ИНТЕЛЛЕКТУАЛЬНОЕ МОДЕЛИРОВАНИЕ

Образовательная программа «6B06103 – Компьютерная инженерия»

Лекция 3

Обработка данных в pandas: Series, DataFrame, индексация

Неделя 3

Цель и план лекции

Цель: научиться создавать, читать, выбирать и преобразовывать табличные данные в pandas

1. Series и DataFrame
2. Чтение файлов и первичный просмотр
3. Выбор столбцов, loc и iloc
4. Выбор строк по условию
5. Новые столбцы, группировка, объединение таблиц

Библиотека pandas

- Основной инструмент обработки табличных данных в Python
- Создана Уэсом Маккинни в 2008 году
- Построена на NumPy: добавляет подписи строк и столбцов, разные типы, пропуски
- Строка таблицы — **объект**, столбец — **признак**

```
import pandas as pd
```

Series — столбец с метками

```
s = pd.Series([10, 20, 30],
              index=['a', 'b', 'c'])
s['a']        # 10
s * 2         # векторная операция
s[s > 15]    # фильтр
```

dtype	Содержимое	Тип признака
int64, float64	Числа (NaN — пропуск)	Количественный
category	Набор категорий	Номинальный, порядковый
datetime64	Дата и время	Интервальный

DataFrame — таблица

```
df = pd.DataFrame({
    'name': ['Perizat', 'Kuan', 'Doniyor', 'Klara'],
    'gender': [1, 2, 2, 1],
    'age': [23, 21, 21, 25]})
```

- Строки с **индексом** 0, 1, 2, 3; столбцы с **именами**
- В столбце — один тип, в разных столбцах — разные

Чтение и запись файлов

```
df = pd.read_csv('test.csv', sep=';')
df = pd.read_excel('data.xlsx')
df.to_csv('out.csv', index=False)
```

- **sep** — разделитель (в русском Excel — «;»)
- **encoding** — 'utf-8' или 'cp1251'
- **index_col**, **na_values** — индекс и обозначения пропусков

Первичный просмотр данных

Команда	Что показывает
df.head()	Первые 5 строк
df.shape	Размер: (4, 3)
df.info()	Типы столбцов и пропуски
df.describe()	Статистика числовых столбцов
df.columns	Имена столбцов
df.index	Метки строк

Выбор столбцов

```
df['name']           # Series
df.name              # то же, короче
df[['name', 'age']] # DataFrame
```

- Двойные скобки — список имён → таблица
- Через точку нельзя, если в имени пробел или оно совпадает с методом

loc — доступ по меткам

```
df.loc[1]                          # строка с меткой 1
df.loc[0:2]                        # метки 0, 1 и 2
df.loc[0:2, 'name':'gender']
df.loc[[0, 3], ['name', 'age']]
df.loc[0, 'age'] = 24              # изменение
```

Срез в loc включает правую границу

iloc — доступ по позициям

```
df.iloc[0:2]              # позиции 0 и 1
df.iloc[0:2, 0:2]
df.iloc[[0, 2], [0, 2]]
df.iloc[-1]               # последняя строка
df.iloc[0, 2]             # одна ячейка
```

Срез в iloc НЕ включает правую границу

loc и iloc: сравнение

Свойство	loc	iloc
Основа выбора	Метки, имена	Позиции 0, 1, 2...
Правая граница	Включается	Не включается
df....[0:2]	3 строки	2 строки
По условию	Да	Нет

После сортировки по возрасту метки ≠ позиции: **loc[0]** → Perizat, **iloc[0]** → Kuan

Выбор строк по условию

```
df[df['age'] > 21]
df.loc[df['gender'] == 1, ['name', 'age']]
df.loc[(df['age'] ⩾ 21) & (df['gender'] == 2)]
df[df['name'].isin(['Kuan', 'Klara'])]
df.query('age > 21 and gender == 1')
```

- Операторы: **&** (и), **|** (или), **~** (не)
- Каждое условие — в круглых скобках

Столбцы, сортировка, группировка

```
df['age5'] = df['age'] + 5     # новый признак
df = df.drop('age5', axis=1)  # удалить столбец
df.sort_values('age', ascending=False)
df.groupby('gender')['age'].mean()
```

- drop возвращает новую таблицу — присвойте результат
- groupby = **агрегация** из учебника Tan: меньше данных, устойчивее значения, но теряются детали

Объединение таблиц

```
pd.concat([df1, df2], ignore_index=True)
pd.merge(students, grades, on='id')
pd.merge(students, grades, on='id', how='left')
```

- **concat** — «склеить» таблицы одну под другой
- **merge** — соединить по ключу, как JOIN в SQL
- how: inner (совпадающие ключи), left, right, outer; нет пары → NaN

Выводы

- Series — столбец, DataFrame — таблица объектов и признаков
- Столбцы — по именам, строки и ячейки — loc (метки) и iloc (позиции)
- Срез loc включает правую границу, iloc — нет
- Фильтры, groupby и merge — основа любого анализа

Далее: описательная статистика и предобработка

Контрольные вопросы

1. Объясните назначение pandas и связь с NumPy; сравните Series и DataFrame.
2. Создайте DataFrame о пяти студентах и выведите размер, столбцы и типы.
3. Чем отличаются df['name'] и df[['name']]?
4. Приведите пример, когда loc и iloc дают разный результат.
5. Тремя способами получите возраст Klara; выберите имена и возраст юношей старше 20 лет.
6. Сравните concat и merge, слияния inner и left.

Литература

1. McKinney W. Python for Data Analysis. — O’Reilly. — Chapters 5, 8.
2. VanderPlas J. Python Data Science Handbook. — O’Reilly, 2016. — Chapter 3.
3. Tan P.-N., Steinbach M., Kumar V. Introduction to Data Mining. — Pearson, 2014. — 2.1.2, 2.3.1.
4. Документация pandas: Indexing and selecting data.
5. Набор данных Titanic (datasciencedojo, GitHub).