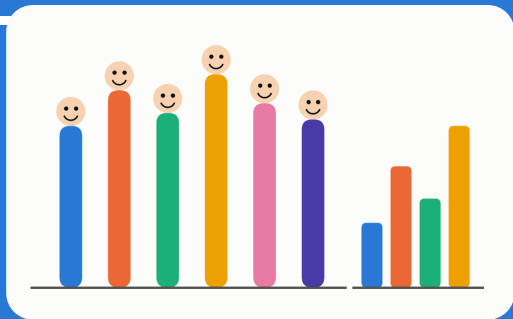


Статистика это просто!

Учимся понимать числа на
примере нашего класса: рост,
конфеты, любимые фрукты и
мороженое



Что мы сегодня узнаем

- 1 **Центр:** среднее, медиана, мода
- 2 **Разброс:** размах, дисперсия, std
- 3 **Процентили и квартили**
- 4 **Форма:** асимметрия и эксцесс
- 5 **Связь:** корреляция и ковариация
- 6 **pandas:** describe()
- 7 **pandas:** value_counts() и unique()
- ? Викторина в конце!

Главная идея

Статистика помогает **описать много чисел парой понятных слов**: «какие мы обычно?» и «насколько мы разные?»

Наш класс — 15 ребят

```
import pandas as pd
df = pd.read_csv("klass.csv")
df.head()
```

	имя	рост	конфеты	фрукт	оценка	телефон_мин
0	Аня	128	3	яблоко	5	60
1	Боря	135	5	банан	4	90
2	Вика	131	2	яблоко	5	45
3	Гоша	140	8	груша	3	150
4	Даша	126	4	банан	4	70

- **рост** — в сантиметрах
- **конфеты** — сколько съел за день
- **фрукт** — любимый фрукт
- **оценка** — за контрольную
- **телефон_мин** — минут в телефоне

Числа и слова

Рост и конфеты — **числа**.
Фрукт — **слово** (категория).

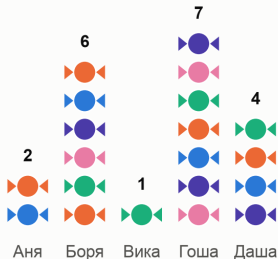
1

Центр: среднее, медиана,
мода

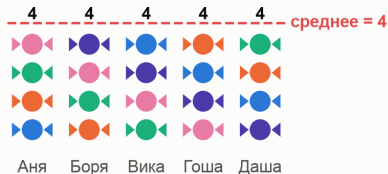


Среднее (mean) — «поделим поровну»

Было: у всех по-разному



Стало: поделили поровну



Как считать

Сложить всё и разделить на количество:

$$\frac{2 + 6 + 1 + 7 + 4}{5} = \frac{20}{5} = 4$$

Если собрать все конфеты в одну кучу и раздать поровну — у каждого будет **среднее**.

Медиана (median) — «тот, кто в середине»

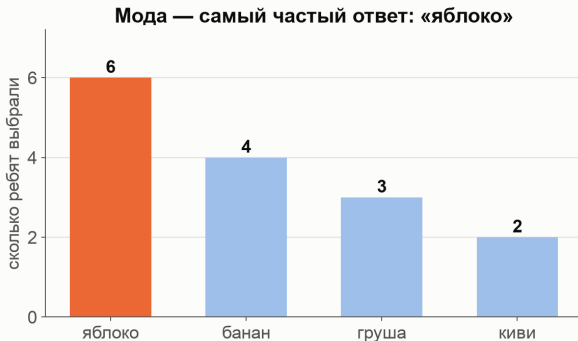


Как найти

- 1) Выстроить всех по порядку.
- 2) Взять того, кто **ровно посередине**.

Если ребят **чётное** число — в середине стоят двое. Тогда медиана = их среднее:
 $(131 + 133) / 2 = 132$.

Мода (mode) — «самый популярный»



Мода

Значение, которое встречается чаще всех.

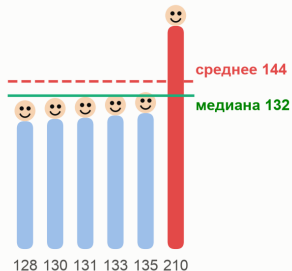
- Работает даже со **словами**: среднего фрукта не бывает, а самый любимый — бывает!
- Мод может быть **несколько**.
- В нашем классе рост у всех разный — значит, `df["рост"].mode()` вернёт **все 15** значений.

Опасность: великан ломает среднее!

5 ребят



...и пришёл баскетболист



Выброс (outlier)

Одно очень большое число **сильно двигает среднее** ($131 \rightarrow 144$), а **медиана почти не меняется** ($131 \rightarrow 132$). Поэтому для зарплат, цен на дома и карманных денег часто смотрят на **медиану**.

Центр в pandas

```
s = df["конфеты"]  
  
s.mean()      # среднее  
s.median()    # медиана  
s.mode()      # мода (список!)
```

```
4.67  
5.0  
0    5
```

Мера	Смысл	Боится выбросов
mean	поровну	да
median	середина	нет
mode	популярное	нет

Запомни

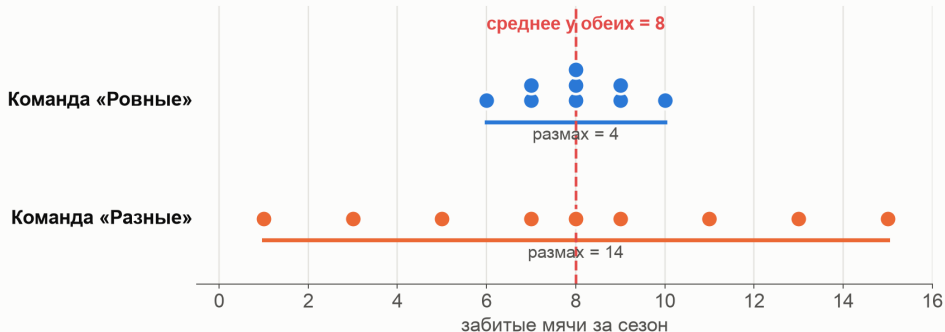
Если среднее и медиана **сильно отличаются** — в данных есть выбросы или длинный «хвост».

2

**Разброс: размах, дисперсия,
стандартное отклонение**



Одинаковое среднее — разные команды



Вопрос

У обеих команд в среднем 8 мячей. Но на какую можно **надеяться** в каждой игре? Среднего мало — нужно знать, **насколько числа разбросаны**.

Размах (range) — от min до max

Формула

$$\text{range} = \text{max} - \text{min}$$

- «Ровные»: $10 - 6 = 4$
- «Разные»: $15 - 1 = 14$
- Наш класс по росту: $142 - 126 = 16$ см

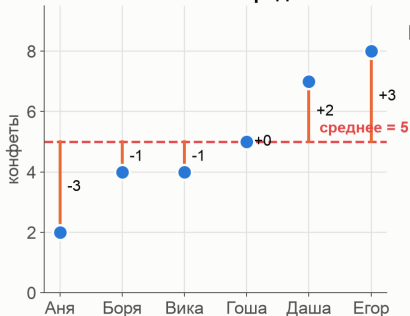
Минус размаха

Смотрит **только на двоих**: самого низкого и самого высокого. Один великан — и размах огромный.

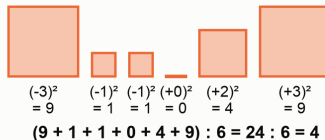
В pandas: `s.max() - s.min()`

Дисперсия (variance) — шаг за шагом

Шаг 1: на сколько каждый отличается от среднего



Шаг 2: возводим в квадрат и берём среднее
дисперсия = 4, $\text{std} = \sqrt{4} = 2$



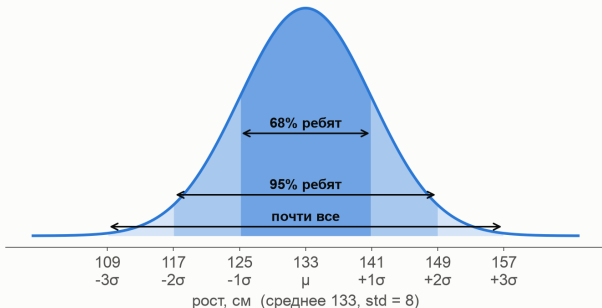
Зачем квадрат? Плюсы и минусы иначе съедят друг друга:

$$-3 - 1 - 1 + 0 + 2 + 3 = 0.$$

Дисперсия = среднее квадратов отклонений. Чем она больше — тем сильнее разброс.

Стандартное отклонение (std) — «обычный шаг»

Стандартное отклонение σ — «обычный шаг» от среднего



$$\text{std} = \sqrt{\text{дисперсия}}$$

Дисперсия в «квадратных сантиметрах» — странно! Корень возвращает **обычные единицы**.

- Большинство ребят — в пределах **одного** std от среднего.
- Почти все — в пределах **трёх** std.
- Кто дальше 3 std — кандидат в выбросы.

```
s = df["рост"]  
  
s.max() - s.min()    # размах  
s.var()              # дисперсия  
s.std()              # станд. отклонение
```

```
16  
24.52  
4.95
```

Маленький секрет pandas

`var()` и `std()` делят не на n , а на $n - 1$ (`ddof=1`). Для конфет на слайде выше: $24 : 6 = 4$, а pandas даст $24 : 5 = 4.8$. Так честнее, когда у нас лишь **часть** всех детей.

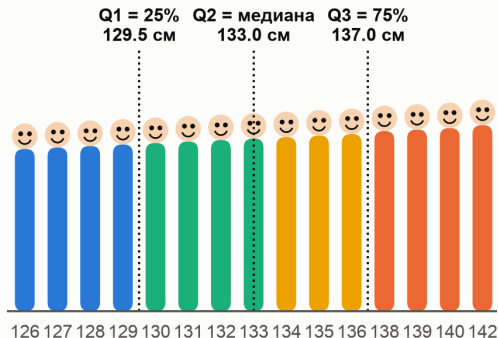
Итог: рост в классе — 133 ± 5 см.

3

Проценти́ли и кварта́ли



Квартили — делим класс на 4 команды



Q1 (25%): четверть ребят
ниже 129.5 см

Q2 (50%): это и есть
медиана

Q3 (75%): три четверти
ниже 137 см

Процентиль — «выше, чем X% ребят»

Пример из жизни

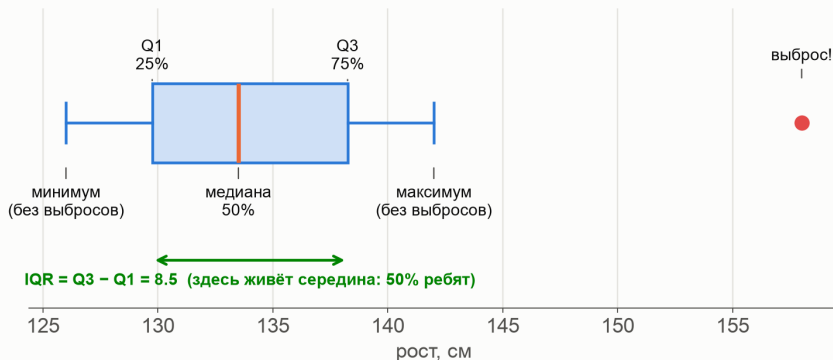
Врач говорит: «Рост на **90-м процентиле**». Это значит: ребёнок выше, чем **90 из 100** сверстников.

- 25-й процентиль = Q1
- 50-й процентиль = Q2 = медиана
- 75-й процентиль = Q3

```
s = df["рост"]  
s.quantile(0.25)  
s.quantile([0.25, 0.5, 0.75])  
s.quantile(0.9)    # 90-й
```

```
129.5  
0.25    129.5  
0.50    133.0  
0.75    137.0  
139.6
```

Ящик с усами (boxplot)



Ящик — средние 50% ребят. **Черта** внутри — медиана. **Усы** — до самых крайних «нормальных» значений.

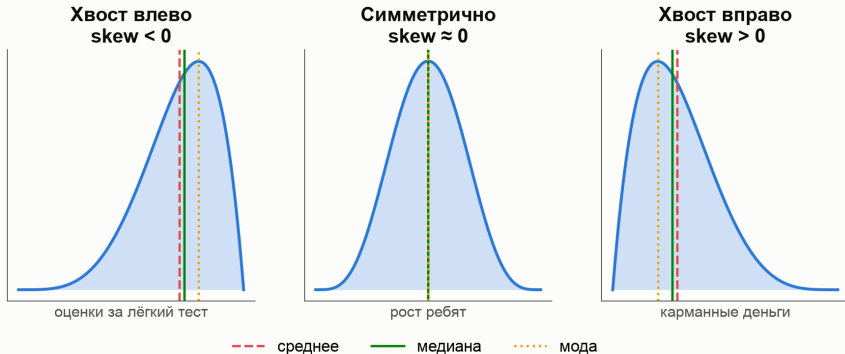
Выброс: дальше, чем $1.5 \cdot IQR$ от ящика.
Нарисовать: `df["рост"].plot.box()`

4

Форма: асимметрия и
эксцесс



Асимметрия (skewness) — куда тянется хвост?

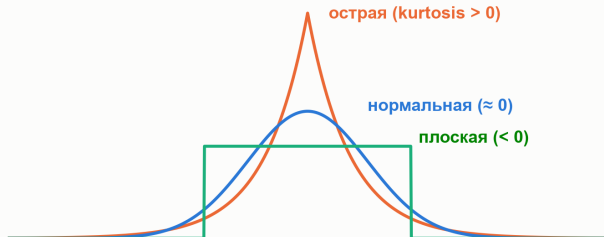


$skew < 0$: почти все получили 5, но несколько — двойки. Среднее **левее** медианы.

$skew \approx 0$: «горка» ровная, $mean = median = mode$.

$skew > 0$: у большинства мало денег, у пары ребят — очень много. Среднее **правее**.

Эксцесс (kurtosis) — острая или плоская горка?



- +** **Острая**: большинство у центра, но бывают **редкие сюрпризы** далеко.
- 0** **Нормальная**: обычный колокол.
- **Плоская**: все значения встречаются примерно одинаково (как кубик).

Простое правило

Большой kurtosis = «ждите выбросов».

Форма в pandas

```
cols = ["рост", "конфеты",  
        "оценка", "телефон_мин"]  
df[cols].skew()  
df[cols].kurt()
```

```
skew  
рост      0.22  
конфеты   0.40  
оценка   -0.23  
телефон_мин 0.72
```

```
kurt  
рост      -1.03  
конфеты   -0.14  
оценка   -0.97  
телефон_мин -0.28
```

Читаем

Телефон: skew = 0.72 — хвост вправо: несколько ребят сидят в телефоне очень долго.

Рост: kurt = -1.03 — горка плоская, рост распределён ровно.

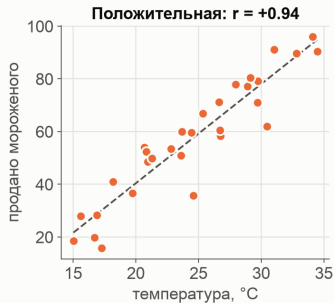
Как понимать число: от -0.5 до 0.5 — почти симметрично; больше 1 по модулю — сильный перекося.

5

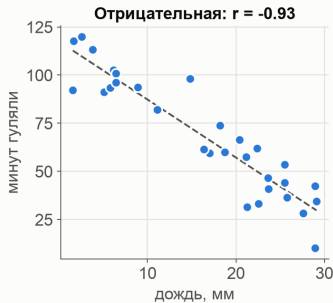
Связь: корреляция и ковариация



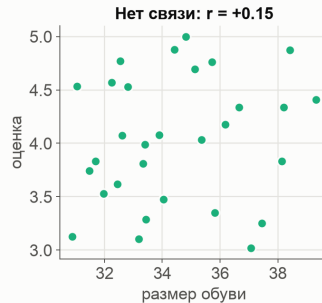
Корреляция — «меняются ли числа вместе?»



Жарче $\Rightarrow$ **больше**
мороженого

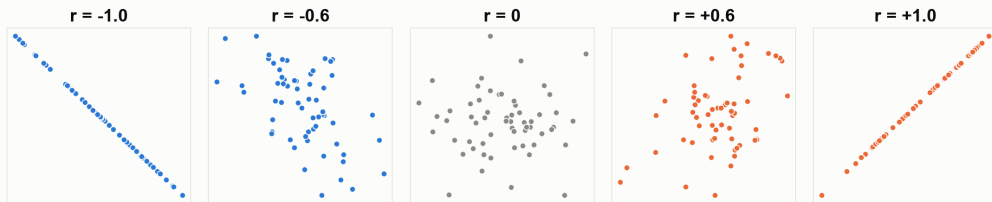


Больше дожда $\Rightarrow$
меньше гуляем



Размер обуви не влияет
на оценки

Корреляция r : от -1 до $+1$



r близко к $+1$

Одно растёт — другое тоже растёт.

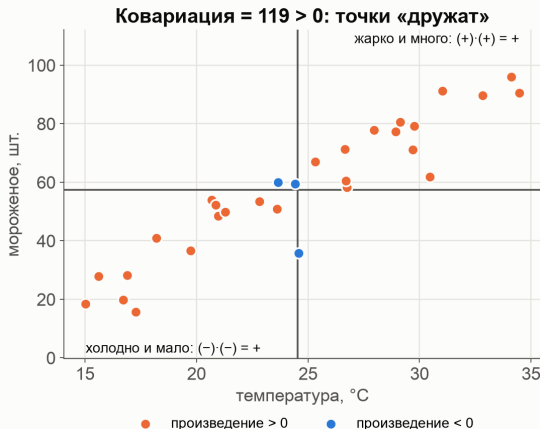
r близко к 0

Связи по прямой нет — облако точек.

r близко к -1

Одно растёт — другое падает.

Ковариация — откуда берётся знак



Для каждой точки перемножаем:
(температура – среднее) ×
(мороженое – среднее)

- Почти все произведения **плюсовые** ⇒ ковариация > 0
- Почти все **минусовые** ⇒ ковариация < 0

Минус ковариации

Её величина зависит от единиц (см, мм, штуки). Поэтому её «сжимают» в r :

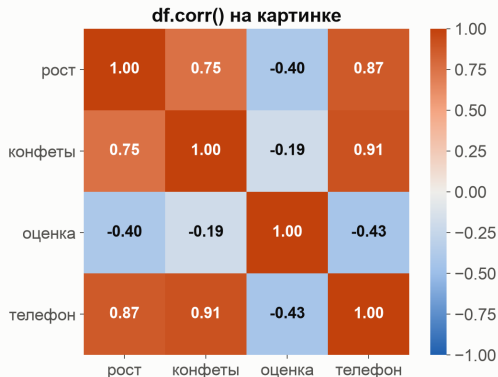
$$r = \frac{\text{cov}(x, y)}{\text{std}(x) \cdot \text{std}(y)}$$

Корреляция и ковариация в pandas

```
cols = ["рост", "конфеты",  
        "оценка", "телефон_мин"]  
df[cols].corr()  # таблица r  
df[cols].cov()   # ковариации  
  
df["рост"].corr(df["конфеты"])
```

0.75

Таблица симметрична, на диагонали всегда 1.



Осторожно: связь $\neq$ причина!

Ловушка

В нашем классе: чем больше **конфет**, тем больше **минут в телефоне** ($r = 0.91$).

Значит ли это, что конфеты заставляют сидеть в телефоне? **Нет!**

Возможные причины:

- есть **третий фактор** (например, ребёнок старше: выше, ест больше и дольше в телефоне);
- просто **совпадение** — у нас всего 15 ребят.

Пример

Летом растут продажи мороженого **и** число солнечных ожогов. Мороженое не обжигает — виновато **солнце**.

6

Глубокий анализ `describe()`



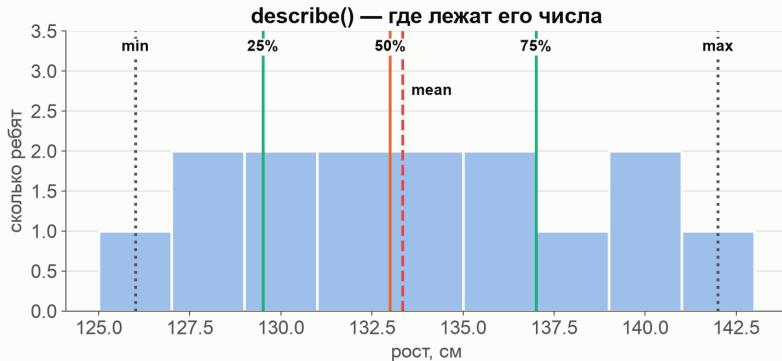
describe() — вся статистика одной командой

```
df.describe().round(2)
```

	рост	конфеты	оценка	телефон_мин
count	15.00	15.00	15.00	15.00
mean	133.33	4.67	4.13	83.00
std	4.95	2.19	0.74	33.58
min	126.00	1.00	3.00	40.00
25%	129.50	3.00	4.00	57.50
50%	133.00	5.00	4.00	80.00
75%	137.00	5.50	5.00	97.50
max	142.00	9.00	5.00	150.00

Строка	Что значит
count	сколько не пустых значений
mean	среднее
std	стандартное отклонение
min	самое маленькое
25%	Q1
50%	медиана (Q2)
75%	Q3
max	самое большое

describe() на картинке



mean $\approx$ **50%** (133.3 и 133) $\Rightarrow$
распределение почти симметричное.

Между **25%** и **75%** — половина ребят
(IQR = 7.5 см).

Как «читать» `describe()`: 5 вопросов детектива

- 1 **count** у всех столбцов одинаковый? Если меньше строк — есть **пропуски**.
- 2 **mean** далеко от **50%**? Значит, есть **хвост или выбросы**.
Телефон: 83 против 80 — небольшой хвост вправо.
- 3 **min** и **max** разумные? Рост -5 или 500 см — **ошибка в данных**.
- 4 **std** большой по сравнению с **mean**? Данные **сильно разбросаны**.
Телефон: std 34 при среднем 83 — ребята очень разные.
- 5 **max** намного дальше от 75%, чем 75% от 50%? **Кандидат в выбросы**.

describe() — полезные настройки

```
# для текстовых столбцов  
df.describe(include="object")
```

	имя	фрукт
count	15	15
unique	15	4
top	Аня	яблоко
freq	1	6

unique — сколько разных, **top** — мода,
freq — сколько раз встретилась мода.

```
# свои процентиля  
df["рост"].describe(  
    percentiles=[0.1, 0.9])
```

```
# всё сразу  
df.describe(include="all")
```

```
# удобно читать: повернуть  
df.describe().T
```

.T меняет строки и столбцы местами —
удобно, когда столбцов много.

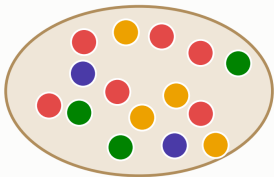
7

`value_counts()` и `unique()`

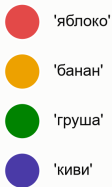


unique() — «какие бывают?»

15 ответов в мешке



.unique() — по одному
каждого вида (4)



```
df["фрукт"].unique()
```

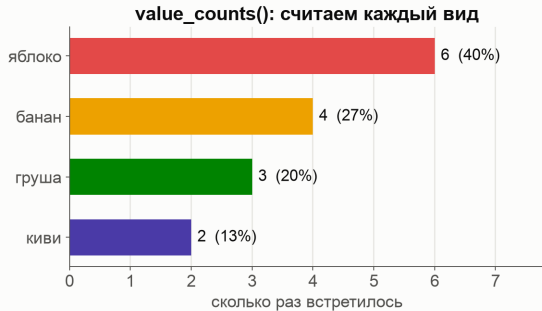
```
['яблоко', 'банан', 'груша', 'киви']
```

```
df["фрукт"].nunique()
```

```
4
```

unique() — список разных значений
(в порядке появления), nunique() —
их **количество**.

value_counts() — «сколько каждого?»



```
df["фрукт"].value_counts()
```

яблоко	6
банан	4
груша	3
киви	2

```
df["фрукт"].value_counts(normalize=True)
```

яблоко	0.40
банан	0.27
груша	0.20
киви	0.13

value_counts() — полезные приёмы

```
# первая строка = мода
df["фрукт"].value_counts().index[0]

# сначала редкие
df["фрукт"].value_counts(ascending=True)

# считать и пропуски (NaN)
df["фрукт"].value_counts(dropna=False)

# числа -> группы (корзины)
df["рост"].value_counts(bins=3)

# сразу график
df["фрукт"].value_counts().plot.bar()
```

Когда что использовать

unique()	какие бывают?
nunique()	сколько видов?
value_counts()	сколько каждого?
mode()	какой чаще всех?

Проверка данных

value_counts() быстро находит опечатки: «Яблоко», «яблоко » и «яблоко» — три **разных** значения!

8

Итоги и викторина



Вопрос	Мера	pandas
Какие мы «обычно»?	среднее, медиана, мода	<code>mean()</code> , <code>median()</code> , <code>mode()</code>
Насколько мы разные?	размах, дисперсия, std	<code>max()-min()</code> , <code>var()</code> , <code>std()</code>
Где граница 25% / 75%?	квартили, процентиля	<code>quantile()</code>
Куда тянется хвост?	асимметрия, эксцесс	<code>skew()</code> , <code>kurt()</code>
Меняются ли вместе?	корреляция, ковариация	<code>corr()</code> , <code>cov()</code>
Всё сразу?	сводка	<code>describe()</code>
Какие бывают и сколько?	частоты	<code>unique()</code> , <code>value_counts()</code>

Викторина!

- 1 Оценки: 2, 5, 5, 5, 3. Найди среднее, медиану и моду.
- 2 В класс пришёл ученик ростом 190 см. Что изменится сильнее — среднее или медиана?
- 3 У двух команд $\text{std} = 1$ и $\text{std} = 6$. Какая команда играет стабильнее?
- 4 Твой рост на 75-м процентиле. Сколько ребят из 100 ниже тебя?
- 5 $r = -0.9$ между «часами игр» и «часами сна». Что это значит?
- 6 Чем отличаются `unique()` и `value_counts()`?

- 1 Среднее $= 20/5 = 4$; по порядку 2, 3, **5**, 5, 5 $\Rightarrow$ медиана 5; мода 5.
- 2 **Среднее** — оно чувствительно к выбросам, медиана сдвинется чуть-чуть.
- 3 Команда с **std = 1**: её результаты почти всегда около среднего.
- 4 **75** ребят ниже тебя (и 25 — выше).
- 5 Сильная обратная связь: кто больше играет, тот **меньше спит**. Но это ещё не доказывает причину!
- 6 `unique()` — **какие** значения есть; `value_counts()` — **сколько раз** каждое встречается.

Домашнее задание

Собери данные своей семьи или друзей (рост, любимый цвет, часы сна) и запусти `describe()` и `value_counts()`!