ИНТЕЛЛЕКТУАЛЬНОЕ МОДЕЛИРОВАНИЕ

Образовательная программа «6B06103 – Компьютерная инженерия»

Лекция 5

# Визуализация данных средствами matplotlib

Неделя 5 · Tan, Steinbach, Kumar: глава 3.3

# Цель и план лекции

**Цель:** выбирать подходящий тип графика и строить его в matplotlib

1. Зачем визуализировать данные
2. Принципы: отображение, упорядочивание, отбор
3. Типы графиков и их выбор
4. Графики в matplotlib
5. Правила хорошей визуализации

# Зачем визуализировать данные

- **Исследование** — увидеть распределения, выбросы, связи
- **Представление** — донести вывод до аудитории
- Карта температуры океана из учебника: ≈ 250 000 чисел понятны за секунды

**Квартет Энскомба:** четыре набора с одинаковыми средними, дисперсиями и корреляцией — и совершенно разными графиками

# Три принципа (Tan, 3.3.2)

- **Отображение** — признаки → положение, размер, цвет, форма. Номинальный признак не кодируют оттенками одного цвета: появится ложный порядок
- **Упорядочивание** — перестановка строк и столбцов или сортировка выявляет структуру
- **Отбор** — показываем пары признаков, выборку объектов, прозрачность

# Какой график выбрать

| Задача | График |
|---|---|
| Изменение во времени | Линейный |
| Сравнить категории | Столбчатая диаграмма |
| Доли целого | Круговая (до 5–6 долей) |
| Распределение признака | Гистограмма, ECDF, ящик с усами |
| Связь двух признаков | Диаграмма рассеяния |
| Матрица значений | Тепловая карта |

# Библиотека matplotlib

```
import matplotlib.pyplot as plt
```

| Элемент | Функция |
|---|---|
| Фигура, размер | plt.figure(figsize=(8, 5)) |
| Заголовок | plt.title() |
| Подписи осей | plt.xlabel(), plt.ylabel() |
| Легенда, сетка | plt.legend(), plt.grid(True) |
| Показать / сохранить | plt.show(), plt.savefig() |

# Линейный график

```
months = ['Янв', 'Фев', 'Мар', 'Апр', 'Май']
almaty = [-5, -3, 4, 12, 17]
astana = [-15, -13, -6, 6, 13]
plt.plot(months, almaty, marker='o', label='Алматы')
plt.plot(months, astana, linestyle='--', label='Астана')
plt.title('Средняя температура, °C')
plt.legend(); plt.grid(True); plt.show()
```

Параметры: color, linestyle, marker, linewidth, label

# Столбчатая и круговая диаграммы

```
plt.bar(['A', 'B', 'C', 'D'], [15, 30, 45, 20])
plt.barh(...)      # горизонтальные столбцы

plt.pie([40, 30, 15, 15],
        labels=['Жильё', 'Питание', 'Транспорт', 'Прочее'],
        autopct='%1.1f%%')
```

Высоты столбцов сравнивать легче, чем углы секторов

# Гистограмма и ECDF

```
data = rng.normal(loc=170, scale=8, size=1000)
plt.hist(data, bins=20, edgecolor='black')


x = np.sort(data)
y = np.arange(1, len(x) + 1) / len(x)
plt.step(x, y)    # эмпирическая функция распределения
```

- Мало бинов скрывает детали, много — даёт «шум»
- По ECDF сразу видна медиана: где кривая = 0,5

# Диаграмма рассеяния: Iris

```
iris = load_iris()
X, y = iris.data, iris.target
for code, name in enumerate(iris.target_names):
    mask = y == code
    plt.scatter(X[mask, 2], X[mask, 3], label=name)
plt.xlabel('Длина лепестка, см'); plt.legend()
```

Setosa отделён от других видов, versicolor и virginica перекрываются

# Ящик с усами

```
plt.boxplot(data, orientation='horizontal')
```

- Линия внутри — **медиана**
- Границы ящика — **Q1 и Q3**
- Усы — до 1,5·IQR от ящика
- Точки за усами — **выбросы** (тот же критерий IQR)

# Несколько графиков на фигуре

```
fig, axes = plt.subplots(1, 3, figsize=(12, 4))
axes[0].hist(X[:, 2], bins=20)
axes[1].boxplot([X[y == k, 2] for k in range(3)])
axes[2].scatter(X[:, 0], X[:, 2], c=y)
fig.tight_layout()
fig.savefig('iris.png', dpi=150)
```

Объектно-ориентированный стиль: явная фигура fig и оси ax

# Принципы ACCENT

| Принцип | Вопрос к графику |
|---|---|
| Apprehension | Видны ли связи между переменными? |
| Clarity | Выделено ли главное? |
| Consistency | Согласованы ли цвета и обозначения? |
| Efficiency | Изображено ли сложное просто? |
| Necessity | Нужен ли график и каждый элемент? |
| Truthfulness | Верен ли масштаб? |

# Практические правила

- Подписывайте оси с единицами и давайте заголовок
- Ось столбчатой диаграммы начинайте с нуля
- Не используйте 3D для двумерных данных
- Палитры, различимые при нарушениях цветового зрения (viridis)
- Несколько простых графиков лучше одного перегруженного

Тафти: больше идей за меньшее время, меньше «чернил», правда о данных

# Выводы

- Визуализация нужна и для исследования, и для представления
- Тип графика выбирается по данным и задаче
- matplotlib строит любой из них несколькими строками кода
- Хороший график ясен, экономен и правдив

Далее: разведочный анализ с pandas и seaborn

# Контрольные вопросы

1. Какие две задачи решает визуализация? Что показывает квартет Энскомба?
2. Объясните отображение, упорядочивание и отбор.
3. Чем гистограмма отличается от столбчатой диаграммы?
4. Сравните функциональный и объектно-ориентированный стили matplotlib.
5. Постройте фигуру: линейный график с двумя сериями и ящики с усами для двух групп.
6. Оцените один из графиков лекции по принципам ACCENT.

# Литература


1. Tan P.-N., Steinbach M., Kumar V. Introduction to Data Mining. — Pearson, 2014. — 3.3. Visualization.
2. VanderPlas J. Python Data Science Handbook. — O’Reilly, 2016. — Chapter 4.
3. McKinney W. Python for Data Analysis. — O’Reilly. — Chapter 9.1.
4. Tufte E. R. The Visual Display of Quantitative Information. — 2001.
5. Документация matplotlib: matplotlib.org