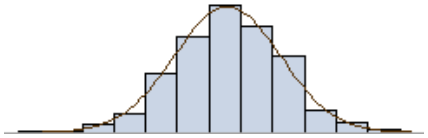
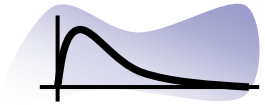
Лекция 7:
Обобщенные линейные модели, логистическая регрессия

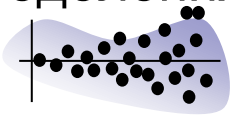
Важное предположение линейной регрессии

- Нормальное распределение ошибки с константной дисперсией:



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





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

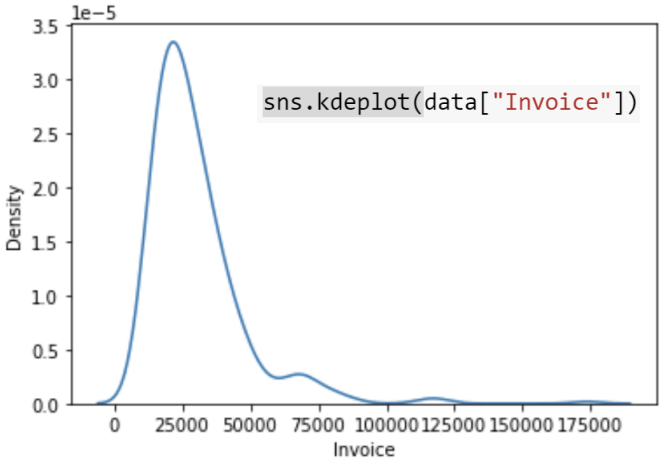
НО, в общем случае: $\mathbf{g}^{-1}(\mathbf{E}(\mathbf{g}(\mathbf{y}) \mid \mathbf{x})) \neq \mathbf{E}(\mathbf{y} \mid \mathbf{x})$

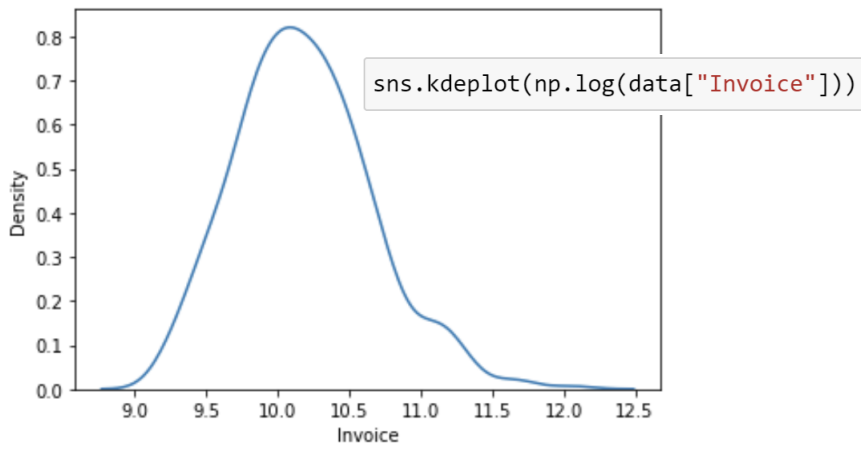
  - Использовать функцию связи: $\mathbf{g}(\mathbf{E}(\mathbf{y} \mid \mathbf{x}))$

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





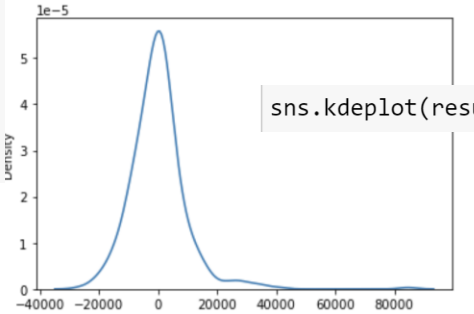
Пример (МНК) – все плохо

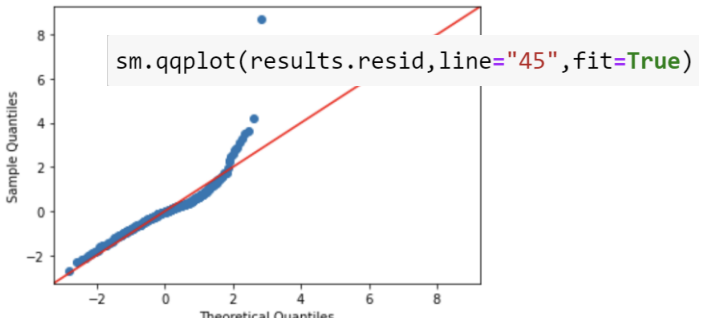
```
import statsmodels.api as sm

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

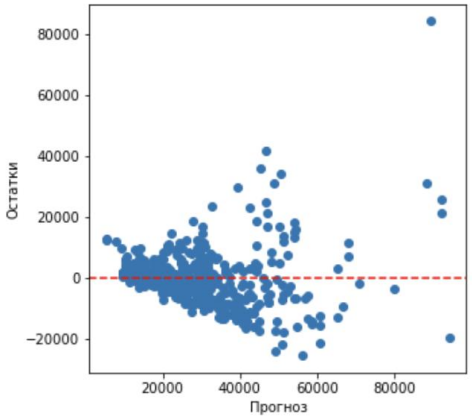
Dep. Variable:	Invoice	**R-squared:**	0.704
Model:	OLS	**Adj. R-squared:**	0.702
Method:	Least Squares	**F-statistic:**	336.3
Date:	Wed, 01 Nov 2023	**Prob (F-statistic):**	1.06e-111
Time:	01:43:01	**Log-Likelihood:**	-4531.2
No. Observations:	428	**AIC:**	9070.
Df Residuals:	424	**BIC:**	9087.
Df Model: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

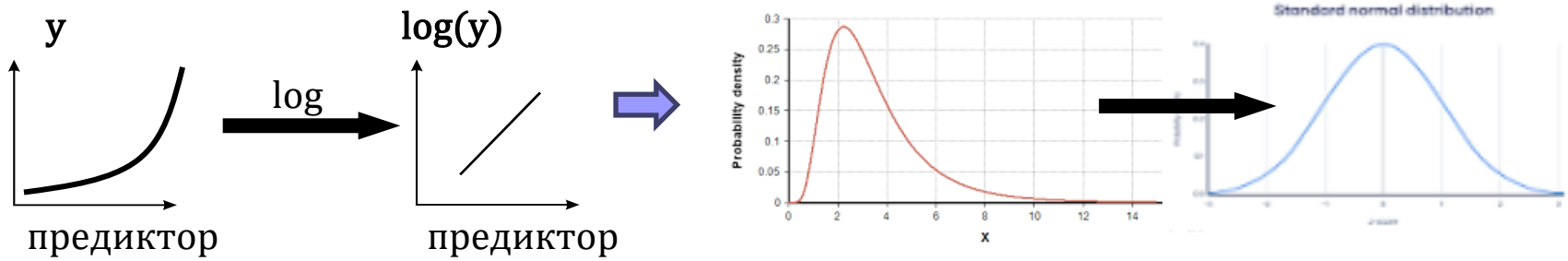






Преобразование отклика и логнормальная регрессия

- 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:

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

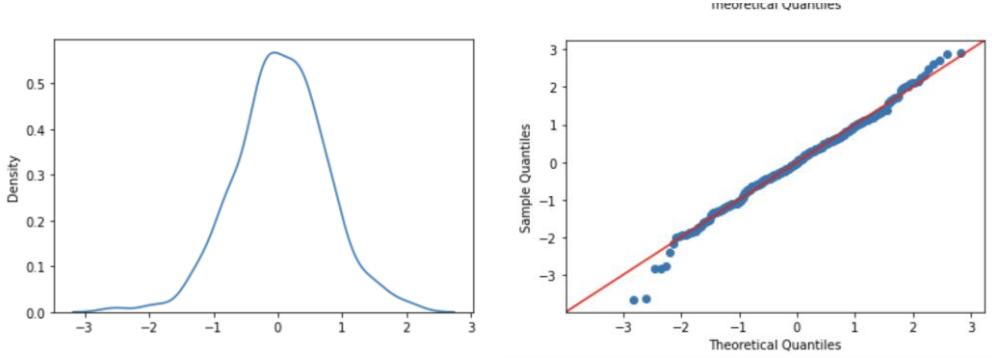
- Откуда брать σ^2? Можно взять оценку $\sigma^2 \approx MSE_{val}$, на *валидационном* наборе

Пример (логнормальная регрессия) – лучше

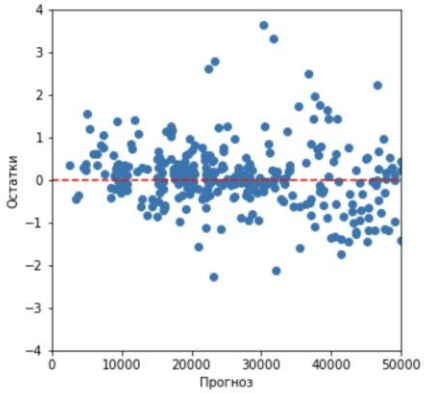
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

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 θ 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 θ (среднее)	Параметр разброса ϕ	Дисперсия	Каноническая функция связи
Линейная	непрерывный неограниченный	μ	σ	σ^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	μ/ ν^2	обратная $g(\mu)= 1/\mu$

Примеры вывода функции связи

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

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

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

- Пуассоновская регрессия (распределение Пуассона):

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

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

- Логистическая регрессия (распределение Бернулли):

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

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

Оценка отклонения (D^2)

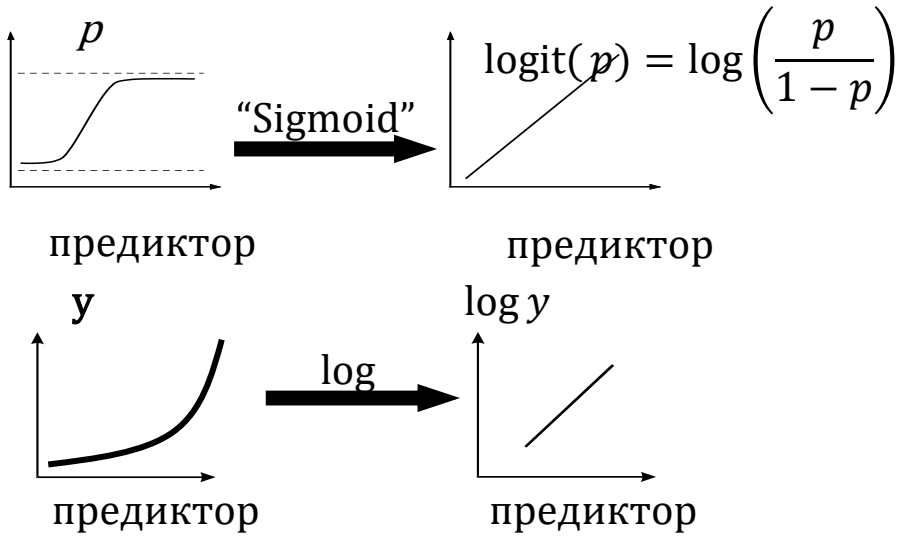
- Поиск параметров модели решается задача оптимизациимax loglik с заданным распределением и функцией связи

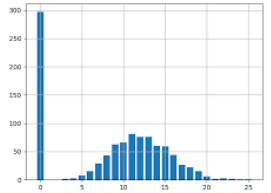
Распределение	$D^2(w) = 2[loglik(y) - loglik(\mu)]$
Normal	$D^2(\mathbf{w}) = \sum (y - \mu(\mathbf{w}))^2$
Poisson	$D^2(\mathbf{w}) = 2\sum [y \ln(y/\mu(\mathbf{w})) - (y - \mu(\mathbf{w}))]$
Gamma	$D^2(\mathbf{w}) = 2\sum [-\ln(y/\mu(\mathbf{w})) + (y - \mu(\mathbf{w}))/\mu(\mathbf{w})]$
Bernoulli	$D^2(\mathbf{w}) = -2\sum [y \ln(\mu(\mathbf{w})) + (1 - y)\ln(1 - \mu(\mathbf{w}))]$

- Нормализованное отклонение $D^2(w)/\phi$
  - Для некоторых распределений ϕ известно, также можно оценить $\hat{\phi} = Pearson\ \chi^2/(n-p)$, где $Pearson\ \chi^2 = \frac{\sum(y-\mu(w))^2}{V(\mu(w))}$, а $V(\mu(w))$ - дисперсия распределения из экспоненциального семейства (например, μ^2 для Гамма)

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 его квадрата
  - 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()
```

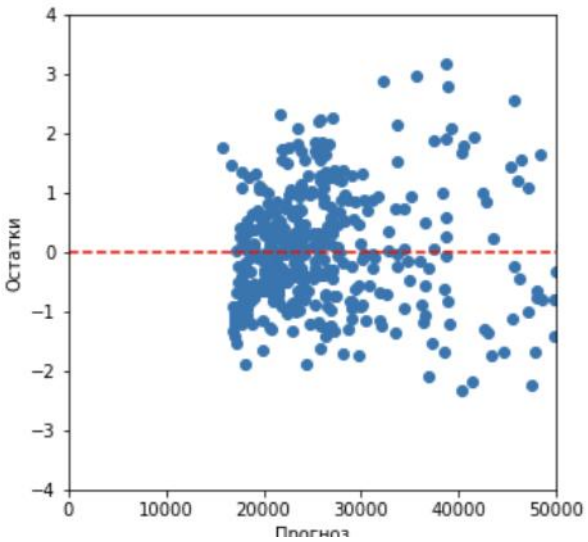
Dep. Variable:	Invoice	**No. Observations:**	428
Model:	GLM	**Df Residuals:**	424
Model Family:	Gamma	**Df Model:**	3
Link Function:	inverse_power	**Scale:**	0.11306
Method:	IRLS	**Log-Likelihood:**	-5686.6
Date:	Wed, 01 Nov 2023	**Deviance:**	310.53
Time:	02:03:07	**Pearson chi2:**	47.9
No. Iterations:	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 = '--')
```



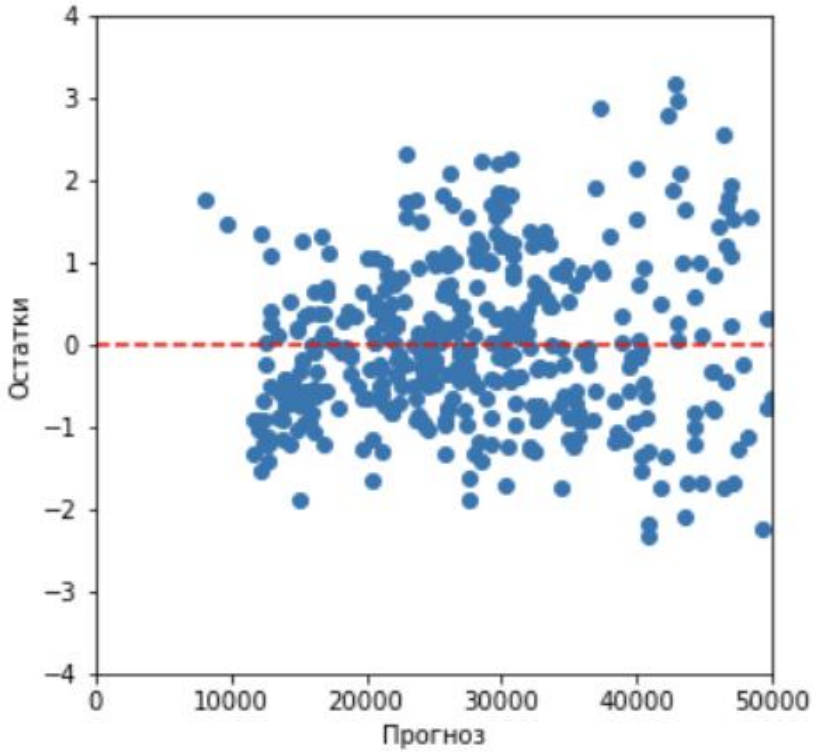
гетероскедастичность?

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

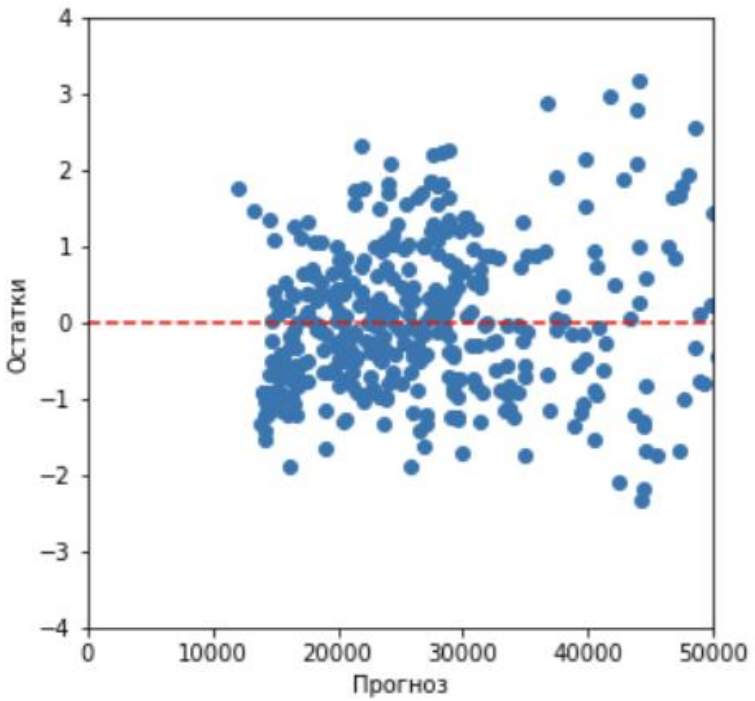


Пример гамма регрессии с неканонической функцией связи log

```
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 курса)
  - статистика 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:

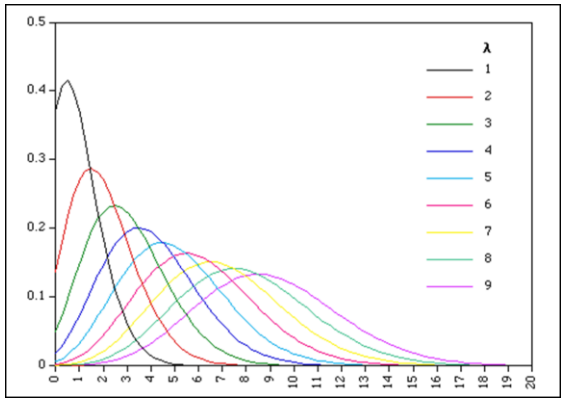
$$\log(E(y|x)) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_px_k \Rightarrow$$
$$\mu(w) = e^{w_0} \cdot e^{w_1x_1} \ldots e^{w_px_p}$$

- 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

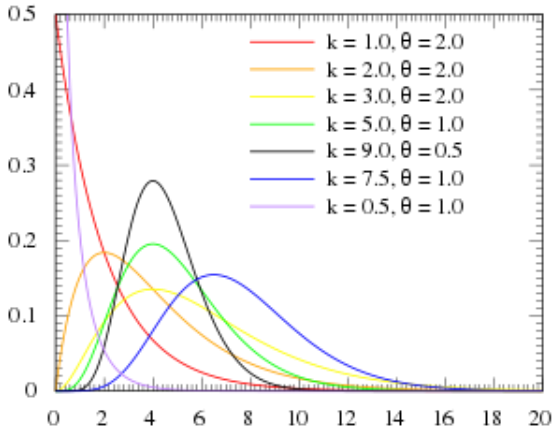
Пуассоновская регрессия

- 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

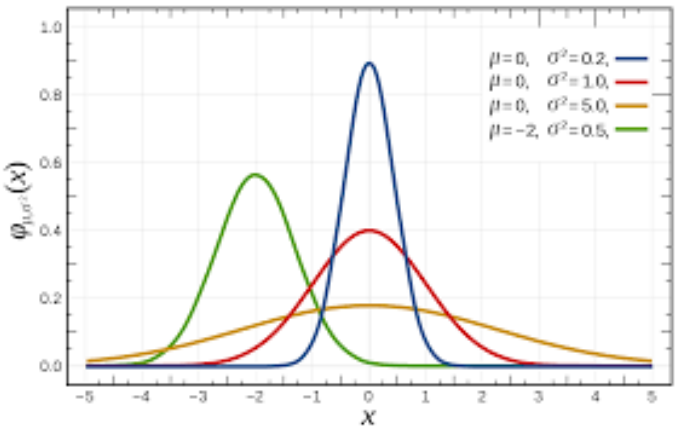
Пуассоновское



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

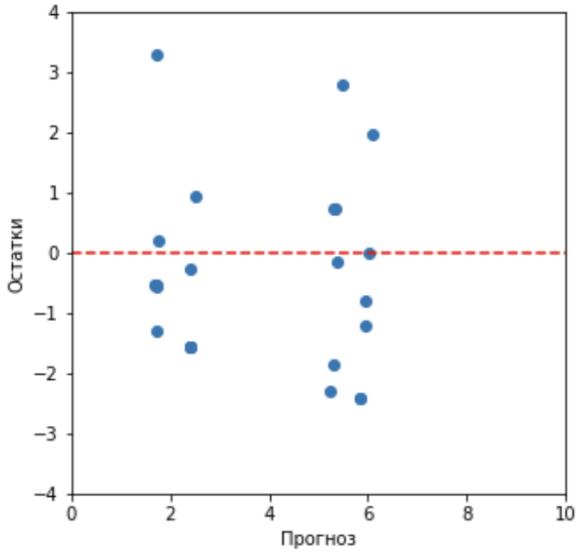
Dep. Variable:	damages	**No. Observations:**	34
Model:	GLM	**Df Residuals:**	28
Model Family:	Poisson	**Df Model:**	5
Link Function:	Log	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-125.73
Date:	Tue, 31 Oct 2023	**Deviance:**	153.59
Time:	02:55:48	**Pearson chi2:**	151.
No. Iterations:	6	**Pseudo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

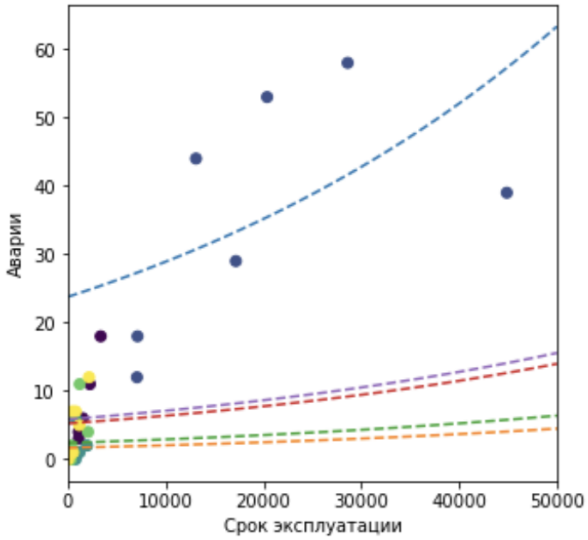
Пример пуассоновской регрессии

```
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>
```

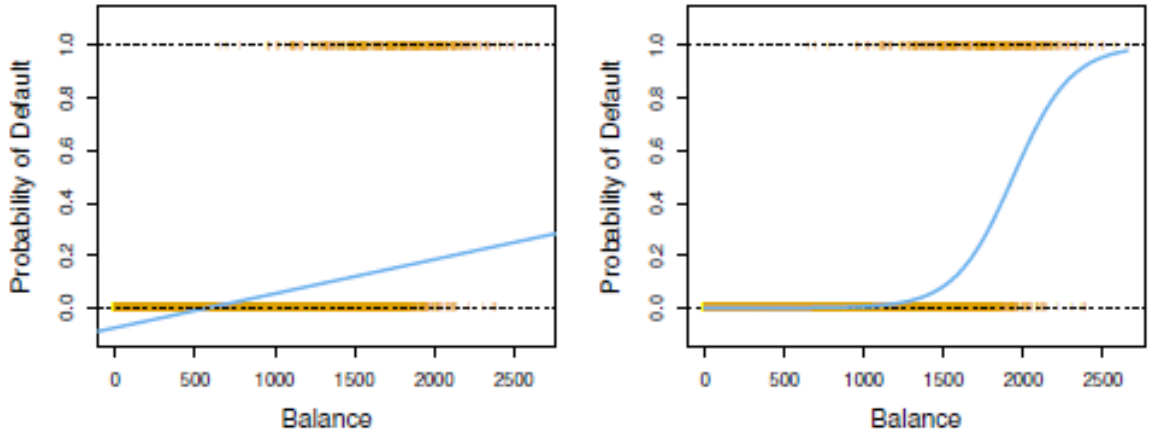



```
fig, ax = plt.subplots(figsize=(5, 5))
ax.scatter(dt['months'], dt["damages"],c=dt["type"])
m = 50000
p = 100
plt.xlim(0, m)
ax.set_ylabel('Аварии')
ax.set_xlabel('Срок эксплуатации')

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

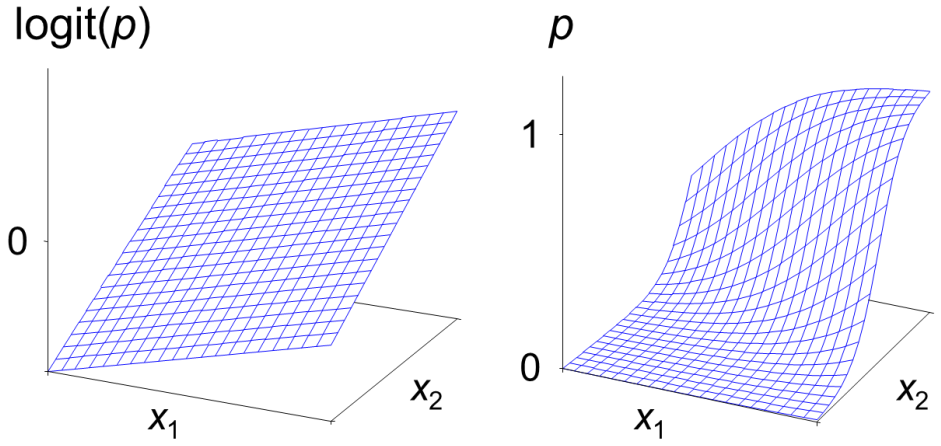
$$p_i = p(y = 1|x) = 1 - p(y = -1|x)$$

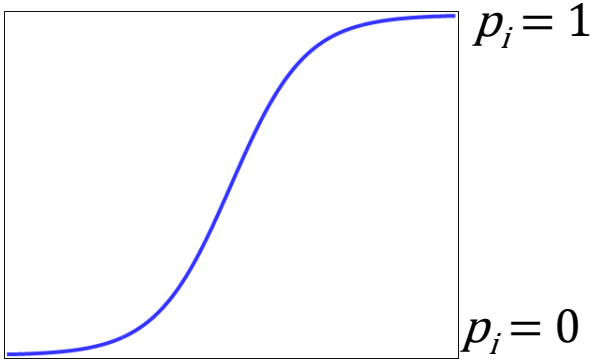
Функция связи (логит) и обратная ей (логистическая):

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

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

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





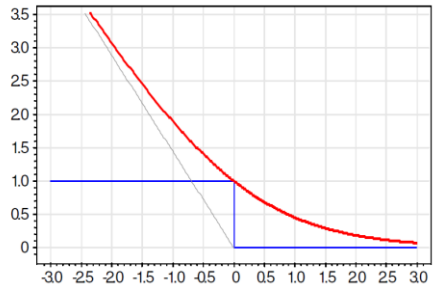
меньше $\leftarrow \mu \rightarrow$ больше

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

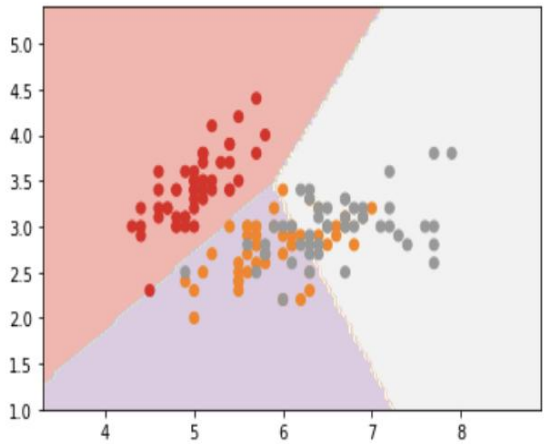
- На каждом шаге:
  - МНК линейной регрессии с взвешенными наблюдениями и модифицированными остатками, старающийся улучшить эмпирический риск на самых «сложных» примерах:

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

- 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 0.5, 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()
```



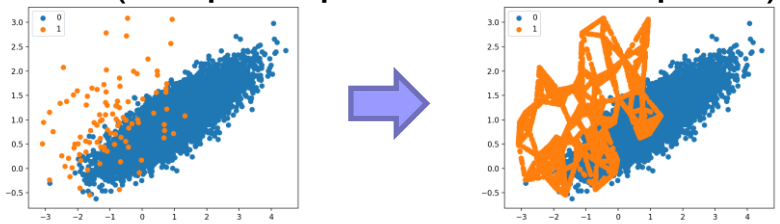
- Логистическая регрессия с двумя классами обобщается на случай K классов (многомерная логистическая функция):

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

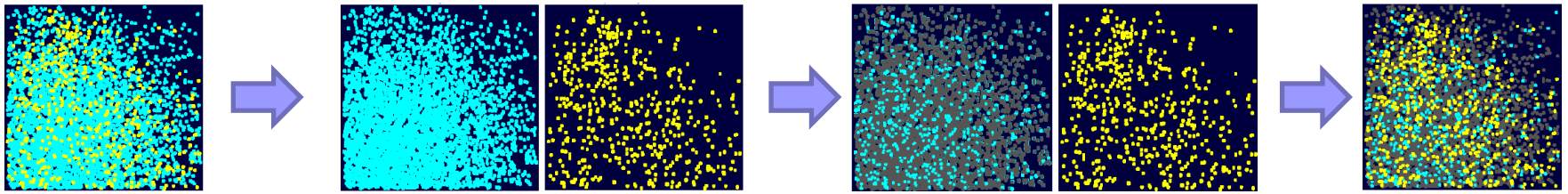
- Для *каждой* пары классов существует своя граница - линейная разделяющая функция, где вероятности классов совпадают
- Многоклассовая логистическая регрессия также называется *мультиномиальной регрессией, а* многомерная логистическая функция -softmax, *которая* «нормализует» K-мерный вектор так, чтобы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

«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):



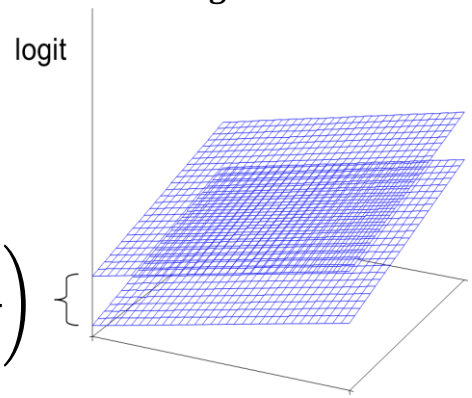
  - «Балансировка» **undersampling** – с помощью случайной выборки уменьшаем большой класс



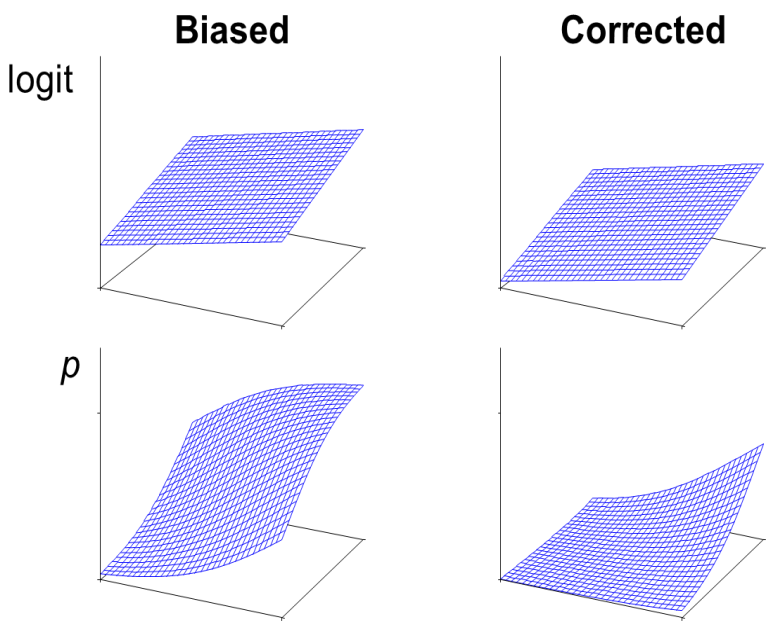
Корректировка логистической регрессии после undersamplig

- 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:

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

π_1, π_0 - до undersampling

ρ_1, ρ_0 - после undersampling

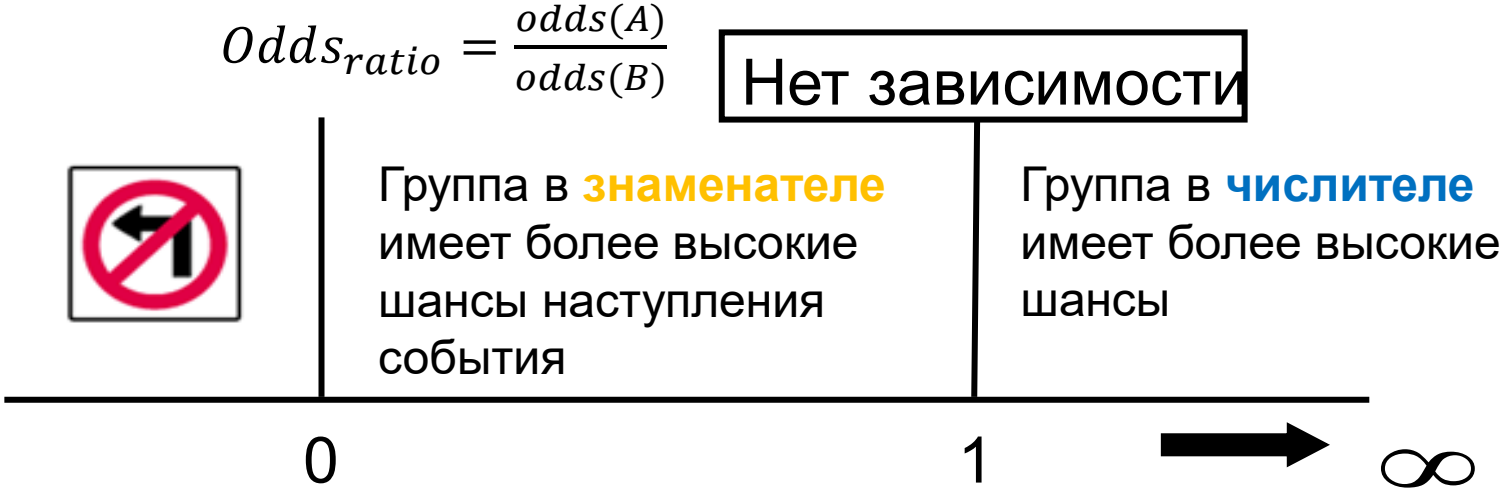
Оценка «силы» ассоциации между предиктором и бинарным откликом

- **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):

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

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

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

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

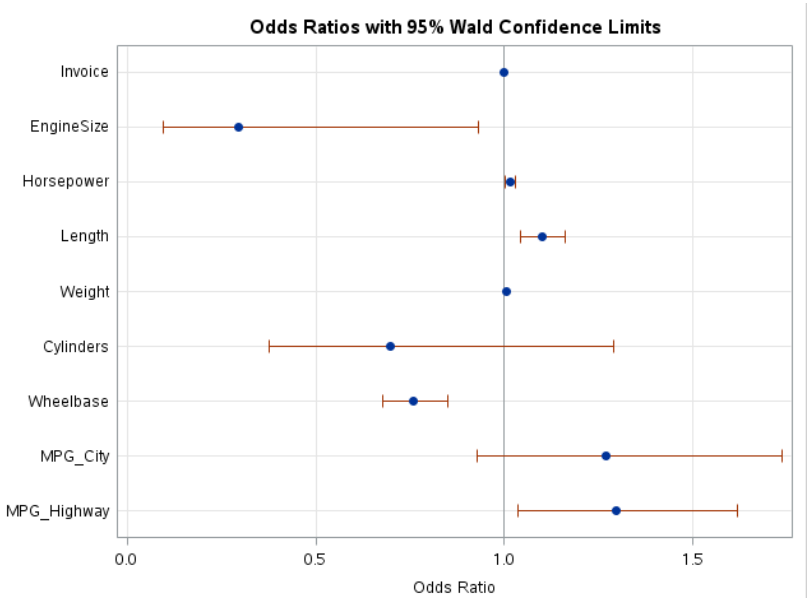
$$odds_{ratio} = \frac{odds'}{odds} = \exp(w_i)$$

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

Отношение шансов и важность переменных

exp(.)

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



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