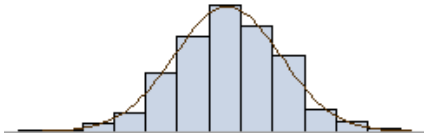
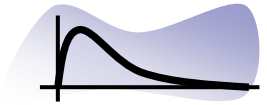
# Лекция 9:
# Обобщенные линейные модели, логистическая регрессия

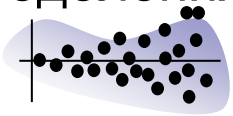
# Важное предположение линейной регрессии

- Нормальное распределение ошибки с константной дисперсией:



- Часто возникающие «особенности»:
  - Несимметричные распределения отклика
  - Гетероскедастичность
  - Ограниченная область определения отклика






- Что делать?
  - Явно преобразовывать отклик: $\mathbf{E}(\mathbf{g}(\mathbf{y}) \mid \mathbf{x})$

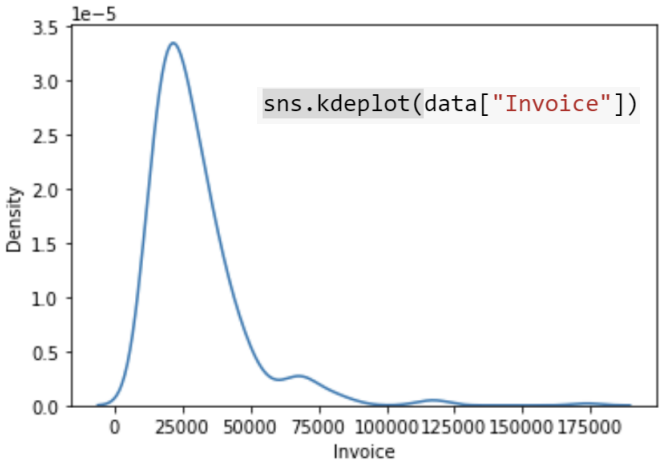
НО, в общем случае: $\mathbf{g}^{-1}(\mathbf{E}(\mathbf{g}(\mathbf{y}) \mid \mathbf{x})) \neq \mathbf{E}(\mathbf{y} \mid \mathbf{x})$

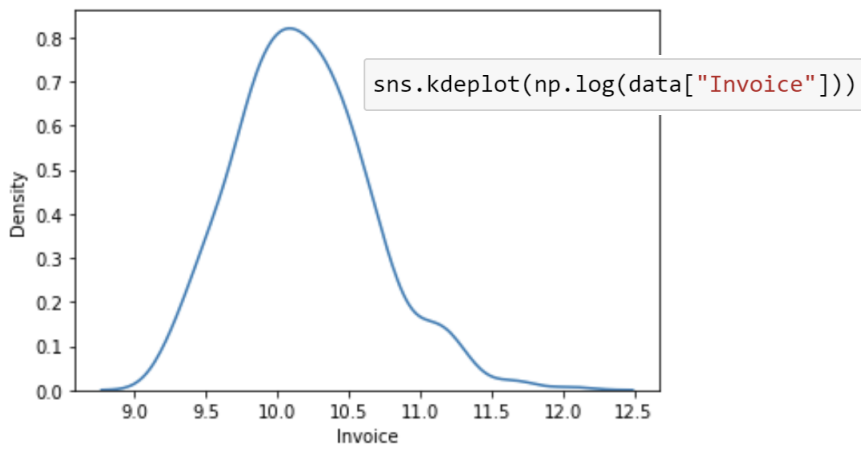
  - Использовать функцию связи: $\mathbf{g}(\mathbf{E}(\mathbf{y} \mid \mathbf{x}))$

# Пример

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data=pd.read_csv("cars0.csv",delimiter=",")
data.head()
```

| | Make | Model | Type | Origin | DriveTrain | MSRP | Invoice | EngineSize | Cylinders | Horsepower | MPG_City | MPG_Highway | Weight | Wheelbase | Length |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Acura | MDX | SUV | Asia | All | 36945.0 | 33337.0 | 3.5 | 6.0 | 265 | 17 | 23 | 4451 | 106 | 189 |
| **1** | Acura | RSX Type S 2dr | Sedan | Asia | Front | 23820.0 | 21761.0 | 2.0 | 4.0 | 200 | 24 | 31 | 2778 | 101 | 172 |
| **2** | Acura | TSX 4dr | Sedan | Asia | Front | 26990.0 | 24647.0 | 2.4 | 4.0 | 200 | 22 | 29 | 3230 | 105 | 183 |
| **3** | Acura | TL 4dr | Sedan | Asia | Front | 33195.0 | 30299.0 | 3.2 | 6.0 | 270 | 20 | 28 | 3575 | 108 | 186 |
| **4** | Acura | 3.5 RL 4dr | Sedan | Asia | Front | 43755.0 | 39014.0 | 3.5 | 6.0 | 225 | 18 | 24 | 3880 | 115 | 197 |



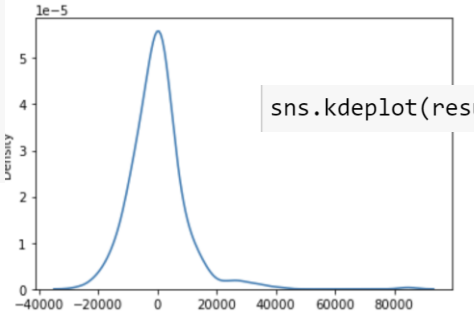



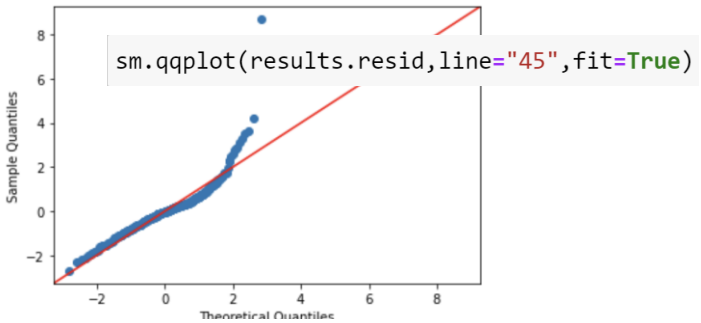
# Пример (МНК) – все плохо

```
import statsmodels.api as sm

ols = sm.OLS(endog=data['Invoice'],
             exog=sm.add_constant(data[['Weight','Length','Horsepower']]), data=data)
results=ols.fit()
results.summary()
```

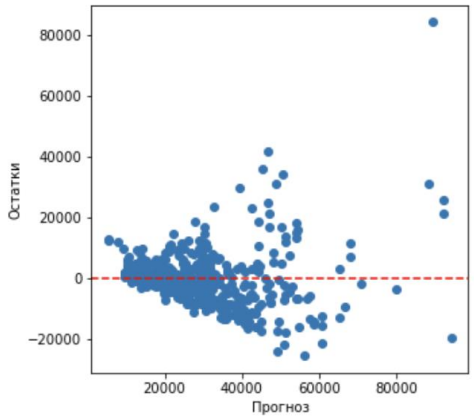
| | | | |
|---|---|---|---|
| **Dep. Variable:** | Invoice | **R-squared:** | 0.704 |
| **Model:** | OLS | **Adj. R-squared:** | 0.702 |
| **Method:** | Least Squares | **F-statistic:** | 336.3 |
| **Date:** | Wed, 01 Nov 2023 | **Prob (F-statistic):** | 1.06e-111 |
| **Time:** | 01:43:01 | **Log-Likelihood:** | -4531.2 |
| **No. Observations:** | 428 | **AIC:** | 9070. |
| **Df Residuals:** | 424 | **BIC:** | 9087. |
| **Df Model:** | 3 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 2.25e+04 | 6682.648 | 3.367 | 0.001 | 9362.779 | 3.56e+04 |
| **Weight** | 0.0255 | 1.015 | 0.025 | 0.980 | -1.970 | 2.021 |
| **Length** | -213.1397 | 45.054 | -4.731 | 0.000 | -301.696 | -124.583 |
| **Horsepower** | 218.3874 | 8.400 | 26.000 | 0.000 | 201.877 | 234.898 |



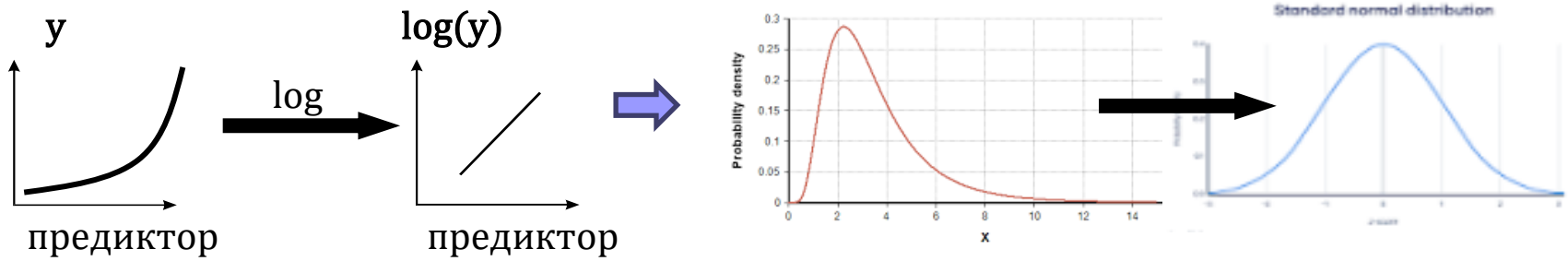




```
fig, ax = plt.subplots(figsize=(5, 5))
ax.scatter(results.predict(sm.add_constant(data[['Weight','Length','Horsepower']])),
           results.resid)
ax.set_ylabel('Остатки')
ax.set_xlabel('Прогноз')
plt.axhline(y = 0, color = 'r', linestyle = '--')
```



# Преобразование отклика и логнормальная регрессия

- Распределение отклика $y$ логнормальное, тогда распределение с.в. $\log(y)$ – нормальное: $\log(y) \sim N(\mu, \sigma^2)$




- Связь моментов исходной с.в. $y$ и $\log(y)$:

$$E(y) = \exp(\mu + \frac{\sigma^2}{2}),\ D(y) = \left(e^{\sigma^2} - 1\right)(E(y))^2$$

- Это значит, что можно построить МНК регрессию для прогнозирования $\log(y) = w^T x$ и получить исходный отклик:

$$\mu = E(\log(y)|x) = w^T x \Rightarrow E(y|x) = \exp\left(w^T x + \frac{\sigma^2}{2}\right)$$

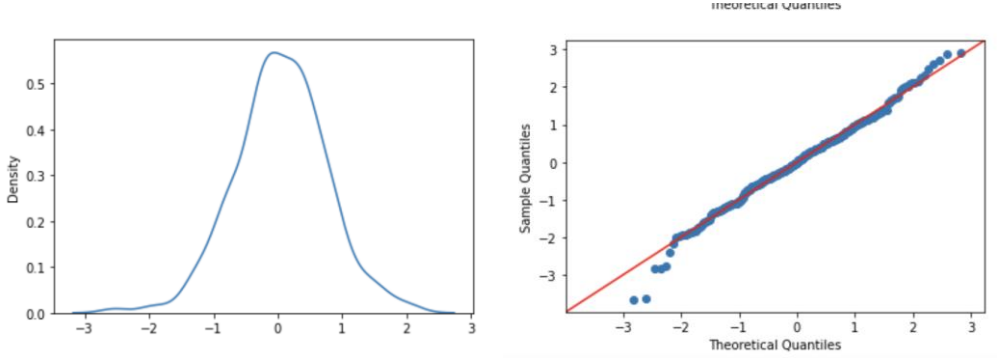
- Откуда брать $\sigma^2$? Можно взять оценку $\sigma^2 \approx MSE_{val}$, желательно на *валидационном* наборе

# Пример (логнормальная регрессия) – лучше

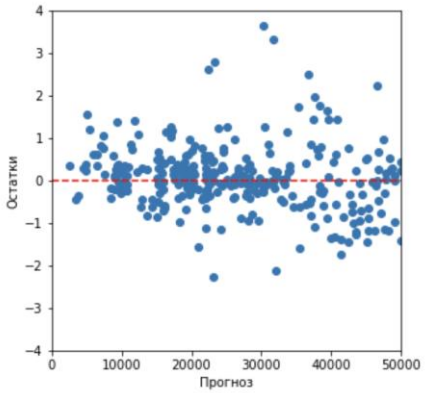
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

X_train, X_test, y_train, y_test = train_test_split(
    sm.add_constant(data[['Weight','Length','Horsepower']]),
    np.log(data['Invoice']),test_size=0.3)
lnr = sm.OLS(endog=y_train, exog=X_train)
lnr_results=lnr.fit()
mse=mean_squared_error(y_test, lnr_results.predict(X_test))
lnr_results.summary()
```

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Invoice | **R-squared:** | 0.768 |
| **Model:** | OLS | **Adj. R-squared:** | 0.766 |
| **Method:** | Least Squares | **F-statistic:** | 325.9 |
| **Date:** | Wed, 01 Nov 2023 | **Prob (F-statistic):** | 2.68e-93 |
| **Time:** | 01:54:31 | **Log-Likelihood:** | 9.5115 |
| **No. Observations:** | 299 | **AIC:** | -11.02 |
| **Df Residuals:** | 295 | **BIC:** | 3.779 |
| **Df Model:** | 3 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 9.6851 | 0.209 | 46.330 | 0.000 | 9.274 | 10.097 |
| **Weight** | 0.0001 | 2.83e-05 | 4.264 | 0.000 | 6.5e-05 | 0.000 |
| **Length** | -0.0060 | 0.001 | -4.324 | 0.000 | -0.009 | -0.003 |
| **Horsepower** | 0.0055 | 0.000 | 22.407 | 0.000 | 0.005 | 0.006 |




```
fig, ax = plt.subplots(figsize=(5, 5))
ax.scatter(np.exp(mse/2+lnr_results.predict(
    sm.add_constant(data[['Weight','Length','Horsepower']]))),
    results.resid_pearson)
plt.xlim(0, 50000)
plt.ylim(-4, 4)
ax.set_ylabel('Остатки')
ax.set_xlabel('Прогноз')
plt.axhline(y = 0, color = 'r', linestyle = '--')
```



# Обобщенная линейная модель

**Функция связи** → $g\big(\boldsymbol{E}(\boldsymbol{y}|\boldsymbol{x})\big) = w_0 + w_1 x_1 \ldots + w_k x_p = \langle x, w \rangle$

- Распределение отклика принадлежит экспоненциальному семейству $y_i \sim \mathrm{Exp}(\theta, \phi)$, где плотность определена как:

$$p(y|\theta,\phi) = exp\left(\frac{y\theta - c(\theta)}{\phi} + h(y,\phi)\right)$$

- Математическое ожидание с.в. $y$ зависит только от $\theta$ через некоторую монотонную *функцию связи* $g(.)$ (link function) как: $\mu = E(y) = \mathrm{c}'(\theta) \Rightarrow \theta = g(\mu) = [c']^{-1}(\mu)$
- Дисперсия с.в. $y$ есть функция от среднего: $D(y) = \phi c''(\theta)$
- Распределение отклика наблюдений может подсказать какую функцию связи и функцию потерь следует выбрать

# Важные частные случаи

- Линейная регрессия: $p(y|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{(y-\mu)^2}{2\sigma^2}\right)$
- Логистическая регрессия: $p(y|\mu) = \mu^y(1-\mu)^{1-y}$
- Пуассоновская регрессия: $p(y|\lambda) = \frac{e^{-\lambda}\lambda^y}{y!}$
- Гамма регрессия: $p(y|\nu,\mu) = \frac{1}{\Gamma(\nu)y}\left(\frac{y\nu}{\mu}\right)^{\nu} e^{-\frac{y\nu}{\mu}}$

| Регрессия | Отклик | Параметр $\theta$ (среднее) | Параметр $\phi$ | Дисперсия | Каноническая функция связи |
|---|---|---|---|---|---|
| Линейная | непрерывный неограниченный | μ | σ | $\sigma^2$ | тождество $g(\mu)= \mu$ |
| Логистическая | бинарный категориальный | μ | 1 | (1- μ) μ | логит $g(\mu)= \log(\mu/(1- \mu))$ |
| Пуассоновская | «Счетчик» - дискретный положительный | λ | 1 | λ | логарифм $g(\mu)= \log(\mu)$ |
| Гамма | непрерывный положительный | μ | ν | $\mu/ \nu^2$ | обратная $g(\mu)= 1/\mu$ |

# Примеры вывода функции связи

- Суть: приведение распределения к каноническому виду $\mathrm{Exp}(\theta, \phi)$
- Линейная регрессия (нормальное распределение):

$$p(y|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{(y-\mu)^2}{2\sigma^2}\right) = \exp\left(\frac{y\mu - \frac{1}{2}\mu^2}{\sigma^2} - \frac{y^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)\right)$$

$$\theta = g(\mu) = \mu,\ c(\theta) = \tfrac{1}{2}\mu^2 = \tfrac{1}{2}\theta^2$$

- Пуассоновская регрессия (распределение Пуассона):

$$p(y|\lambda) = \frac{e^{-\lambda}\lambda^y}{y!} = \exp\left(\frac{y\log(\lambda) - \lambda}{1} - \log(y)!\right)$$

$$\theta = g(\lambda) = \log(\lambda),\ c(\theta) = \lambda = e^{\theta}$$

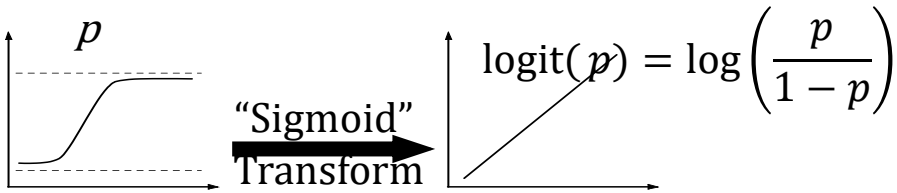
- Логистическая регрессия (распределение Бернулли):

$$p(y|\mu) = \mu^y(1-\mu)^{1-y} = \exp\left(y\log\left(\frac{\mu}{1-\mu}\right) - \log(1-\mu)\right)$$

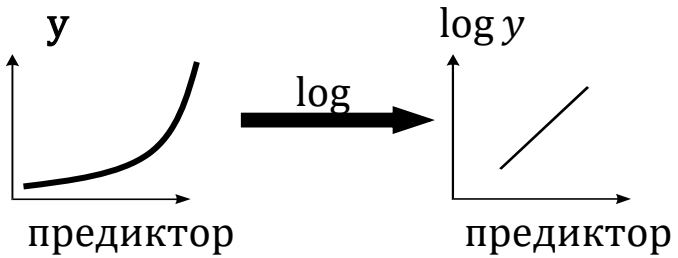
$$\theta = g(\mu) = \log\left(\tfrac{\mu}{1-\mu}\right),\ c(\theta) = -\log(1-\mu) = \log(1+e^{\theta})$$

# Не все так однозначно

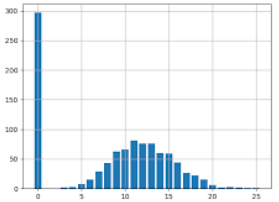
- На практике часто используют неканонические функции связи
- Например, для логистической регрессии:
  - Каноническая logit
  - probit: $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\mu} z^2 dz$
  - log-log: $\log(-\log(1-\mu))$




- Для гамма регрессии:
  - Каноническая обратная
  - log, тождественная и др.




- Для «счетчиков»:
  - Чрезмерная дисперсия - может не выполнятся условие $E(y) = D(y) = \lambda$ и тогда используют отрицательно биномиальное распределение, где дисперсия моделируется как функция от среднего и квадрата среднего
  - Может быть «смесь» счетчиков
  - “zero inflated” – смесь 0 и пуассоновского счетчика



# Пример гамма регрессии

```
gamma_model = sm.GLM(data['Invoice'],
    sm.add_constant(data[['Weight','Length','Horsepower']]),
    family=sm.families.Gamma())
gamma_results = gamma_model.fit()
gamma_results.summary()
```

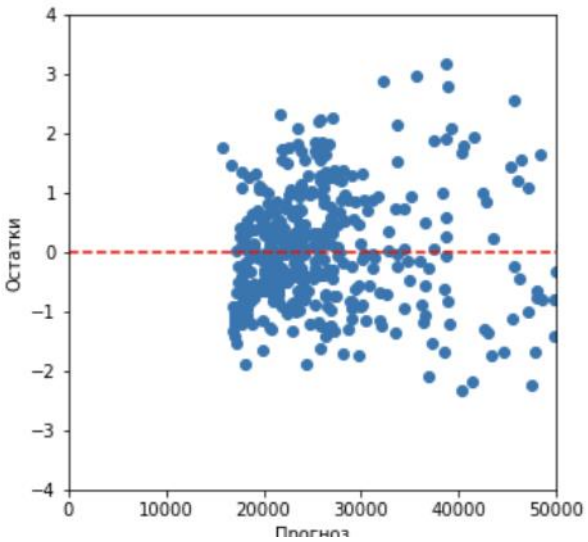
| | | | |
|---|---|---|---|
| **Dep. Variable:** | Invoice | **No. Observations:** | 428 |
| **Model:** | GLM | **Df Residuals:** | 424 |
| **Model Family:** | Gamma | **Df Model:** | 3 |
| **Link Function:** | inverse_power | **Scale:** | 0.11306 |
| **Method:** | IRLS | **Log-Likelihood:** | -5686.6 |
| **Date:** | Wed, 01 Nov 2023 | **Deviance:** | 310.53 |
| **Time:** | 02:03:07 | **Pearson chi2:** | 47.9 |
| **No. Iterations:** | 8 | **Pseudo R-squ. (CS):** | -74.85 |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 4.818e-05 | 6.76e-06 | 7.124 | 0.000 | 3.49e-05 | 6.14e-05 |
| **Weight** | -6.088e-09 | 5.99e-10 | -10.164 | 0.000 | -7.26e-09 | -4.91e-09 |
| **Length** | 2.391e-07 | 4.43e-08 | 5.402 | 0.000 | 1.52e-07 | 3.26e-07 |
| **Horsepower** | -1.467e-07 | 2.88e-09 | -50.999 | 0.000 | -1.52e-07 | -1.41e-07 |

Как считать?
Ответ позже

Статистика Уальда
(аналогично
Стьюденту для МНК)

```
fig, ax = plt.subplots(figsize=(5, 5))
ax.scatter(gamma_results.predict(data[['Weight','Length','Horsepower']]),
           results.resid_pearson)
plt.xlim(0, 50000)
plt.ylim(-4, 4)
ax.set_ylabel('Остатки')
ax.set_xlabel('Прогноз')
plt.axhline(y = 0, color = 'r', linestyle = '--')
```



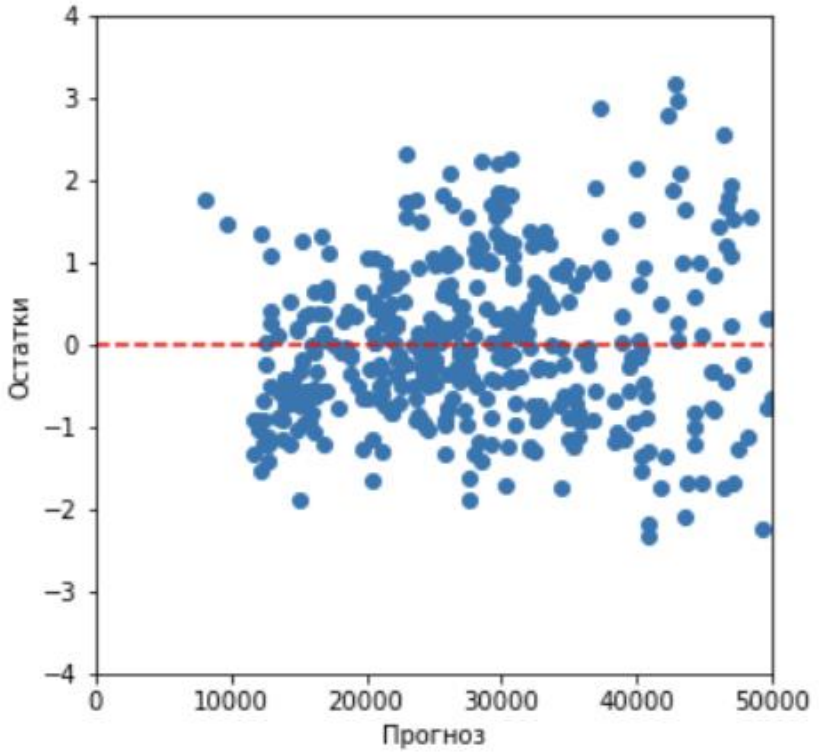

гетероскедастичность?

# Пример гамма регрессии с неканонической тождественной функцией связи

```
gamma_model = sm.GLM(data['Invoice'],
    sm.add_constant(data[['Weight','Length','Horsepower']]),
    family=sm.families.Gamma(sm.families.links.identity()))
gamma_results = gamma_model.fit()
gamma_results.summary()
```

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Invoice | **No. Observations:** | 428 |
| **Model:** | GLM | **Df Residuals:** | 424 |
| **Model Family:** | Gamma | **Df Model:** | 3 |
| **Link Function:** | identity | **Scale:** | 0.066351 |
| **Method:** | IRLS | **Log-Likelihood:** | -4359.6 |
| **Date:** | Wed, 01 Nov 2023 | **Deviance:** | 24.571 |
| **Time:** | 02:06:57 | **Pearson chi2:** | 28.1 |
| **No. Iterations:** | 19 | **Pseudo R-squ. (CS):** | 0.9438 |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 2.359e+04 | 4377.250 | 5.389 | 0.000 | 1.5e+04 | 3.22e+04 |
| **Weight** | 2.8085 | 0.849 | 3.307 | 0.001 | 1.144 | 4.473 |
| **Length** | -209.3271 | 31.087 | -6.734 | 0.000 | -270.256 | -148.398 |
| **Horsepower** | 161.8258 | 8.531 | 18.969 | 0.000 | 145.105 | 178.547 |

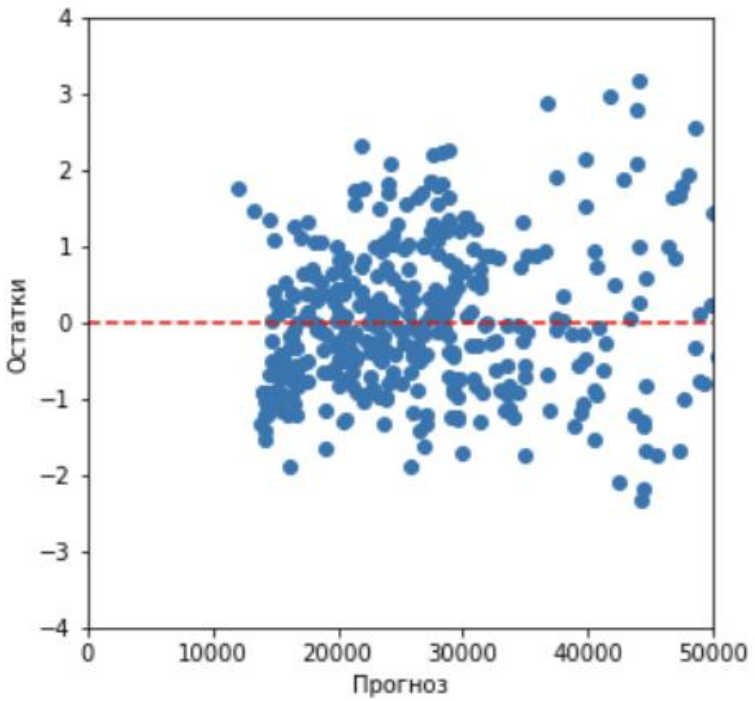


# Пример гамма регрессии с неканонической функцией связи log

```
gamma_model = sm.GLM(data['Invoice'],
    sm.add_constant(data[['Weight','Length','Horsepower']]),
    family=sm.families.Gamma(sm.families.links.Log()))
gamma_results = gamma_model.fit()
gamma_results.summary()
```

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Invoice | **No. Observations:** | 428 |
| **Model:** | GLM | **Df Residuals:** | 424 |
| **Model Family:** | Gamma | **Df Model:** | 3 |
| **Link Function:** | Log | **Scale:** | 0.059580 |
| **Method:** | IRLS | **Log-Likelihood:** | -4346.8 |
| **Date:** | Wed, 01 Nov 2023 | **Deviance:** | 23.319 |
| **Time:** | 02:10:09 | **Pearson chi2:** | 25.3 |
| **No. Iterations:** | 12 | **Pseudo R-squ. (CS):** | 0.9614 |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 9.6571 | 0.169 | 57.017 | 0.000 | 9.325 | 9.989 |
| **Weight** | 0.0001 | 2.57e-05 | 4.863 | 0.000 | 7.47e-05 | 0.000 |
| **Length** | -0.0058 | 0.001 | -5.077 | 0.000 | -0.008 | -0.004 |
| **Horsepower** | 0.0055 | 0.000 | 25.850 | 0.000 | 0.005 | 0.006 |



# Максимизация правдоподобия для GLM методом Ньютона-Рафсона

- Принцип максимума правдоподобия:

$$L(w) = -\log \prod_{i=1}^{l} p(y_i|\theta_i, \phi_i) = -\sum_{i=1}^{l} [y_i\theta_i - c(\theta_i)]/\phi_i \to \min_w \quad ,$$

где $\theta_i = w^T x_i$

- Метод Ньютона-Рафсона ($t$– номер итерации):

$$w^{t+1} = w^t - \eta_t \, (\nabla^2 L(w^t))^{-1} \nabla L(w^t)$$

- Градиент $\nabla L\left(w^t\right)$ :

$$\frac{\partial L(w)}{\partial w_j} = \sum_{i=1}^{l} \frac{y_i - c'(w^T x_i)}{\phi_i} x_i$$

- Матрица Гессе $\nabla^2 L(w^t)$:

$$\frac{\partial^2 L(w)}{\partial w_j \partial w_k} = -\sum_{i=1}^{l} \frac{c''(w^T x_i)}{\phi_i} x_i x_k$$

# Метод IRLS (Iteratively reweighted least squares)

- Обозначения:
  - Взвешенная (по наблюдениям) матрица признаков $\tilde{X} = W_t X$,
  - где $X$ исходная матрица данных,
  - $W_t = diag\left(\sqrt{\frac{c''(\theta_i)}{\phi_i}}\right)$ – веса наблюдений на $t$-ой итерации
  - $\tilde{y_i} = \frac{y_i - c'(\theta_i)}{\sqrt{\phi_i c''(\theta_i)}}$ – модифицированные отклики
- Метод Ньютона-Рафсона принимает вид:

$$w^{t+1} = w^t - \eta_t \underbrace{\left(\mathrm{X^T W_t W_t X}\right)^{-1} \mathrm{X^T W_t}}_{\left(\tilde{X}^T \tilde{X}\right)^{-1} \tilde{X}^T} \left( \underbrace{\sqrt{\frac{\phi_i}{c''(\theta_i)}} \frac{y_i - c'(\theta_i)}{\phi_i}}_{\tilde{y}_i} \right)$$

- На каждом шаге - МНК линейной регрессии с взвешенными наблюдениями и модифицированными откликами:

$$\left\| \tilde{X} - \tilde{y} w \right\|^2 \to \min_w$$

# Особенности поиска решения

- При небольшой выборке IRLS – лучший вариант
- Но на больших выборках используют методы:
  - градиентные (в том числе стохастические)
  - квазиньютоновские (в том числе lbfgs)
- Есть варианты борьбы с переобучением:
  - $L_1$ и $L_2$ регуляризация
  - пошаговый отбор переменных (вместо тестов Фишера или Стьюдента – тест Уальда, информационные критерии и кросс-валидация работают как и для МНК)
- Для оценки важности переменных используются:
  - стандартные ошибки расчета коэффициентов (за рамками нашего курса)
  - статистика Уальда для оценки важности коэффициентов:

$$\frac{w_i}{SE(w_i)} \sim N(0,1)$$

# Пуассоновская регрессия

- Для моделирования количества наступлений события или доли (rate) наступлений события как функции от предикторов:

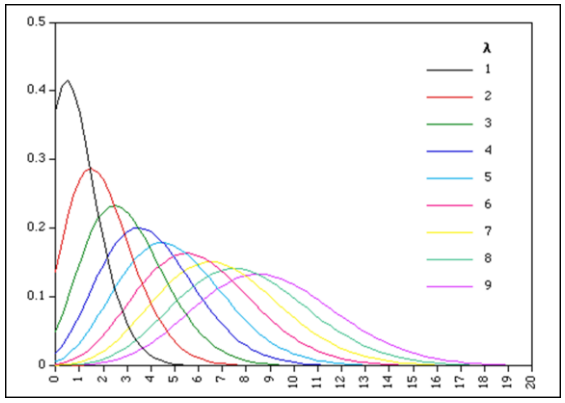
$$\log(E(y|x)) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_px_k \Rightarrow$$
$$\mu(w) = e^{w_0} \cdot e^{w_1x_1} \ldots e^{w_px_p}$$

- Положительный (и как правило дискретный) отклик
- **Функция связи**: log
- **Функция потерь**: $L(x, y, w) = y\log\left(\frac{y}{\mu(w)}\right) - (y - \mu(w))$
- Интерпретация построенной модели:
  - $e^w$ – мультипликативный эффект на отклик от изменения предиктора на единицу
  - Например, если $e^{w_1} = 1.2$, тогда увеличение $x_1$ на одну единицу вызывает 20% увеличение ожидаемого отклика, а если $e^{w_2} = 0.8$, тогда увеличение $x_2$ на одну единицу вызывает 20% уменьшение ожидаемого отклика

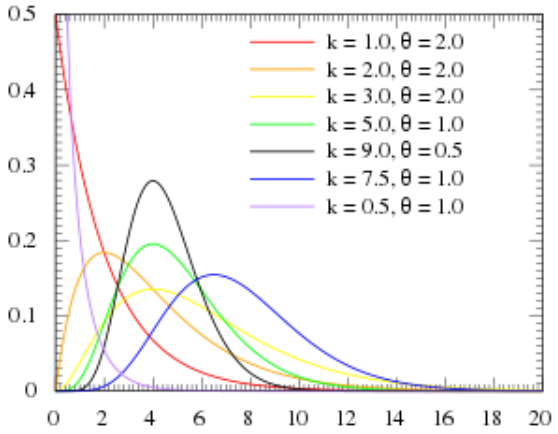
# Пуассоновская регрессия

- Пуассоновская регрессия наиболее подходит для *редких* событий
  - распределение отклика должно иметь маленькое среднее (<10 или даже <5, в идеале ~1)
  - иначе гамма и логнормальное распределение может быть лучше чем пуассоновское, если распределение сильно ассиметричное или есть чрезмерная дисперсия
  - или нормальное, если распределение достаточно симметричное

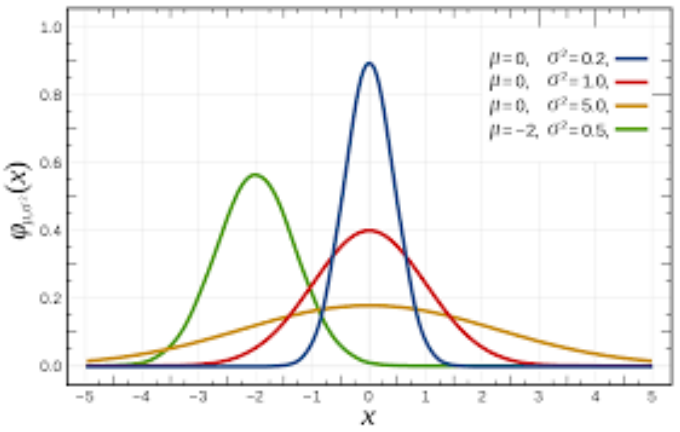
Пуассоновское




Гамма




Нормальное



# Пример пуассоновской регрессии

```
dt=pd.read_csv("ships.csv",delimiter=",")
dt.head()
```

| | type | age_period | operation_period | months | damages |
|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | 127 | 0 |
| **1** | 1 | 1 | 2 | 63 | 0 |
| **2** | 1 | 2 | 1 | 1095 | 3 |
| **3** | 1 | 2 | 2 | 1095 | 4 |
| **4** | 1 | 3 | 1 | 1512 | 6 |

```
X.head()
```

| | Intercept | C(type)[T.2] | C(type)[T.3] | C(type)[T.4] | C(type)[T.5] | months |
|---|---|---|---|---|---|---|
| **0** | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 127.0 |
| **1** | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 63.0 |
| **2** | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1095.0 |
| **3** | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1095.0 |
| **4** | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1512.0 |

```
from patsy import dmatrices
import statsmodels.api as sm
y, X = dmatrices("damages~C(type)+months", dt, return_type="dataframe")
pois_model = sm.GLM(y,X, family=sm.families.Poisson())
pois_results = pois_model.fit()
pois_results.summary()
```

Generalized Linear Model Regression Results

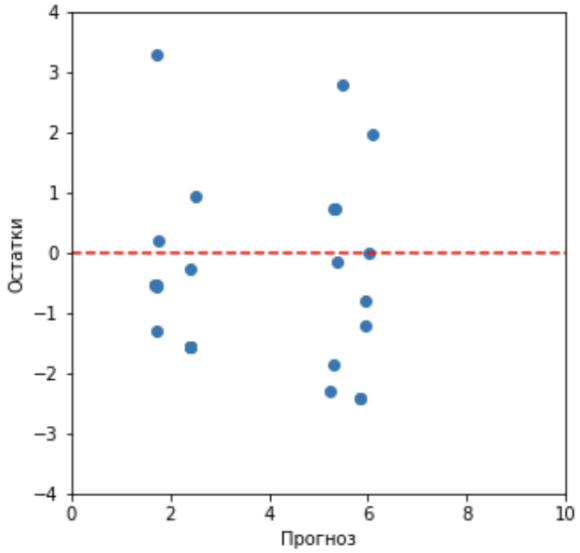
| | | | |
|---|---|---|---|
| **Dep. Variable:** | damages | **No. Observations:** | 34 |
| **Model:** | GLM | **Df Residuals:** | 28 |
| **Model Family:** | Poisson | **Df Model:** | 5 |
| **Link Function:** | Log | **Scale:** | 1.0000 |
| **Method:** | IRLS | **Log-Likelihood:** | -125.73 |
| **Date:** | Tue, 31 Oct 2023 | **Deviance:** | 153.59 |
| **Time:** | 02:55:48 | **Pearson chi2:** | 151. |
| **No. Iterations:** | 6 | **Pseudo R-squ. (CS):** | 1.000 |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **Intercept** | 1.7650 | 0.154 | 11.429 | 0.000 | 1.462 | 2.068 |
| **C(type)[T.2]** | 1.4035 | 0.194 | 7.219 | 0.000 | 1.022 | 1.785 |
| **C(type)[T.3]** | -1.2434 | 0.327 | -3.798 | 0.000 | -1.885 | -0.602 |
| **C(type)[T.4]** | -0.8902 | 0.287 | -3.097 | 0.002 | -1.454 | -0.327 |
| **C(type)[T.5]** | -0.1078 | 0.235 | -0.460 | 0.646 | -0.568 | 0.352 |
| **months** | 1.96e-05 | 4.61e-06 | 4.249 | 0.000 | 1.06e-05 | 2.86e-05 |

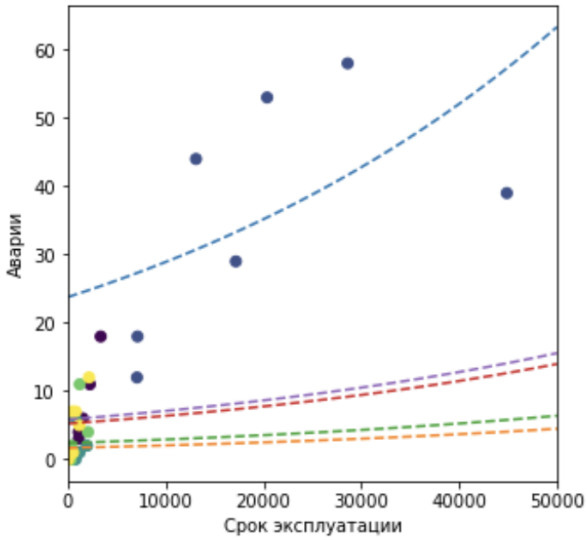
# Пример пуассоновской регрессии

```
fig, ax = plt.subplots(figsize=(5, 5))
ax.scatter(pois_results.predict(X),
           pois_results.resid_pearson)
plt.xlim(0, 10)
plt.ylim(-4, 4)
ax.set_ylabel('Остатки')
ax.set_xlabel('Прогноз')
plt.axhline(y = 0, color = 'r', linestyle = '--')
```

```
<matplotlib.lines.Line2D at 0x2e7d9070c70>
```

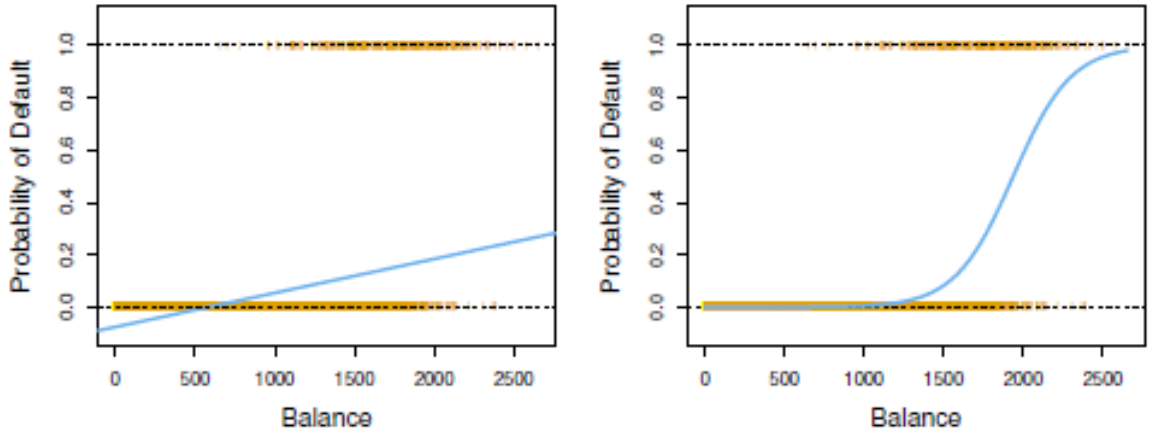



```
fig, ax = plt.subplots(figsize=(5, 5))
ax.scatter(dt['months'], dt["damages"],c=dt["type"])
m = 50000
p = 100
plt.xlim(0, m)
ax.set_ylabel('Аварии')
ax.set_xlabel('Срок эксплуатации')

for i in range(1,6):
    X=np.array([np.full(p,1), np.full(p,i==1),np.full(p,i==2),
                np.full(p,i==3),np.full(p,i==4),
                np.linspace(0,m,p)]).transpose()
    plt.plot(np.linspace(0,m,p),
         (pois_results.predict(X)), linestyle="--")
```



# Логистическая регрессия

- Почему нельзя моделировать вероятность как непрерывный отклик с помощью линейной регрессии?




- Как представить категориальный отклик в виде числовой переменной?
- Если отклик закодирован (1=Yes, 0=No), а прогноз 1.1 или -0.4, что это означает?
- Если переменная имеет только два значения (или несколько), имеет ли смысл требовать постоянство дисперсии или нормальность ошибок?
- Вероятность ограничена, а линейная функция нет. Принимая во внимание ограниченность вероятности, можно ли предполагать линейную связь между предиктором и откликом?

# Логистическая регрессия

**Уравнение** регрессии:

$$\text{logit}(p_i) = \mu = w_0 + w_1 x_{1i} + \cdots + w_p x_{pi}$$

Вероятность

параметр

предиктор

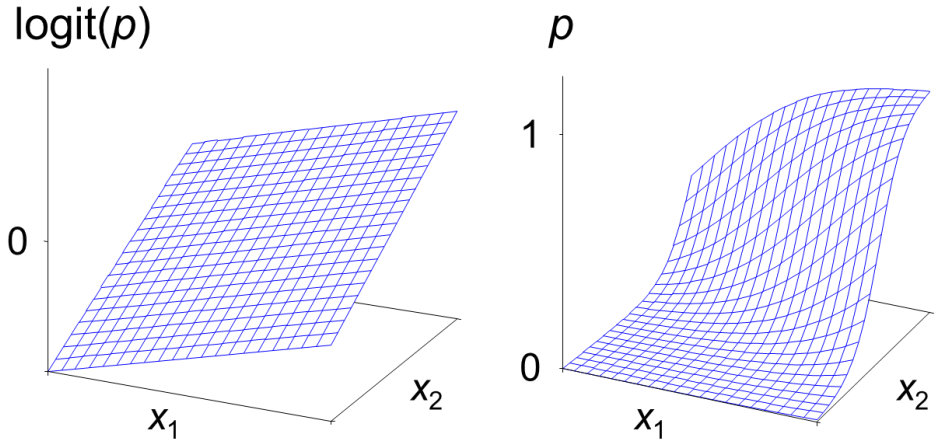
$$p_i = p(y = 1|x) = 1 - p(y = -1|x)$$

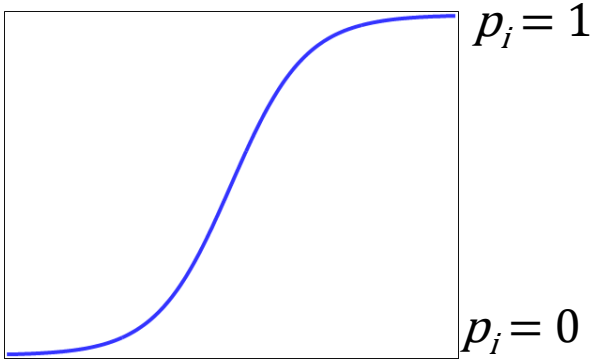
**Функция связи** (логит) и обратная ей (логистическая):

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right) = \mu \Rightarrow$$

$$\Rightarrow \quad p_i = \sigma(\mu) = \frac{1}{1+e^{-\mu}} = \frac{1}{1+e^{-x^T w}}$$

Основное предположение линейной логистической регрессии (линейная зависимость логита вероятности от предикторов):



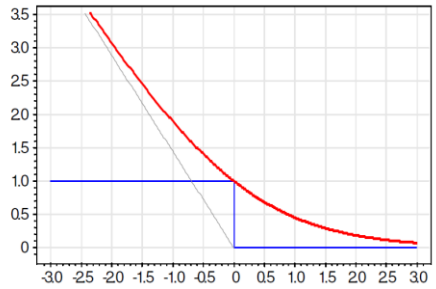

меньше $\leftarrow \mu \rightarrow$ больше

Ограничивает значение отклика

# Функция потерь логистической регрессии

- **Функция потерь** (логарифмическая) является аппроксимацией негладкой функции потерь sign(.):

$$L(y, x, w) = \log\left[1 + \exp\left(-yw^T x\right)\right] \geq$$
$$\geq \text{ sign}(yw^T x)$$




- Градиент $\nabla Q(w)$ и матрица Гессе $\nabla^2 Q(w)$ для метода Ньютона-Рафсона:

$$w^{t+1} = w^t - \eta_t \left(\nabla^2 Q(w^t)\right)^{-1} \nabla Q(w^t)$$

$$\frac{\partial Q(w)}{\partial w_j} = \sum_{i=1}^{l} (1 - \sigma_i) y_i x_i, \quad \frac{\partial^2 Q(w)}{\partial w_j \partial w_k} = -\sum_{i=1}^{l} (1 - \sigma_i) \sigma_i y_i x_i x_k$$

где $\sigma_i = \sigma\left(y_i w^T x_i\right)$, $\sigma(z) = \frac{1}{1+e^{-z}}$ - сигмоидальная функция

# IRLS для логистической регрессии

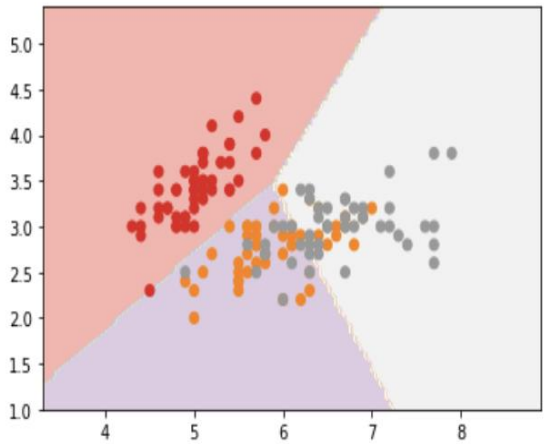
- На каждом шаге:
  - МНК линейной регрессии с взвешенными наблюдениями и модифицированными остатками, старающийся улучшить эмпирический риск на самых «сложных» примерах:

$$Q(w) = \sum_{i=1}^{l}(1-\sigma_i)\sigma_i\left(w^T x_i - \frac{y_i}{\sigma_i}\right)^2 \to \min_w \quad \leftrightarrow \quad \left\|\tilde{X} - \tilde{y}w\right\|^2 \to \min_w$$

- где:
  - Взвешенная (по наблюдениям) матрица признаков $\tilde{X} = W_t X$
  - $X$ исходная матрица данных,
  - $W_t = diag\big((1-\sigma_i)\sigma_i\big)$ – веса наблюдений на $t$-ой итерации,
  - поскольку $\sigma_i = P(y_i|x_i)$- вероятность правильной классификации $x_i$, то чем ближе $x_i$ к границе, тем больше вес $(1-\sigma_i)\sigma_i$ и «сложнее» пример
  - $\tilde{y_i} = \frac{y_i}{\sigma_i}$ – модифицированные отклики, чем выше вероятность ошибки тем больше $\frac{1}{\sigma_i}$

# Многоклассовая логистическая регрессия и функция softmax

```
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
from sklearn.inspection import DecisionBoundaryDisplay

iris = datasets.load_iris()
X = iris.data[:, :2]
Y = iris.target

logreg = LogisticRegression()
logreg.fit(X, Y)

DecisionBoundaryDisplay.from_estimator(
    logreg, X, cmap="Pastel1")
plt.scatter(X[:, 0], X[:, 1], c=Y, cmap="Set1")
plt.show()
```




- Логистическая регрессия с двумя классами обобщается на случай K классов (многомерная логистическая функция):

$$p(y = k|x) = \frac{e^{w_k^T x}}{\sum_{j=1}^{K} e^{w_j^T x}}$$

- Для *каждой* пары классов существует своя граница - линейная разделяющая функция, где вероятности классов совпадают
- Многоклассовая логистическая регрессия также называется *мультиномиальной регрессией, а* многомерная логистическая функция -softmax, *которая* «нормализует» K-мерный вектор так, чтобы сумма координат = 1

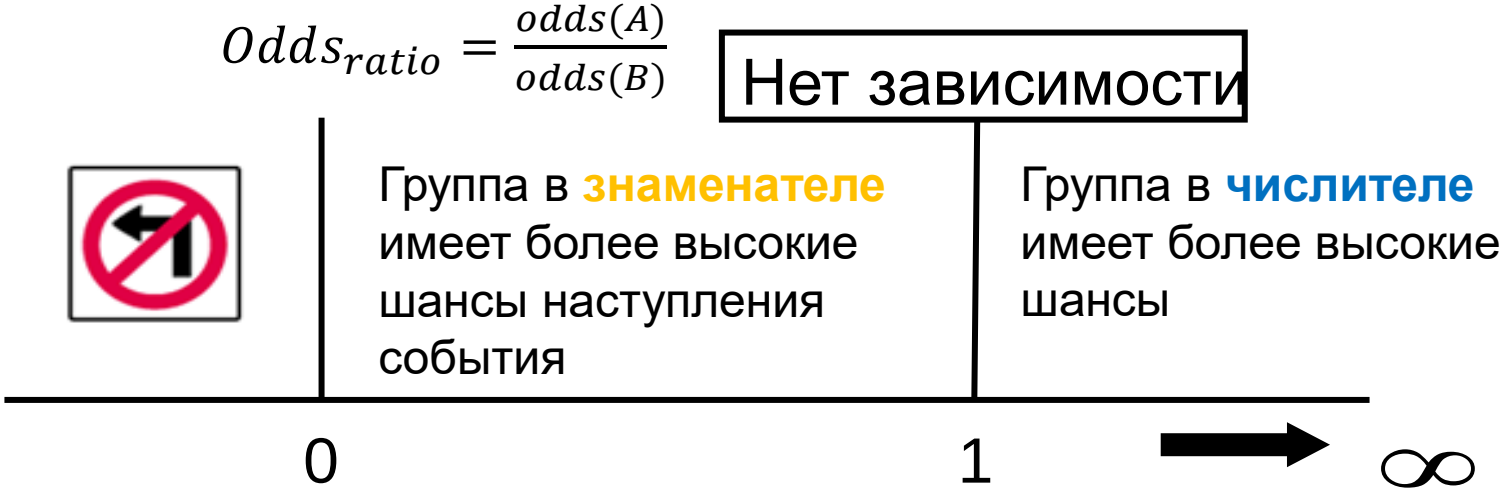
# Оценка «силы» ассоциации между предиктором и бинарным откликом

- **Шанс** (это не вероятность) – отношение вероятностей события к не событию:

$$Odds = \frac{p_{event}}{p_{nonevent}}$$

- **Отношение шансов** (тоже не вероятность) показывает насколько вероятнее в терминах шансов появления события в группе A (соответствующей набору значений предикторов) по сравнению с другой группой B:

$$Odds_{ratio} = \frac{odds(A)}{odds(B)}$$



# Сравнение вероятностей и шансов

| | Заболел | | |
|---|---|---|---|
| | **Да** | **Нет** | **Total** |
| **Прививка** | 60 | 20 | 80 |
| **Без прививки** | 90 | 10 | 100 |
| **Total** | 150 | 30 | 180 |

Всего **Заболел** Без прививки

÷

Всего исходов Без прививки

**Вероятность Заболел** Без прививки **=90÷100=0.9**

# Сравнение вероятностей и шансов

| | Заболел | | |
|---|---|---|---|
| | **Да** | **Нет** | **Total** |
| **Прививка** | 60 | 20 | 80 |
| **Без прививки** | 90 | 10 | 100 |
| **Total** | 150 | 30 | 180 |

Вероятность **Заболел** Без прививки =0.90

÷

Вероятность **Не заболел** Без прививки =0.10

Шанс **Заболеть** Без прививки = **0.90÷0.10=9**

**Без прививки шанс заболеть в 9 раз выше чем с прививкой**

# Сравнение вероятностей и шансов

| | Заболел | | |
|---|---|---|---|
| | **Да** | **Нет** | **Total** |
| **Прививка** | 60 | 20 | 80 |
| **Без прививки** | 90 | 10 | 100 |
| **Total** | 150 | 30 | 180 |

Шанс Заболеть **с прививкой**=**3**

÷

Шанс Заболеть **Без прививки**=**9**

Отношение шансов =**3**÷**9**=**0.3333**

**Шансов заболеть с прививкой в 3 раза меньше чем без**

# Отношение шансов в логистической регрессии

- Используется для оценки влияния переменной на отклик и показывает как изменятся шансы при изменении i-ой переменной на 1 (равно exp от коэффициента):

$$\text{logit}(p) = \log(odds) = w_0 + w_i x_i + \sum_{j \neq i} w_j x_j \Rightarrow$$

$$odds = \exp(w_0 + w_i x_i + \sum_{j \neq i} w_j x_j)$$

$$\text{logit}(p`) = \log(odds`) = w_0 + w_i(x_i + 1) + \sum_{j \neq i} w_j x_j \Rightarrow$$

$$odds` = \exp(w_0 + w_i(x_i + 1) + \sum_{j \neq i} w_j x_j)$$

$$odds_{ratio} = \frac{odds'}{odds} = \exp(w_i)$$

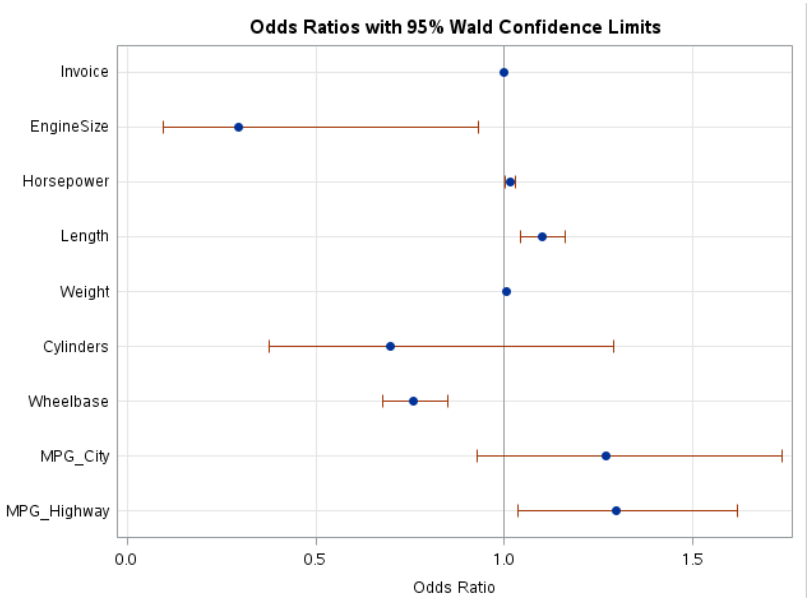
- Если больше 1 – шансы увеличиваются, если меньше, то уменьшаются, интерпретация как в пуассоновской регрессии

# Отношение шансов и важность переменных

| Odds Ratio Estimates | | | |
|---|---|---|---|
| Effect | Point Estimate | 95% Wald Confidence Limits | |
| Invoice | 1.000 | 1.000 | 1.000 |
| EngineSize | 0.295 | 0.094 | 0.931 |
| Horsepower | 1.016 | 1.003 | 1.029 |
| Length | 1.100 | 1.044 | 1.160 |
| Weight | 1.005 | 1.004 | 1.007 |
| Cylinders | 0.696 | 0.376 | 1.289 |
| Wheelbase | 0.757 | 0.676 | 0.849 |
| MPG_City | 1.270 | 0.929 | 1.736 |
| MPG_Highway | 1.295 | 1.036 | 1.618 |

exp(.)

| Analysis of Maximum Likelihood Estimates | | | | | |
|---|---|---|---|---|---|
| Parameter | DF | Estimate | Standard Error | Wald Chi-Square | Pr > ChiSq |
| Intercept | 1 | -10.7686 | 4.6784 | 5.2983 | 0.0213 |
| Invoice | 1 | -0.00013 | 0.000028 | 21.9445 | <.0001 |
| EngineSize | 1 | -1.2200 | 0.5858 | 4.3365 | 0.0373 |
| Horsepower | 1 | 0.0156 | 0.00666 | 5.4867 | 0.0192 |
| Length | 1 | 0.0957 | 0.0270 | 12.6146 | 0.0004 |
| Weight | 1 | 0.00529 | 0.000908 | 33.9767 | <.0001 |
| Cylinders | 1 | -0.3625 | 0.3146 | 1.3275 | 0.2493 |
| Wheelbase | 1 | -0.2778 | 0.0580 | 22.9685 | <.0001 |
| MPG_City | 1 | 0.2389 | 0.1595 | 2.2421 | 0.1343 |
| MPG_Highway | 1 | 0.2584 | 0.1136 | 5.1710 | 0.0230 |




- Можно найти не только точечную оценку ОШ (OR), но и доверительный интервал
- Если он содержит 1, то доверительный интервал коэффициента содержит 0, т.е. предиктор не значимый
- Не учитывается разброс переменной

# Категориальные предикторы

- Схемы кодировки:
  - Effect coding (относительно «среднего»)

| Переменная | Значение | Обозначение | 1 | 2 |
|---|---|---|---|---|
| **IncLevel** | 1 | Low Income | 1 | 0 |
| | 2 | Medium Income | 0 | 1 |
| | 3 | High Income | -1 | -1 |

  - Reference coding (относительно «базового»)

| Переменная | Значение | Обозначение | 1 | 2 |
|---|---|---|---|---|
| **IncLevel** | 1 | Low Income | 1 | 0 |
| | 2 | Medium Income | 0 | 1 |
| | 3 | High Income | 0 | 0 |

# Effect coding: Пример

$$\text{logit}(p) = w_0 + w_1 * D_{\text{Low income}} + w_2 * D_{\text{Medium income}}$$

$w_0$= Общий логарифм от шанса по всем категориям

$w_1$= разница между логарифмом шанса Low income и $w_0$

$w_2$ = разница между логарифмом шанса Medium income и общим

| Analysis of Maximum Likelihood Estimates | | | | | | |
|---|---|---|---|---|---|---|
| **Parameter** | | **DF** | **Estimate** | **Standard Error** | **Wald Chi-Square** | **Pr > ChiSq** |
| **Intercept** | | 1 | -0.5363 | 0.1015 | 27.9143 | <.0001 |
| **IncLevel** | **1** | 1 | -0.2259 | 0.1481 | 2.3247 | 0.1273 |
| **IncLevel** | **2** | 1 | -0.2200 | 0.1447 | 2.3111 | 0.1285 |

# Reference coding: Пример

$$\text{logit}(p) = w_0 + w_1 * D_{\text{Low income}} + w_2 * D_{\text{Medium income}}$$

$w_0$ = Логарифм шанса для High

$w_1$ = Разница между логарифмами шанса Low и High

$w_2$ = Разница между логарифмами шанса между Medium и High

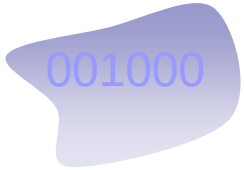
| Analysis of Maximum Likelihood Estimates | | | | | | |
|---|---|---|---|---|---|---|
| **Parameter** | | **DF** | **Estimate** | **Standard Error** | **Wald Chi-Square** | **Pr > ChiSq** |
| **Intercept** | | 1 | -0.0904 | 0.1608 | 0.3159 | 0.5741 |
| **IncLevel** | **1** | 1 | -0.6717 | 0.2465 | 7.4242 | 0.0064 |
| **IncLevel** | **2** | 1 | -0.6659 | 0.2404 | 7.6722 | 0.0056 |

# Категориальные предикторы

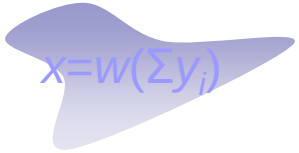
| ***Пример переменной*** | Мощность | ***Подход*** |
|---|---|---|
| Physical characteristics<br>Region<br>Partnership status | 10 | Бинарное кодирование |
| Education level<br>Urbanicity codes | 100 | **Преобразования или отображение на числовую шкалу** |
| State<br>Ethnicity<br>Employment classification | 1000 | |
| Postal Code<br>Address | 1000000 | Связывание |
| Social security number | 100000000 | |
| text | *Бесконечность* | Текстовые модели |

# Подходы к кодировке категориальных признаков


Случайное кодирование




Бинарное кодирование




Преобразование
с учётом отклика

**Unsupervised:**

- Backward Difference Contrast [2][3]
- BaseN [6]
- Binary [5]
- Gray [14]
- Count [10]
- Hashing [1]
- Helmert Contrast [2][3]
- Ordinal [2][3]
- One-Hot [2][3]
- Rank Hot [15]
- Polynomial Contrast [2][3]
- Sum Contrast [2][3]

**Supervised:**

- CatBoost [11]
- Generalized Linear Mixed Model [12]
- James-Stein Estimator [9]
- LeaveOneOut [4]
- M-estimator [7]
- Target Encoding [7]
- Weight of Evidence [8]
- Quantile Encoder [13]
- Summary Encoder [13]

https://github.com/scikit-learn-contrib/category_encoders

# Преобразование с учётом отклика

| *Level* | $N_i$ | $\Sigma Y_i$ | $p_i$ |
|---|---|---|---|
| **A** | **1562** | **430** | **0.28** |
| **B** | **970** | **432** | **0.45** |
| **C** | **223** | **45** | **0.20** |
| **D** | **111** | **36** | **0.32** |
| **E** | **85** | **23** | **0.27** |
| **F** | **50** | **20** | **0.40** |
| **G** | **23** | **8** | **0.35** |
| **H** | **17** | **5** | **0.29** |
| **I** | **12** | **6** | **0.50** |
| **J** | **5** | **5** | **1.00** |

**«редкие значение»** переменной - источник нестабильности, недостоверности в модели

***Level –*** различные значения переменной ***X***

$N_i$**–** число наблюдений, что ***X*** принимает ***i***-е значение

$\Sigma y_i$ - сумма бинарных откликов для наблюдений, где ***X*** принимает ***i***-е значение

$p_i = \Sigma y_i / N_i$- условная вероятность положительного отклика, если ***X*** принимает ***i***-е значение

# Преобразование с учётом отклика

| *Level* | $N_i$ | $\Sigma Y_i$ | $p_i$ |
|---|---|---|---|
| **J** | **5** | **5** | **1.00** |
| **I** | **12** | **6** | **0.50** |
| **B** | **970** | **432** | **0.45** |
| **F** | **50** | **20** | **0.40** |
| **G** | **23** | **8** | **0.35** |
| **D** | **111** | **36** | **0.32** |
| **H** | **17** | **5** | **0.29** |
| **A** | **1562** | **430** | **0.28** |
| **E** | **85** | **23** | **0.27** |
| **C** | **223** | **45** | **0.20** |

Сортируем по $\boldsymbol{p_i}$, если значения ***X*** «рядом» после сортировки, то можно предположить, что они «похоже» влияют на отклик

# Кодирование категориального признака порядковой (ординальной) переменной

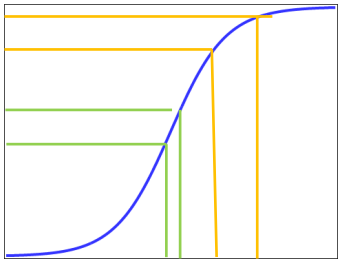
| $X'$ | $N_i$ | $\Sigma Y_i$ | $p_i$ |
|---|---|---|---|
| 1 | 5 | 5 | 1.00 |
| 2 | 12 | 6 | 0.50 |
| 3 | 970 | 432 | 0.45 |
| 4 | 50 | 20 | 0.40 |
| 5 | 23 | 8 | 0.35 |
| 6 | 111 | 36 | 0.32 |
| 7 | 17 | 5 | 0.29 |
| 8 | 1562 | 430 | 0.28 |
| 9 | 85 | 23 | 0.27 |
| 10 | 223 | 45 | 0.20 |

Можем отобразить категориальный ***X*** на порядковую шкалу (новая переменная ***X'***), но не учитываем насколько похожи отклики и не решается проблема редких значений

# Шансы

| *Level* | $N_i$ | $\Sigma Y_i$ | $p_i$ | $\log(p_i/1-p_i)$ | |
|---|---|---|---|---|---|
| J | 5 | 5 | 1.00 | . | |
| I | 12 | 6 | 0.50 | 0.00 | $\Delta p_i = 0.05 \Rightarrow$ |
| B | 970 | 432 | 0.45 | -0.10 | $\Delta\text{logit}(p_i)$=0.1 |
| F | 50 | 20 | 0.40 | -0.18 | |
| G | 23 | 8 | 0.35 | -0.27 | |
| D | 111 | 36 | 0.32 | -0.32 | |
| H | 17 | 5 | 0.29 | -0.38 | $\Delta p_i = 0.05 \Rightarrow$ |
| A | 1562 | 430 | 0.28 | -0.42 | $\Delta\text{logit}(p_i)$=0.11 |
| E | 85 | 23 | 0.27 | -0.43 | |
| C | 223 | 45 | 0.20 | -0.60 | |

Рассмотрим логарифм шансов положительного отклика для ***i***-го значения ***X***, т.е. **logit** от $p_i$, сортировка не меняется, но более корректно учитываются различия в областях определенности (около 0 и 1) и неопределенности



# Группировка значений переменной по шансам

| *Level* | $N_i$ | $\Sigma Y_i$ | $p_i$ | $\log(p_i/1-p_i)$ |
|---|---|---|---|---|
| J | 5 | 5 | 1.00 | . |
| I | 12 | 6 | 0.50 | 0.00 |
| B | 970 | 432 | 0.45 | -0.10 |
| F | 50 | 20 | 0.40 | -0.18 |
| G | 23 | 8 | 0.35 | -0.27 |
| D | 111 | 36 | 0.32 | -0.32 |
| H | 17 | 5 | 0.29 | -0.38 |
| A | 1562 | 430 | 0.28 | -0.42 |
| E | 85 | 23 | 0.27 | -0.43 |
| C | 223 | 45 | 0.20 | -0.60 |

Можем агрегировать (например, с помощью одномерной кластеризации) «похожие» значения ***X***, объединив их в однородные (с точки зрения поведения отклика) группы …

# Группировка значений переменной по шансам

| $X''$ | $N_i$ | $\Sigma Y_i$ | $p_i$ | $\log(p_i/1-p_i)$ |
|---|---|---|---|---|
| CL1 | 1037 | 463 | 0.45 | -0.09 |
| CL2 | 134 | 44 | 0.33 | -0.31 |
| CL3 | 1664 | 458 | 0.28 | -0.42 |
| CL4 | 223 | 45 | 0.20 | -0.60 |

… и создать новую категориальную переменную ***X''***, без редких уровней и с меньшим числом различных значений – уменьшаем число степеней свободы модели, увеличиваем стабильность модели и уменьшаем шанс переобучиться

# Weight of evidence и шансы

- Снова **формула Байеса** (еще будем разбираться подробнее):
  - Обозначения : маленькие p – плотности, большие P – вероятности
  - $p(x)$ - плотность распределения наблюдений в пространстве признаков
  - $P(y)$ - априорная и $P(y|x)$ - апостериорная вероятности классов
  - $p(x|y)$ – функция правдоподобия
  - Совместная плотность распределения:

$$\boldsymbol{p(x,y) = p(x)P(y|x) = P(y)p(x|y) \Rightarrow P(y|x) = P(y)p(x|y)/p(x)}$$

- Логарифм условных шансов:

$$\log\left(\frac{P(y=1|x)}{P(y=0|x)}\right) = \log\left(\frac{P(y=1)p(x|y=1)/p(x)}{P(y=0)p(x|y=0)/p(x)}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) + \log\left(\frac{p(x|y=1)}{p(x|y=0)}\right)$$

- WOE и «наивное» предположение (независимость переменных):

$$\log\left(\frac{P(y=1|x)}{P(y=0|x)}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) + \sum_{j=1}^{p} \mathrm{WOE}(x_j), \quad \text{где } \mathrm{WOE}(x_j) = \log\left(\frac{p(x_j|y=1)}{p(x_j|y=0)}\right)$$

Априорные шансы (log равен 0, если выборка «сбалансирована»)

Вклад шансов каждой из p переменных

# Weight of Evidence для категориальной переменной

- Пусть в задаче с бинарным откликом категориальная (или дискретизированная) переменная $x_j$ принимает $k$ различных значений $\{v_1, \dots, v_k\}$, тогда (опять по формуле Байеса):
  - $WOE(x_j)$ - $j$–й переменной: $WOE(x_j) = \sum_{i=1}^{k} \mathrm{WOE}_i(x_j)$,
  - где $WOE_i(x_j)$ - weight of evidence $i$–го значения $v_i$

$$\mathrm{WOE}_i(x_j) = \log\left(\frac{P(x_j = v_i | y = 1)}{P(x_j = v_i | y = 0)}\right) = \log\left(\frac{count((y = 1) \text{ and } (x_j = v_i))/count(y = 1)}{count((y = 0) \text{ and } (x_j = v_i))/count(y = 0)}\right)$$

- Интерпретация WOE:
  - ***WOE**i* **> 0***: (ОШ>1) $i$-$e$ значение связано с более высоким шансом положительного отклика
  - ***WOE**i* **< 0***: (ОШ<1) $i$-$e$ значение связано с более низким шансом положительного отклика
  - ***WOE**i* **= 0***: (ОШ=1) $i$-$e$ значение не влияет на уровень отклика
  - ***WOE*** в целом оказывает *предиктивную силу* категориальной (или дискретезированной) переменной и ее отдельных значений

# Weight of Evidence

| *Level* | $N_i$ | $\Sigma Y_i$ | $p_i$ | $WOE_i$ |
|---|---|---|---|---|
| J | 5 | 5 | 1.00 | . |
| I | 12 | 6 | 0.50 | 0.71 |
| B | 970 | 432 | 0.45 | 0.49 |
| F | 50 | 20 | 0.40 | 0.3 |
| G | 23 | 8 | 0.35 | 0.08 |
| D | 111 | 36 | 0.32 | -0.03 |
| H | 17 | 5 | 0.29 | -0.17 |
| A | 1562 | 430 | 0.28 | -0.26 |
| E | 85 | 23 | 0.27 | -0.28 |
| C | 223 | 45 | 0.20 | -0.67 |
| | 3058 | 1010 | | 0.17 |

Не решает проблему редких уровней

# Информационная важность переменных

- Дивергенция Кульбака-Лейблера (различающая информация, относительная энтропия и другие термины):
  - Ассиметричная мера расхождения двух распределений
  - Для дискретных распределений P и Q: $D_{KL}(P||Q) = \sum_i \ p_i \log(p_i/q_i)$
  - Симметричный вариант: $D_{KL}(P;Q) = D_{KL}(P||Q) + D_{KL}(Q||P)$
- IV **(информационная важность** или информационный индекс) переменной:
  - Симметричная дивергенция (расстояние) Кульбака-Лейблера, которое показывает насколько отличаются распределения переменной (отдельных значений переменной) внутри положительного и отрицательного классов:
  - Пусть в задаче с бинарным откликом категориальная переменная $x$ принимает $k$ различных значений $\{v_1, \dots, v_k\}$, тогда:

$$IV(x) = \sum_{j=1}^{k} (P(x = v_j | y = 1) - P(x = v_j | y = 0)) WOE_j(x)$$

# Information Value

| *Level* | $N_i$ | $\Sigma Y_i$ | $p_i$ | $IV_i$ |
|---|---|---|---|---|
| J | 5 | 5 | 1.00 | . |
| I | 12 | 6 | 0.50 | 0.0021 |
| B | 970 | 432 | 0.45 | 0.0809 |
| F | 50 | 20 | 0.40 | 0.0015 |
| G | 23 | 8 | 0.35 | 0 |
| D | 111 | 36 | 0.32 | 0 |
| H | 17 | 5 | 0.29 | 0.0002 |
| A | 1562 | 430 | 0.28 | 0.033 |
| E | 85 | 23 | 0.27 | 0.0021 |
| C | 223 | 45 | 0.20 | 0.0284 |
| | 3058 | 1010 | | 0.1482 |

- Эвристические пороги на IV:
  - Меньше 0.02 – незначимая переменная
  - 0.02 – 0.10 низкая прогнозная сила
  - 0.10 – 0.30 средняя прогнозная сила
  - 0.30 – 0.50 высокая прогнозная сила
  - Больше 0.50 – что-то пошло не так

# Пример

```
import pandas as pd
import numpy as np
mydata = pd.read_csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
```

```
mydata.head(10)
```

| | admit | gre | gpa | rank |
|---|---|---|---|---|
| **0** | 0 | 380 | 3.61 | 3 |
| **1** | 1 | 660 | 3.67 | 3 |
| **2** | 1 | 800 | 4.00 | 1 |
| **3** | 1 | 640 | 3.19 | 4 |
| **4** | 0 | 520 | 2.93 | 4 |
| **5** | 1 | 760 | 3.00 | 2 |
| **6** | 1 | 560 | 2.98 | 1 |
| **7** | 0 | 400 | 3.08 | 2 |
| **8** | 1 | 540 | 3.39 | 3 |
| **9** | 0 | 700 | 3.92 | 2 |

```
def iv_woe(data, target, bins=10, show_woe=False):

    #Empty Dataframe
    newDF,woeDF = pd.DataFrame(), pd.DataFrame()

    #Extract Column Names
    cols = data.columns

    #Run WOE and IV on all the independent variables
    for ivars in cols[~cols.isin([target])]:
        if (data[ivars].dtype.kind in 'bifc') and (len(np.unique(data[ivars]))>10):
            binned_x = pd.qcut(data[ivars], bins,  duplicates='drop')
            d0 = pd.DataFrame({'x': binned_x, 'y': data[target]})
        else:
            d0 = pd.DataFrame({'x': data[ivars], 'y': data[target]})
        d0 = d0.astype({"x": str})
        d = d0.groupby("x", as_index=False, dropna=False).agg({"y": ["count", "sum"]})
        d.columns = ['Cutoff', 'N', 'Events']
        d['% of Events'] = np.maximum(d['Events'], 0.5) / d['Events'].sum()
        d['Non-Events'] = d['N'] - d['Events']
        d['% of Non-Events'] = np.maximum(d['Non-Events'], 0.5) / d['Non-Events'].sum()
        d['WoE'] = np.log(d['% of Non-Events']/d['% of Events'])
        d['IV'] = d['WoE'] * (d['% of Non-Events']-d['% of Events'])
        d.insert(loc=0, column='Variable', value=ivars)
        print("Information value of " + ivars + " is " + str(round(d['IV'].sum(),6)))
        temp =pd.DataFrame({"Variable" : [ivars], "IV" : [d['IV'].sum()]}, columns = ["Variable", "IV"])
        newDF=pd.concat([newDF,temp], axis=0)
        woeDF=pd.concat([woeDF,d], axis=0)

        #Show WOE Table
        if show_woe == True:
            print(d)
    return newDF, woeDF
```

```
iv, woe = iv_woe(data = mydata, target = 'admit', bins=10, show_woe = True)
```

# Пример

```
woe
```

| | Variable | Cutoff | N | Events | % of Events | Non-Events | % of Non-Events | WoE | IV |
|---|---|---|---|---|---|---|---|---|---|
| **0** | gre | (219.999, 440.0] | 48 | 6 | 0.047244 | 42 | 0.153846 | 1.180625 | 0.125857 |
| **1** | gre | (440.0, 500.0] | 51 | 12 | 0.094488 | 39 | 0.142857 | 0.413370 | 0.019994 |
| **2** | gre | (500.0, 520.0] | 24 | 10 | 0.078740 | 14 | 0.051282 | -0.428812 | 0.011774 |
| **3** | gre | (520.0, 560.0] | 51 | 15 | 0.118110 | 36 | 0.131868 | 0.110184 | 0.001516 |
| **4** | gre | (560.0, 580.0] | 29 | 6 | 0.047244 | 23 | 0.084249 | 0.578450 | 0.021406 |
| **5** | gre | (580.0, 620.0] | 53 | 21 | 0.165354 | 32 | 0.117216 | -0.344071 | 0.016563 |
| **6** | gre | (620.0, 660.0] | 45 | 17 | 0.133858 | 28 | 0.102564 | -0.266294 | 0.008333 |
| **7** | gre | (660.0, 680.0] | 20 | 9 | 0.070866 | 11 | 0.040293 | -0.564614 | 0.017262 |
| **8** | gre | (680.0, 740.0] | 44 | 12 | 0.094488 | 32 | 0.117216 | 0.215545 | 0.004899 |
| **9** | gre | (740.0, 800.0] | 35 | 19 | 0.149606 | 16 | 0.058608 | -0.937135 | 0.085278 |
| **0** | gpa | (2.259, 2.9] | 43 | 8 | 0.062992 | 35 | 0.128205 | 0.710622 | 0.046342 |
| **1** | gpa | (2.9, 3.048] | 37 | 11 | 0.086614 | 26 | 0.095238 | 0.094917 | 0.000819 |
| **2** | gpa | (3.048, 3.17] | 42 | 8 | 0.062992 | 34 | 0.124542 | 0.681634 | 0.041955 |
| **3** | gpa | (3.17, 3.31] | 42 | 10 | 0.078740 | 32 | 0.117216 | 0.397866 | 0.015308 |
| **4** | gpa | (3.31, 3.395] | 36 | 8 | 0.062992 | 28 | 0.102564 | 0.487478 | 0.019290 |
| **5** | gpa | (3.395, 3.494] | 40 | 14 | 0.110236 | 26 | 0.095238 | -0.146246 | 0.002193 |
| **6** | gpa | (3.494, 3.61] | 41 | 16 | 0.125984 | 25 | 0.091575 | -0.318998 | 0.010976 |
| **7** | gpa | (3.61, 3.752] | 39 | 20 | 0.157480 | 19 | 0.069597 | -0.816578 | 0.071764 |
| **8** | gpa | (3.752, 3.94] | 42 | 13 | 0.102362 | 29 | 0.106227 | 0.037062 | 0.000143 |
| **9** | gpa | (3.94, 4.0] | 38 | 19 | 0.149606 | 19 | 0.069597 | -0.765285 | 0.061230 |
| **0** | rank | 1 | 61 | 33 | 0.259843 | 28 | 0.102564 | -0.929588 | 0.146204 |
| **1** | rank | 2 | 151 | 54 | 0.425197 | 97 | 0.355311 | -0.179558 | 0.012548 |
| **2** | rank | 3 | 121 | 28 | 0.220472 | 93 | 0.340659 | 0.435110 | 0.052295 |
| **3** | rank | 4 | 67 | 12 | 0.094488 | 55 | 0.201465 | 0.757142 | 0.080997 |

```
iv
```

| | Variable | IV |
|---|---|---|
| **0** | gre | 0.312882 |
| **0** | gpa | 0.270020 |
| **0** | rank | 0.292044 |

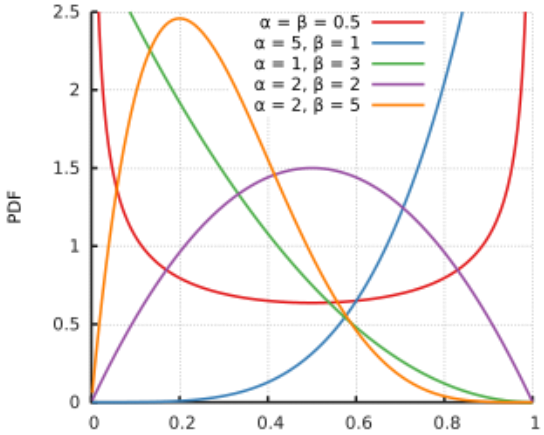
# Сглаженное WOE для борьбы с редкими значениями

| *Level* | $N_i$ | | $\Sigma Y_i$ | | $p_i$ | SWOE |
|---|---|---|---|---|---|---|
| J | 5 | +24 | 5 | +8 | 0.45 | 0.5 |
| I | 12 | +24 | 6 | +8 | 0.39 | 0.25 |
| B | 970 | +24 | 432 | +8 | 0.44 | 0.48 |
| F | 50 | | 20 | | 0.38 | 0.21 |
| G | 23 | +24 | 8 | +8 | 0.34 | 0.05 |
| D | 111 | +24 | 36 | +8 | 0.33 | -0.02 |
| H | 17 | +24 | 5 | +8 | 0.32 | -0.06 |
| A | 1562 | | 430 | | 0.28 | -0.26 |
| E | 85 | +24 | 23 | +8 | 0.28 | -0.22 |
| C | 223 | +24 | 45 | +8 | 0.21 | -0.59 |

- Для «исправления» ситуации с редкими значениями:
  - Добавим для каждого значения переменной «виртуальный» набор наблюдений (фиксированного размера) с вероятностью положительного отклика равной априорной.
  - Чем более редкий уровень, тем выше влияние априорного распределения.

# Бета распределение для кодирования категориальных предикторов по выборке

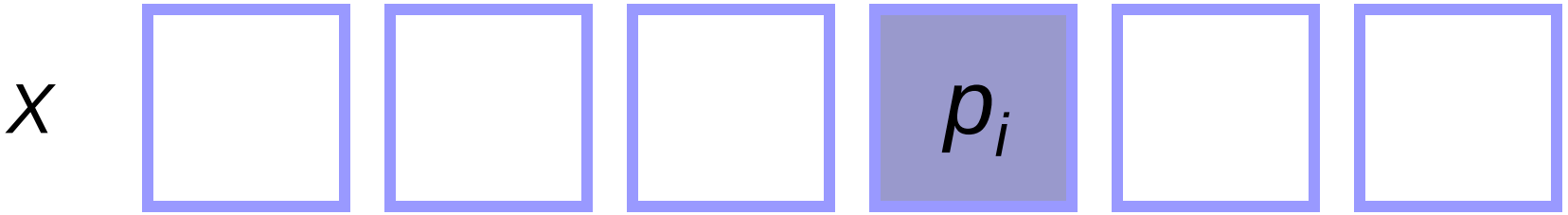
- Бета распределение:
  - двухпараметрическое (альфа и бета)
  - мат. ожидание: $\frac{a}{a+b}$
  - используется для моделирования случайных величин, заданных на интервале.

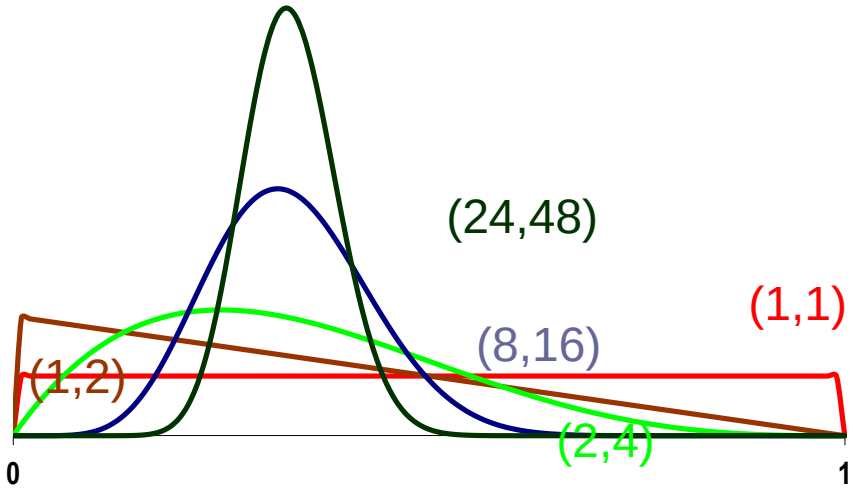



- Моделируем:
  - условную вероятность положительного отклика для фиксированного значения категориальной переменной как случайную величину $p_i \sim \text{Beta}\,(a_s, b_s)$,
  - $a_0, b_0$ - параметры априорного распределения
  - $a_s, b_s$ пересчитываются последовательно, проходя всю выборку, где категориальная переменная принимает $i$-$e$ значение, при этом …
  - $a_{s+1} = a_s + 1$, если встретили наблюдение с откликом $y = 1$,
  - $b_{s+1} = b_s + 1$, если встретили наблюдение с откликом $y = 0$,

# Сглаженный WOE

*Значения X*

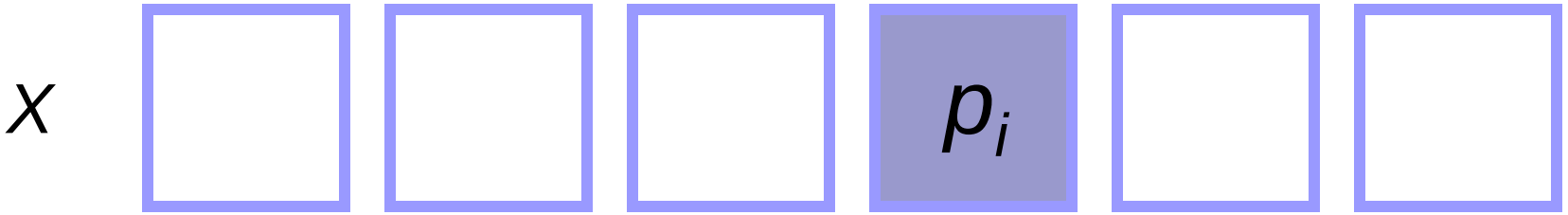


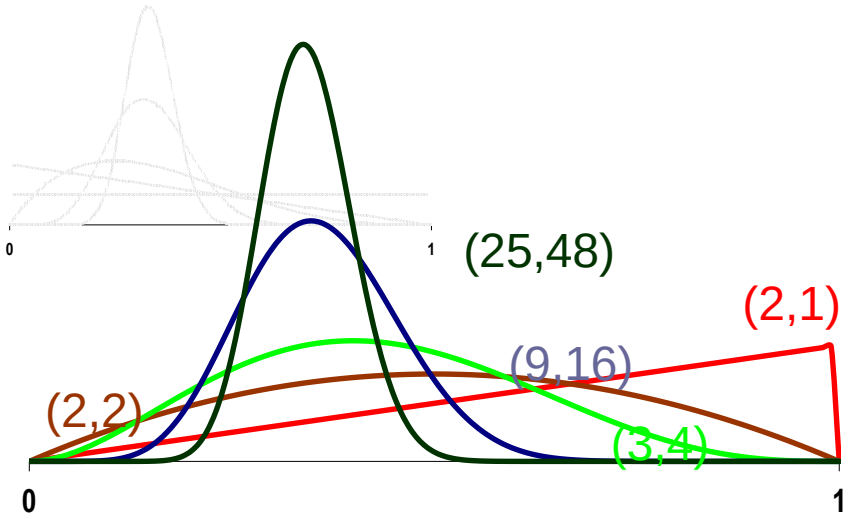




$$p_i \sim \text{Beta}\,(a, b)$$

# Сглаженный WOE

*Значение X*



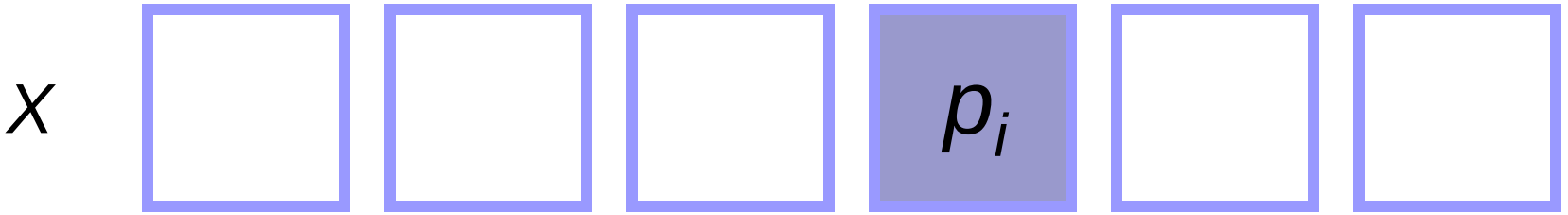



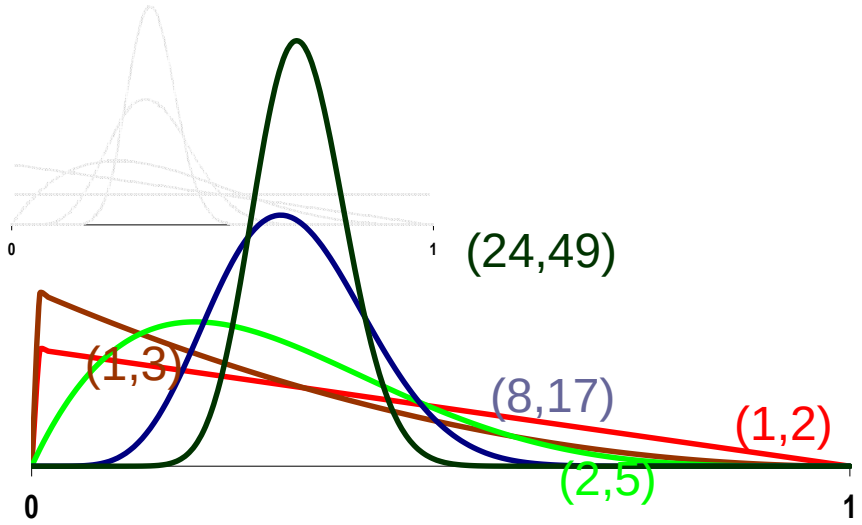

$$p_i \sim \text{Beta}\,(a, b)$$

$$p_i | y = 1 \sim \text{Beta}\,(a + 1, b)$$

# Сглаженный WOE

*Значения X*



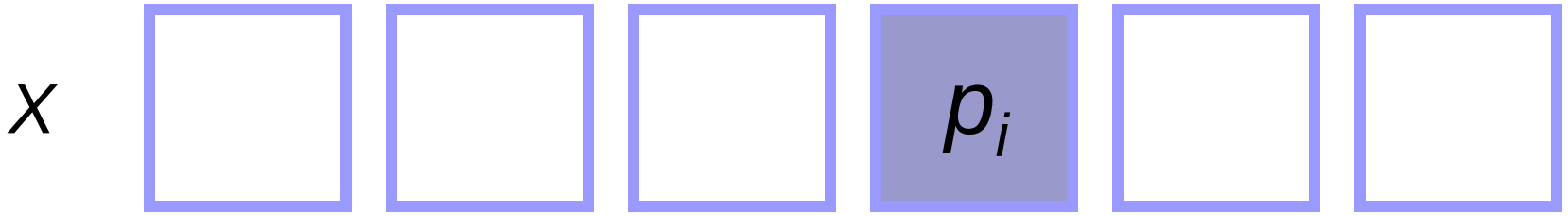

$$p_i \sim \text{Beta}\,(a, b)$$

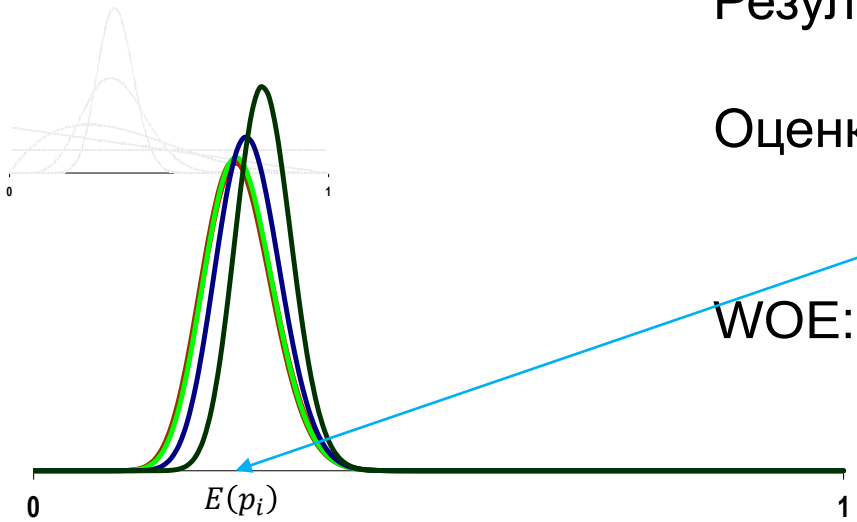
$$p_i | y = 1 \sim \text{Beta}\,(a + 1, b)$$

$$p_i | y = 0 \sim \text{Beta}\,(a, b + 1)$$

# Сглаженный WOE

*Значения X*




Результат:

$$p_i \sim \text{Beta}\,(a + n_1, b + n_0)$$

Оценка:

$$E(p_i) = \frac{n_1 + a}{n_1 + n_0 + a + b}$$

WOE:

$$\log\left(\frac{E(p_i)}{1 - E(p_i)}\right)$$

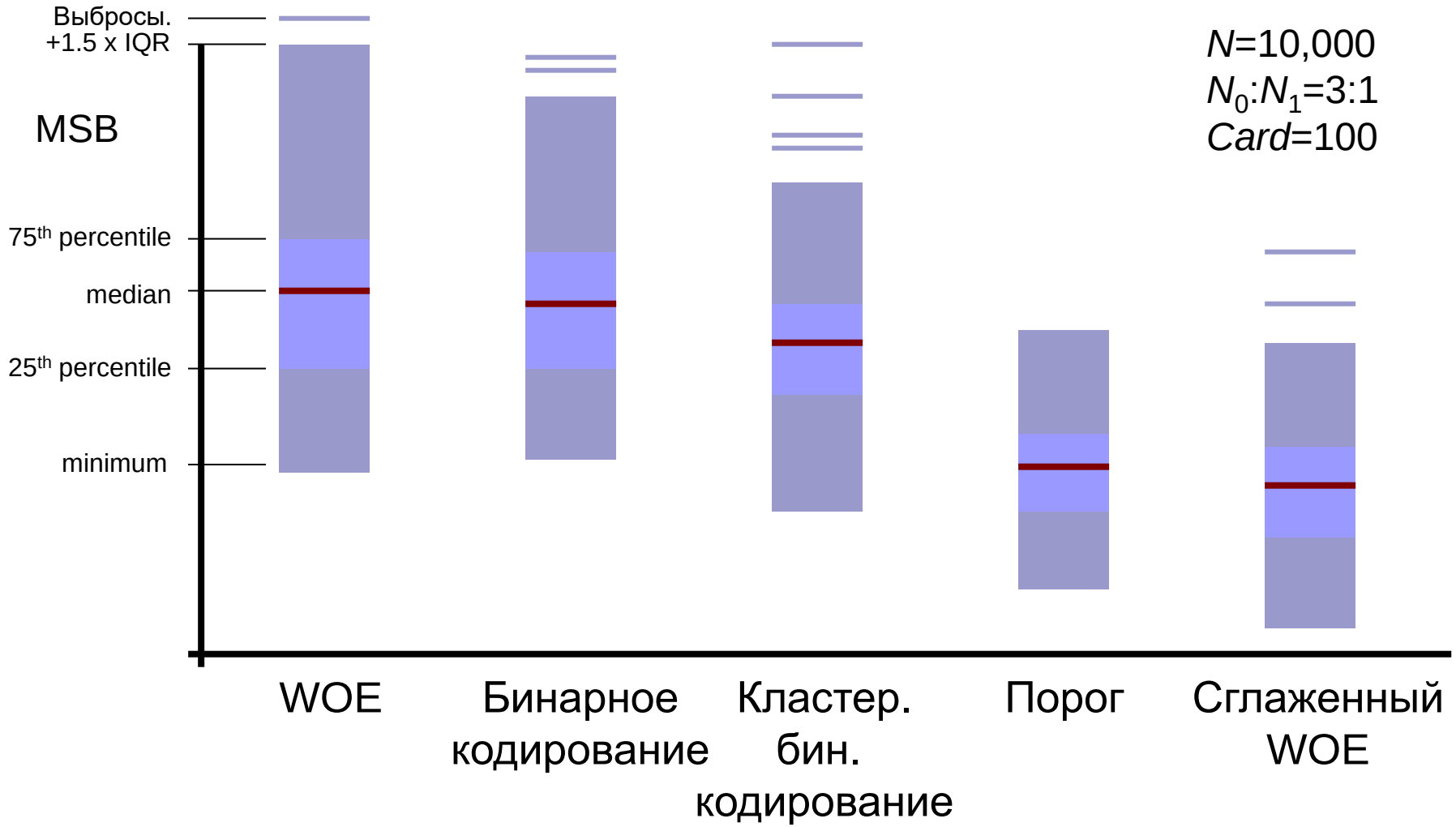
Наблюдаемое число событий, если x принимает i-е значение

$$SWOE_i(x) = \log\left(\frac{n_1 + c\rho_1}{n_0 + c(1 - \rho_1)}\right)$$

Априорная вероятность события

Параметр регуляризации

# Имитационный эксперимент



# Практическое применение WOE и IV

- Моделенезависимый отбор важных переменных:
  - Дискретизируем (например, на квантили  или равные интервалы) числовые переменные, категориальные берем как есть
  - Для всех считаем IV, сортируем переменные по убыванию, оставляем топ k самых важных

$$x_{(1)}, x_{(2)}, \dots, x_{(k)}, x_{(k+1)} \dots, x_{(p)}, \text{ где } IV\left(x_{(s)}\right) \geq IV\left(x_{(s+1)}\right)$$

- Моделенезависимая оценка качества разбиения на тренировочную и проверочную выборку
  - Дискретизируем числовые переменные, категориальные как есть
  - Разбиваем набор данных $Z = Z_{test} \cup Z_{train}$ и по переменным считаем IV на тренировочном и тестовом наборе
  - Если для некоторой переменной $x$: $||IV_{test}(x) - IV_{train}(x)|| > \Delta$, то производим разбиение заново

# Практическое применение WOE и IV

- Отображение категориальных переменных на числовую шкалу для сокращения числа степеней свободы и упрощения модели:

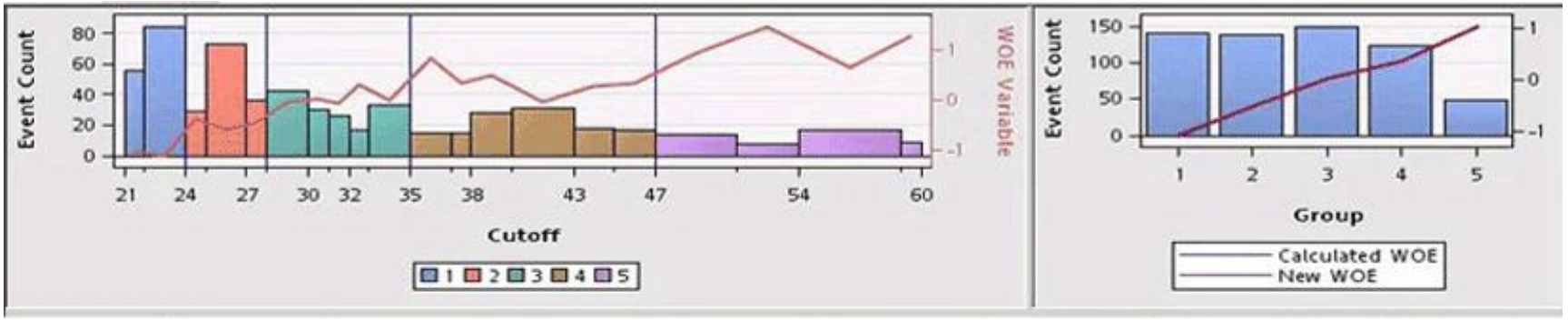
$$x_{new} = WOE(x_{old})$$

| | admit | gre | gpa | rank |
|---|---|---|---|---|
| 0 | 0 | 380 | 3.61 | 3 |
| 1 | 1 | 660 | 3.67 | 3 |
| 2 | 1 | 800 | 4.00 | 1 |
| 3 | 1 | 640 | 3.19 | 4 |

| | Variable | Cutoff | WoE |
|---|---|---|---|
| 0 | rank | 1 | -0.929588 |
| 1 | rank | 2 | -0.179558 |
| 2 | rank | 3 | 0.435110 |
| 3 | rank | 4 | 0.757142 |

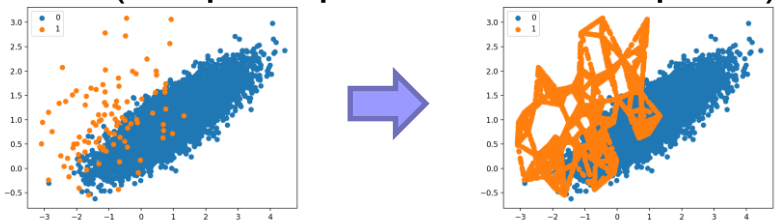
| | admit | gre | gpa | WOE rank |
|---|---|---|---|---|
| 0 | 0 | 380 | 3.61 | 0.4351 |
| 1 | 1 | 660 | 3.67 | 0.4351 |
| 2 | 1 | 800 | 4.00 | -0.9295 |
| 3 | 1 | 640 | 3.19 | 0.7571 |

- Эффективная дискретизации переменных:
  - нахождение (часто «интерактивное) такого разбиения, чтобы максимизировать IV всей переменной и по возможности добиться монотонного роста WOE

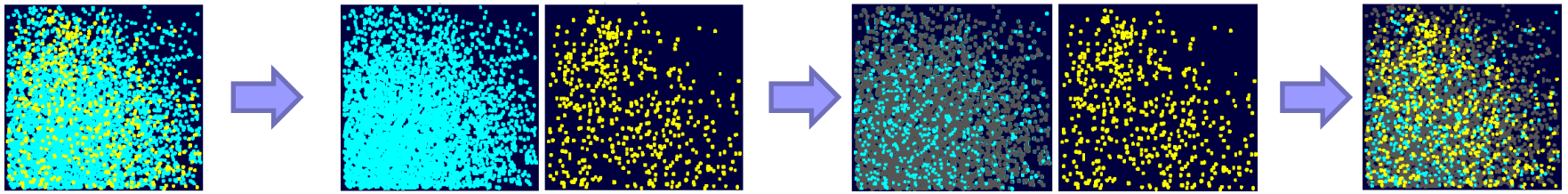


# «Балансировка» выборки

- Варианты борьбы с дисбалансом:
  - Разные **веса у наблюдений** в функции потерь (обратно пропорционально общему числу наблюдений класса)
  - **Сдвиг границы** принятия решения в дискриминантной функции в сторону редкого класса пропорционально отношению размеров
  - «Балансировка» **oversampling** – с помощью некой стратегии генерируем случайные наблюдения для выборки, увеличиваем маленький класс (например, SMOTE алгоритм):



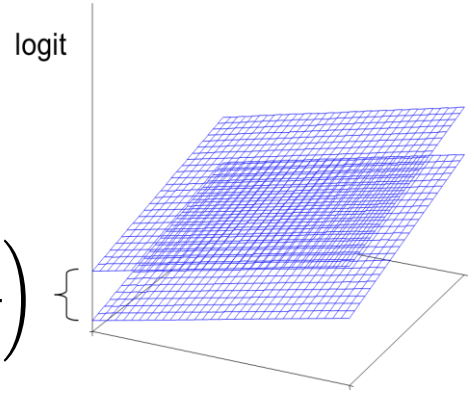
  - «Балансировка» **undersampling** – с помощью случайной выборки уменьшаем большой класс



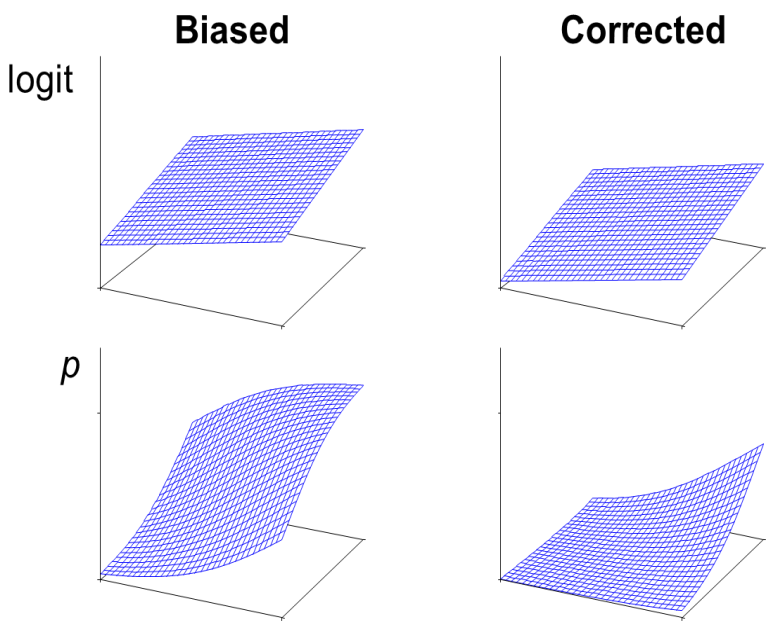
# Корректировка логистической регрессии после undersamplig

- Два способа корректировки:
  - Включить параметр «сдвига» в уравнение модели

$$g(x)^{\text{adj}} = g(x)_{logit} + b$$




$$b = \log\left(\frac{\pi_0\rho_1}{\pi_1\rho_0}\right)$$




  - Скорректировать вероятности на выходе модели:

$$p_1^{adj} = \frac{p_1\pi_1\rho_0}{p_1\pi_1\rho_0 + (1-p_1)\pi_0\rho_1}$$

$\pi_1, \pi_0$ - до undersampling

$\rho_1, \rho_0$ - после undersampling