goo.gl/0btfKa")
fit2 <- glm(ejectives ~n.cons.lapsyd, data = df, family = "binomial")

pred <- predict(fit2, type="response", se.fit = T)          # вероятности и CI
df <- cbind.data.frame(df, fit = pred$fit, se.fit = pred$se.fit)

# Нужно переделать значения переменной ejectives в 0 и 1
df$ejectives.value <- as.numeric(df$ejectives) - 1

library(ggplot2)
ggplot(data = df, aes(x = n.cons.lapsyd, y = ejectives.value))+
                                                              # сигмоида
  geom_smooth(method = "glm", method.args = list(family = "binomial"))+
  geom_point() +                                              # наблюдения
  geom_pointrange(aes(x = n.cons.lapsyd,            # CI для вероятностей
                      ymin = fit - se.fit, ymax = fit + se.fit))+
                                                              # вероятности
  geom_point(aes(x = n.cons.lapsyd, y = fit, colour = ejectives))
```

Визуализация: ggplot2



Визуализация: ggplot2

```
df <- read.csv("http://goo.gl/0btfKa")
fit2 <- glm(ejectives ~n.cons.lapsyd, data = df, family = "binomial")

pred <- predict(fit2, type="response", se.fit = T)          # вероятности и CI
df <- cbind.data.frame(df, fit = pred$fit, se.fit = pred$se.fit)

# Нужно переделать значения переменной ejectives в 0 и 1
df$ejectives.value <- as.numeric(df$ejectives) - 1

library(ggplot2)
ggplot(data = df, aes(x = n.cons.lapsyd, y = ejectives.value))+
                                                              # сигмоида
  geom_smooth(method = "glm", method.args = list(family = "binomial"))+
  geom_point() +                                              # наблюдения
  geom_pointrange(aes(x = n.cons.lapsyd,            # CI для вероятностей
                      ymin = fit - se.fit, ymax = fit + se.fit))+
                                                              # вероятности
  geom_jitter(aes(x = n.cons.lapsyd, y = fit, colour = ejectives))
```

Задача 2

В работе [Coates, Leech 1980: 31] приводятся результаты исследования значений модальных глаголов (*must*, *have to*) в британском и американском английском. Авторы выделяют два значения в употреблении модальных глаголов буквальное (*you must read it*) и эпистемическое (*you must be kidding*). Данные основаны на работе Coates, J., Leech, G. (1980) The Meanings of the Modals in British and American English.

Для начала попробуем предсказать какой будет выбираться глагол на основе значения.

Модель с категориальным предиктором

```
df <- read.csv("http://goo.gl/4iEt4j")
fit3 <- glm(word ~meaning, data = df, family = "binomial")
summary(fit3)
```

```
Call:
glm(formula = word ~meaning, family = "binomial data = df)

Deviance Residuals:                                  # распределение остатков
    Min       1Q   Median       3Q      Max
-2.229   -1.028   -1.028    1.334    1.334

Coefficients:                                        # коэфициенты модели
             Estimate   Std. Error   z value   Pr(>|z|)
(Intercept)   2.3979       0.3148     7.618   2.59e-14   ***   # β0
meaning      -2.7595       0.3236    -8.529    < 2e-16   ***   # β1

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1205.5 on 869 degrees of freedom
Residual deviance: 1075.1 on 868 degrees of freedom
AIC: 1079.1                                          # критерий Акаике

Number of Fisher Scoring iterations: 4
```

Как были получены эти значения?

```
table(df$meaning, df$word)          # построим матрицу сопряженности
                epistemic    root
  have to              11     435
     must             121     303
```

```
fit3 <- glm(word ~meaning, data = df, family = "binomial")
fit3$coefficients
  (Intercept)    meaningroot
    2.397895       -2.759508
```

$$\log(odds) = 2.397895 + (-2.759508) \times \text{meaningroot}$$

В интерсепте логарифм шансов случаев с эпистемической модальностью

$$\log\left(\frac{121}{11}\right) = 2.397895$$

Второй коэффициент в сумме с интерсептом составляют логарифм шансов случаев с прямым значением

$$\log\left(\frac{303}{435}\right) = -0.3616132 = 2.397895 + (-2.759508)$$

Доверительный интервал для коэффициентов

Для коэффициентов модели можно посчитать доверительный интервал:

```
df <- read.csv("http://goo.gl/4iEt4j")
fit3 <- glm(word ~ meaning, data = df, family = "binomial")

confint(fit3)
Waiting for profiling to be done...
                2.5 %     97.5 %
(Intercept)  1.828070   3.074627
meaningroot -3.450629  -2.169762
```

Aspects ~~of the Theory~~ of Syntax

- $y = \beta_0 + \beta_1 \cdot x_1 + \varepsilon_i$ обычная формула
  `y~x`
- $y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \varepsilon_i$ обычная формула
  `y~x + z`
- $y = \beta_0 + \beta_1 \cdot x_2 \cdot x_1 + \varepsilon_i$ только взаимодействие
  `y~x:z`
- $y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \beta_3 \cdot x_2 \cdot x_1 + \varepsilon_i$ с взаимодействием
  `y~x*z`
- $y = \beta_0 + \varepsilon_i$ формула без предикторов
  `y~1`
- $y = \beta_1 \cdot x_1 + \varepsilon_i$ формула без свободного члена
  `y~x - 1`
- $y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_k \cdot x_k + \varepsilon_i$ все предикторы
  `y~.`

Спасибо за внимание

Пишите письма

agricolamz@gmail.com

презентация доступна: http://goo.gl/ZNJoGj

Список литературы

Cedergren, Henrietta Cecilia Jonas (1973). The interplay of social and linguistic factors in Panama. Cornell University.

Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society. Series B (Methodological), 215--242.