



Санкт-Петербургский государственный университет  
Кафедра системного программирования

# Корреляционный анализ в Python (Pearson, Spearman, p-value, доверительные интервалы)

Альшаеб Басель, группа 24.M71-мм

Курс «Методы статистической обработки информации»

**Преподаватель:** д.ф.-м.н. профессор Н.К. Кривулин

Санкт-Петербург  
2025

# План выступления

- ❶ Постановка задачи и данные.
- ❷ Краткое введение в инструмент (Python, экосистема, интерфейсы).
- ❸ Демонстрация на примерах корреляционного анализа (Pearson/Spearman, p-value, CI).
- ❹ Выводы и ограничения.

**Язык:** Python 3.x    **Библиотеки:** pandas, numpy, scipy, matplotlib.

## Почему Python:

- Широкая экосистема статистики.
- Быстрый путь: данные → анализ → графики.

**Где запускать:** Jupyter Notebook/Colab/VS Code.

# Установка и запуск окружения

## Установка:

```
pip install pandas numpy scipy matplotlib scikit-learn
```

## Запуск Jupyter:

```
jupyter notebook
```

## Структура демо:

- Импорт и загрузка данных (`load_diabetes`).
- Описательная статистика, матрица корреляций.
- Pearson/Spearman:  $r$ ,  $p$ -value, доверительные интервалы (Fisher  $z$ ).
- Визуализация: scatter, тепловая карта.

# Виды зависимости (смысл корреляции)

- **Функциональная:**  $Y = f(X)$ .
- **Стохастическая зависимость:** изменение  $X$  влечёт изменение  $Y$  по вероятностному закону.
- **Независимость:**  $P\{X \in A, Y \in B\} = P\{X \in A\}P\{Y \in B\}$ .

Корреляционный анализ изучает меры связи и позволяет проверять гипотезы о наличии/отсутствии связи.

# Ковариация и корреляция (Пирсон)

Ковариация:

$$\text{Cov}(X, Y) = E[XY] - E[X] \cdot E[Y]$$

Коэффициент Пирсона:

$$\rho = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \in [-1, 1]$$

**Свойства:**

- Инвариантен к сдвигу и масштабу
- $\rho = \pm 1 \Leftrightarrow Y = \beta_0 + \beta_1 X$

**Корреляция:** сила линейной связи (не причинность).

## Проверка гипотезы $H_0 : \rho = 0$ и доверительный интервал

Пусть  $r$  — выборочная корреляция,  $n$  — размер выборки.

**t-статистика:**

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \sim t_{n-2} \text{ при } H_0.$$

Критерий: отвергаем  $H_0$  при  $|t| > t_{1-\alpha/2, n-2}$  или по p-value  $< \alpha$ .

**Fisher z-преобразование** для CI  $\rho$ :  $z = \frac{1}{2} \ln \frac{1+r}{1-r}$ ,

$$z \pm z_{1-\alpha/2} / \sqrt{n-3} \Rightarrow \rho = \frac{e^{2z}-1}{e^{2z}+1}.$$

## Pearson vs. Spearman (когда какой)

	<b>Pearson</b> $r$	<b>Spearman</b> $\rho_s$
Связь	Линейная	Монотонная (по рангам)
Выбросы	Чувствителен	Робастнее
Данные	Интервальные/отношений	Порядковые/монотонные



## Данные для демонстрации

**Набор:** `sklearn.datasets.load_diabetes()` (медицина, 442 наблюдения).  
Непрерывные признаки (возраст, ИМТ, АД и др.), целевая `target` — прогресс заболевания.  
Удобно для матрицы корреляций и парных анализов.

## Импорт и подготовка

```
import numpy as np, pandas as pd, matplotlib.pyplot as plt
from scipy import stats
from sklearn.datasets import load_diabetes
```

```
data = load_diabetes()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target, name="target")
df = pd.concat([X, y], axis=1)
```

```
df.head()           # первые строки
df.describe()       # описательная статистика
```

## Матрица корреляций (Pearson)

```
corr_pearson = df.corr(numeric_only=True) # Pearson по умолчанию
print(corr_pearson.round(3))

plt.figure()
plt.imshow(corr_pearson, vmin=-1, vmax=1)
plt.colorbar(); plt.title("Correlation_matrix_(Pearson)")
plt.xticks(range(len(corr_pearson.columns)), corr_pearson.columns, r
plt.yticks(range(len(corr_pearson.index)), corr_pearson.index)
plt.tight_layout(); plt.show()
```

Интерпретация: пары с  $|r| \gtrsim 0.3-0.4$  — кандидаты на значимую линейную связь.

## Pearson: $r$ и $p$ -value для пары признаков

```
x = df["bmi"]          # индекс массы тела
y = df["target"]       # прогресс заболевания
r_xy, p_xy = stats.pearsonr(x, y)
print(f"Pearson r(bmi, target) = {r_xy:.3f}, p-value = {p_xy:.3g}")
```

Гипотеза  $H_0: \rho = 0$ . Если  $p < 0.05$ , связь статистически значима. Знак  $r$  укажет направление связи.

## Spearman (ранговая корреляция)

```
corr_spearman = df.corr(numeric_only=True, method="spearman")  
print(corr_spearman.round(3))
```

```
r_s, p_s = stats.spearmanr(df["bmi"], df["target"])  
print(f"Spearman_rho(bmi, target)={r_s:.3f}, p-value={p_s:.3g}")
```

Когда лучше Spearman: выбросы, нелинейная *монотонная* связь, порядковые шкалы.

## Доверительный интервал для $\rho$ (Fisher z)

```
def fisher_ci(r, n, alpha=0.05):  
    z = 0.5*np.log((1+r)/(1-r))  
    z_se = 1/np.sqrt(n-3)  
    z_crit = stats.norm.ppf(1-alpha/2)  
    z_lo, z_hi = z - z_crit*z_se, z + z_crit*z_se  
    to_r = lambda z: (np.exp(2*z)-1)/(np.exp(2*z)+1)  
    return to_r(z_lo), to_r(z_hi)
```

```
n = len(df)  
lo, hi = fisher_ci(r_xy, n)  
print(f"95% CI for rho(bmi, target): [{lo:.3f}, {hi:.3f}]")
```

Интерпретация: интервал не пересекает 0  $\Rightarrow$  линейная связь статистически значима.

## Визуализация: scatter-плоты

```
plt.figure()  
plt.scatter(df["bmi"], df["target"])  
plt.xlabel("bmi"); plt.ylabel("target")  
plt.title("bmi_vs_target_(scatter)")  
plt.show()
```

Цель: оценить форму связи (линейная/нелинейная), наличие выбросов.

## Где полезно:

- Быстрый отбор признаков (feature screening) перед моделированием.
- Диагностика мультиколлинеарности (высокие межпризнаковые корреляции).

## Предосторожности:

- Pearson отражает **линейную** связь; Spearman — **монотонную**.
- Выбросы и неоднородная дисперсия искажают  $r$  (используйте визуализацию).
- Корреляция  $\neq$  причинность.
- Множественные проверки  $\Rightarrow$  корректировки (Bonferroni/FDR).



- Корреляция — простая и наглядная мера связи.
- Python (pandas, scipy) даёт мгновенный расчёт  $r$ , p-value и CI.
- Визуализация обязательна для корректной интерпретации.

Код демо: ноутбук `correlation_demo.ipynb`.

Пакеты: `pandas`, `numpy`, `scipy`, `matplotlib`, `scikit-learn`.

Курс: «Методы статистической обработки информации».