ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Лекция 4

Агрегирование и сводные таблицы

Оспан Ә.Ғ., PhD, доцент-исследователь

КазНУ, кафедра ИИ и Big Data

2026-2027 учебный год

Цель и содержание лекции

Цель и результаты обучения

- Выбирать сумму, среднее, медиану и счетчик по смыслу задачи.
- Создавать именованные агрегаты и сводные по двум ключам.
- Проверять пустые сочетания, веса и итоговые значения.

Содержание лекции

1. **Выбор агрегата**
2. **agg**
3. **Уровень строки**
4. **pivot_table**
5. **Итоги**

Разные вопросы требуют разных агрегатов

Вопрос	Агрегат	Что учитывать
Каков общий объем?	sum	Зависит от числа записей
Каков средний уровень?	mean	Чувствителен к большим значениям
Где середина значений?	median	Не показывает весь разброс
Каковы границы?	min, max	Могут определяться одной записью
Сколько наблюдений?	size / count	Строки либо непустые значения

Среднее: уровень и чувствительность

$$\text{mean} = \text{sum}(x_i) / n$$

Содержание

- Использует все значения и их величину.
- Сохраняет единицу исходного показателя.

Ограничение

- Большое значение способно заметно сдвинуть mean.
- Без распределения среднее не описывает типичную запись полностью.

Wage: 10, 10, 100. Сумма = 120; среднее = 40.

Медиана: центральное положение без равенства значений

Как вычисляется

- Сначала значения упорядочиваются.
- Нечетное n: центральное значение.
- Четное n: среднее двух центральных.

Как интерпретируется

- Менее чувствительна к отдельным крайним значениям.
- mean = median не означает равенство всех наблюдений.
- Сравнение дополняем n, min и max.

Центральное положение, разброс и размер группы описывают разные стороны данных.

Именованные агрегаты в одном отчете

Пример Python

```
import pandas as pd

demo = pd.DataFrame({
    'Club': ['A', 'A', 'A', 'B', 'B'],
    'Position': ['GK', 'ST', 'ST', 'GK', 'CM'],
    'Wage': [10, 10, 100, 20, 40],
    'SprintSpeed': [40, 70, 90, 50, 80]
})
summary = demo.groupby('Club', as_index=False).agg(
    n=('Wage', 'size'), wage_mean=('Wage', 'mean'),
    wage_median=('Wage', 'median'), wage_sum=('Wage', 'sum')
)
print(summary)
```

Контракт отчета

- Одна строка соответствует клубу.
- Названия полей задаются явно.

Проверка

- A: n=3; mean=40; median=10; sum=120.
- B: n=2; mean=30; median=30; sum=60.

Равенство среднего и медианы: что оно не доказывает

Значения	mean	median	Вывод
30	30	30	Одна запись: совпадение неизбежно
20, 40	30	30	Значения различны при совпавших статистиках
10, 10, 100	40	10	Большое значение сдвинуло среднее

Совпадение двух статистик не определяет распределение; численность и диапазон проверяются отдельно для каждой выбранной группы.

Уровень строки должен сохраняться в расчетах

1

Исходная строка

Запись о характеристиках игрока.

2

Один ключ

Club: одна строка на представленный клуб.

3

Два ключа

Club и Position: одна строка на наблюдаемую пару.

4

Сводная форма

Те же пары распределены по двум осям таблицы.

- as_index=False дает плоские поля, удобные для экспорта.
- Нельзя суммировать повторенные итоги разных уровней детализации.

pivot_table: два ключа и показатель в ячейке

pivot_table

- index и columns задают ключи двух осей.
- values выбирает показатель; aggfunc задает расчет.
- Повторяющиеся пары агрегируются.

pivot

- Меняет форму без агрегирования.
- Пара ключей должна задавать одно значение.
- Дубли пар требуют решения по смыслу данных.

Выбор между pivot и pivot_table зависит от уникальности пар и вопроса агрегации.

Сводная численностей по клубу и позиции

Пример Python

```
counts = demo.pivot_table(
    index='Club', columns='Position', values='Wage',
    aggfunc='size', fill_value=0, observed=True
)
print(counts)
assert counts.to_numpy().sum() == len(demo)
```

Результат: A: CM=0, GK=1, ST=2. B: CM=1, GK=1, ST=0.

Две оси

- Строки: Club; столбцы: Position.
- Ячейка: число исходных записей.

Контроль

- Ноль означает отсутствие записей пары.
- Сумма счетчиков равна числу исходных строк.

Пустая ячейка: ноль или отсутствие оценки?

Показатель	Нет строк сочетания	Подход к представлению
Число записей	Известный нулевой счетчик	Ноль уместен
Средний SprintSpeed	Оценка не определена	Сохранить пропуск
Сумма Wage	Нужно знать полноту данных	Ноль только по объявленному правилу
Неизвестное измерение	Значение не наблюдалось	Не подменять измеренным нулем

margins: общий итог среднего

Пример Python

```
mean_total = demo.pivot_table(
    index='Club', columns='Position', values='SprintSpeed',
    aggfunc='mean', margins=True, margins_name='Total',
    observed=True
)
print(mean_total.loc['Total', 'Total'])
print(demo['SprintSpeed'].mean())
```

Результат: 66.0; 66.0

Средний рейтинг

- Итог рассчитывается по исходным наблюдениям.
- Размеры групп учитываются автоматически.

Повторный счет

- Total уже содержит агрегированный объем.
- Не суммируем итоговую строку повторно.

Среднее по игрокам и среднее по клубам

mean_players = sum(n_g * mean_g) / sum(n_g)

По наблюдениям

- Вес клуба равен числу известных значений.
- Совпадает с прямым mean исходных строк позиции.

По клубам

- Все клубные средние имеют одинаковый вес.
- Отвечает другому аналитическому вопросу.

Веса 3 и 2: (3*66.6667 + 2*65)/5 = 66.

Отчет по клубам исторического football.csv

Пример Python

```
from pathlib import Path
import numpy as np

football = pd.read_csv(Path('data') / 'football.csv')
club_report = football.groupby('Club', as_index=False).agg(
    n=('Wage', 'size'), wage_mean=('Wage', 'mean'),
    wage_median=('Wage', 'median')
)
club_report['same'] = np.isclose(
    club_report['wage_mean'], club_report['wage_median'],
    rtol=1e-10, atol=1e-8
)
print(club_report.head(3))
```

Несколько показателей

- Один ключ Club, три понятных агрегата.
- Численность нужна для интерпретации.

Равенство

- Допуск задан явно.
- Совпадение mean/median не означает одинаковый Wage.

Проверки сводной: структура, суммы и смысл

Структура

- Одна строка на объявленную группу.
- Ключи и названия полей однозначны.

Числа

- Счетчики сходятся с исходником.
- Ячейка совпадает с прямой фильтрацией.

Интерпретация

- Нули и пропуски имеют разный смысл.
- Веса и итоговые строки не создают повторный счет.

Красивое представление не заменяет контроль структуры, исходных строк и смысла показателя.

Пять правил агрегирования

1. Агрегат выбирается по вопросу, а не только по типу данных.
2. Именованная agg делает схему отчета явной.
3. pivot_table агрегирует повторяющиеся сочетания ключей.
4. Пустой счетчик можно представить нулем; пустое среднее не определено.
5. Общие средние требуют правильных весов, а итоги не считают повторно.

Проверьте сводную и ее итоги

1. Почему сумма и среднее Wage могут дать разные ранжирования?
2. Что не доказывает равенство mean и median?
3. Когда нужен pivot_table вместо pivot?
4. Почему ноль допустим для пустого счетчика, но опасен для среднего?
5. Как восстановить общее среднее из групповых средних?

Самостоятельная работа

На demo получите суммы Wage и средние SprintSpeed через pivot_table и groupby с unstack. Сравните ячейки и объясните правила пустых сочетаний.

Источники по агрегатам и сводным

Документация

1. pandas: агрегирование по группам

https://pandas.pydata.org/docs/user_guide/groupby.html

2. pandas: изменение формы и сводные

https://pandas.pydata.org/docs/user_guide/reshaping.html

3. pandas: pivot_table

https://pandas.pydata.org/docs/reference/api/pandas.pivot_table.html

Основа лекции

- Лекция 4 Группировака данных. Продолжение.pptx
- Лабораторная работа 4 .docx
- Лабораторная работа 4 с ответами.docx