



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

Введение

- **Цель:** показать, как выполнить корреляционный анализ в Python и корректно интерпретировать результаты для принятия решений на основе данных
- **Показатели:**
 - ▶ коэффициент Пирсона: для линейной связи
 - ▶ коэффициент Спирмена: для монотонной связи и в ситуациях с выбросами

Постановка задачи и данные

- **Задача:**

- ▶ посчитать значения корреляций
- ▶ проверить их статистическую значимость
- ▶ построить 95% доверительные интервалы
- ▶ визуализировать результаты матрицей и scatter-графиками
- ▶ выделить признаки с наиболее устойчивой связью

- **Данные:** датасет из 442 наблюдений и 10 признаков из

```
from sklearn.datasets import load_diabetes
data = load_diabetes()
X, y = data.data, data.target
```

Python

Язык: Python 3.x:

- высокоуровневый интерпретируемый язык программирования, используемый в статистическом анализе из-за простоты синтаксиса и огромного количества библиотек для работы с данными

Библиотеки: pandas, numpy, scipy, matplotlib.

Почему Python?

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

Где запускать: Jupyter Notebook/Colab/VS Code

Выбор среды разработки: почему Google Colab

Критерий	Colab	Jupyter Notebook	VS Code
Установка	Не требуется (облако)	Требуется локально	Требуется локально
Мощность ресурсов	GPU/TPU (ограниченно)	Зависит от ПК	Зависит от ПК
Портативность	Максимальная	Средняя	Средняя
Совместимость	Браузер	Локальная конфигурация	Расширения/настройка
Работа с данными	Лёгкий импорт из Drive	Локальные файлы	Проекты/репозитории
Стоимость	Бесплатно	Бесплатно	Бесплатно

Почему выбрал Google Colab:

- не требует установки — работает в браузере на любом устройстве
- бесплатных ресурсов достаточно для задач статистики
- легко подключать данные (Google Drive, GitHub)
- позволяет открывать блокнот везде — дома, в университете, на работе

Структура демо

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

Корреляции: виды зависимости

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

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

Коэффициент Пирсона

Коэффициент Пирсона - показывает, насколько две переменные изменяются вместе по прямой (линейно).

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:

- не меняется, если данные смещать или умножать на константу
- $\rho = \pm 1 \Leftrightarrow Y = \beta_0 + \beta_1 X$

Корреляция: показывает связь, но не объясняет причину

Коэффициент Спирмена

Коэффициент Спирмена измеряет монотонность связи двух переменных Основан на взаимном ранжировании, а не на абсолютных значениях

Формула (для данных без совпадающих рангов):

$$\rho_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

где d_i — разность рангов.

Pearson vs. Spearman

	Pearson r	Spearman ρ_s
Связь	Линейная	Монотонная (по рангам)
Выбросы	Чувствителен	Робастность выше
Данные	Интервальные/отношений	Порядковые/монотонные

Данные для демонстрации

Набор данных: `sklearn.datasets.load_diabetes()`

Медицинский датасет из 442 наблюдений с непрерывными признаками:

- возраст,
- индекс массы тела (ИМТ),
- артериальное давление (АД),
- показатели крови и др.

Целевая переменная `target` отражает **прогресс заболевания**.

Набор удобен для:

- анализа корреляций,
- построения матрицы корреляций,
- парных scatter-графиков,
- проверки статистической значимости зависимостей.

Используемые библиотеки

- **NumPy:** базовая библиотека для работы с массивами и матрицами, быстрые численные операции
- **Pandas:** удобная работа с табличными данными (датафреймы, очистка, агрегации, описательная статистика)
- **Matplotlib:** построение графиков (линий, точек, гистограмм, матриц корреляций)
- **SciPy:** функции статистики (p-value, критерии значимости, интервалы, распределения)
- **Scikit-learn:** загрузка датасетов, предобработка, метрики, простые модели машинного обучения

Импорт и подготовка данных

```
import numpy as np
import pandas as pd
import 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()
df.describe()
```

Матрица корреляций (Pearson)

```
corr_pearson = df.corr(numeric_only=True)
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, rotation=90)
plt.yticks(range(len(corr_pearson.index)), corr_pearson.index)
plt.tight_layout(); plt.show()
```

Интерпретация: пары с $|r| \gtrsim 0.3\text{--}0.4$ — 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"Spearmen rho(bmi, target) = {r_s:.3f}, p-value = {p_s:.3g}")
```

Доверительный интервал для ρ (Fisher z-transformation)

```
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-график

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

Цель: определить вид связи (линейная/нелинейная), наличие выбросов

Итоги

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