Конспект по теме «Описательная статистика»

Непрерывные и дискретные переменные

Категориальная (качественная) переменная принимает значение из ограниченного набора.

Количественная (численная) переменная принимает числовое значение в диапазоне. Количественные переменные подразделяются на:

- **Непрерывные**, которые могут принимать любое численное значение.
- **Дискретные**, которые могут принимать строго определённые значения.

Гистограмма частот для непрерывной переменной

Известная нам гистограмма хорошо подходит для работы с **дискретными** переменными. Для отображения частот непрерывных переменных нужно придумать что-то другое.

Одним из подходов к визуализации значений непрерывных переменных является разделение множества значений на интервалы и подсчёт количества значений, попадающих в каждый интервал.

В Pandas при построении гистограммы можно задавать не только количество интервалов - корзин, но и явно указывать их границы:

```
data.hist(bins=[value1, value2, value3, value4, ..., valueN])
```

Однако, этот подход не может дать полное представление о значениях переменной, так как полученная гистограмма сильно зависит от того, как мы разбили множество значений на интервалы

Гистограммы плотностей

Для того, чтобы решить недостаток разбиения на интервалы, применяется метод, отображающий частоту не высотой столбца в

гистограмме, а его площадью. Площадь столбца находят, как площадь прямоугольника: длину интервала умножают на высоту столбца. Найденная площадь — частота непрерывной переменной, а высота столбца — **плотность частоты**. Гистограмма, использующая в качества переменной - столбца плотность частоты, называется **плотностная гистограмма.**

Для того, чтобы оценить, сколько значений попало в любой интервал, не обязательно выбранный для построения, берут два значения и ищут площадь плотностной гистограммы между ними. Полученное число и будет оценкой количества значений, попавших в интервал.

Плотность частоты для непрерывных переменных можно задавать не только прямоугольниками, как на гистограммах, но и кривыми функциями. Работает тот же принцип: площадь между двумя значениями пропорциональна частоте попадания значений в интервал между ними.

Метрики локации данных

Такие характерные значения выборки, как медиана и среднее значение, также называют **метрики локации данных:** по медиане и среднему можно судить, *где* примерно расположен набор данных на числовой оси.

Для расчёта среднего значения берут *все значения датасета* — это наиболее полное использование информации при поиске метрики локации. Среднее значение называют **алгебраическая метрика локации.**

Медиана и квартили просто делят набор данных на части. Медиана — **структурная метрика локации.**

Кто разбросал данные?

Для представления о данных, недостаточно знать метрики локации, нужно ещё понимать, как данные разбросаны вокруг них. У структурной метрики локации есть структурные метрики разброса - квартили.

Для подсчёта разброса значений вокруг алгебраической метрики может применяться такой метод: вычисление среднего расстояние между средним значением и всеми остальными значениями переменной.

Дисперсия

Ранее предложенный метод подсчёта разброса значений вокруг алгебраической метрики, имеет право на существование, но он не всегда даёт полное представление о разбросе.

Улучшенная метрика разброса — не просто среднее расстояние между значениями датасета и средним, а средний квадрат этого расстояния. Эта величина называется **дисперсия** , её находят по формуле:

$$\sigma^2 = \frac{\sum (\mu - x_i)^2}{n}$$

где греческая буква мю μ обозначает среднее арифметическое значение совокупности данных:

$$\mu = \frac{\sum (x_i)}{n}$$

Библиотека **Numpy** в Python содержит большую библиотеку высокоуровневых математических функций. Импортируют её так:

```
import numpy as np
```

Дисперсию рассчитывают методом **var()**

```
import numpy as np
variance = np.var(x)
```

Стандартное отклонение

У дисперсии есть один небольшой недостаток: единица её измерения — это квадрат исходной величины. Чтобы вернуться к исходной единице измерения, из дисперсии извлекают квадратный корень. Получившаяся величина называется **стандартным отклонением:**

$$\sigma = \sqrt{\frac{\sum (\mu - x_i)^2}{n}}$$

Стандартное отклонение находят методом **std()** из библиотеки *Numpy*:

```
import numpy as np
standard_deviation = np.std(x)
```

Если дисперсия известна заранее, можно применить метод **sqrt()** из библиотеки *Numpy*. Корень из дисперсии будет равен стандартному отклонению:

```
import numpy as np
variance = 2.9166666666666665
standard_deviation = np.sqrt(variance)
```

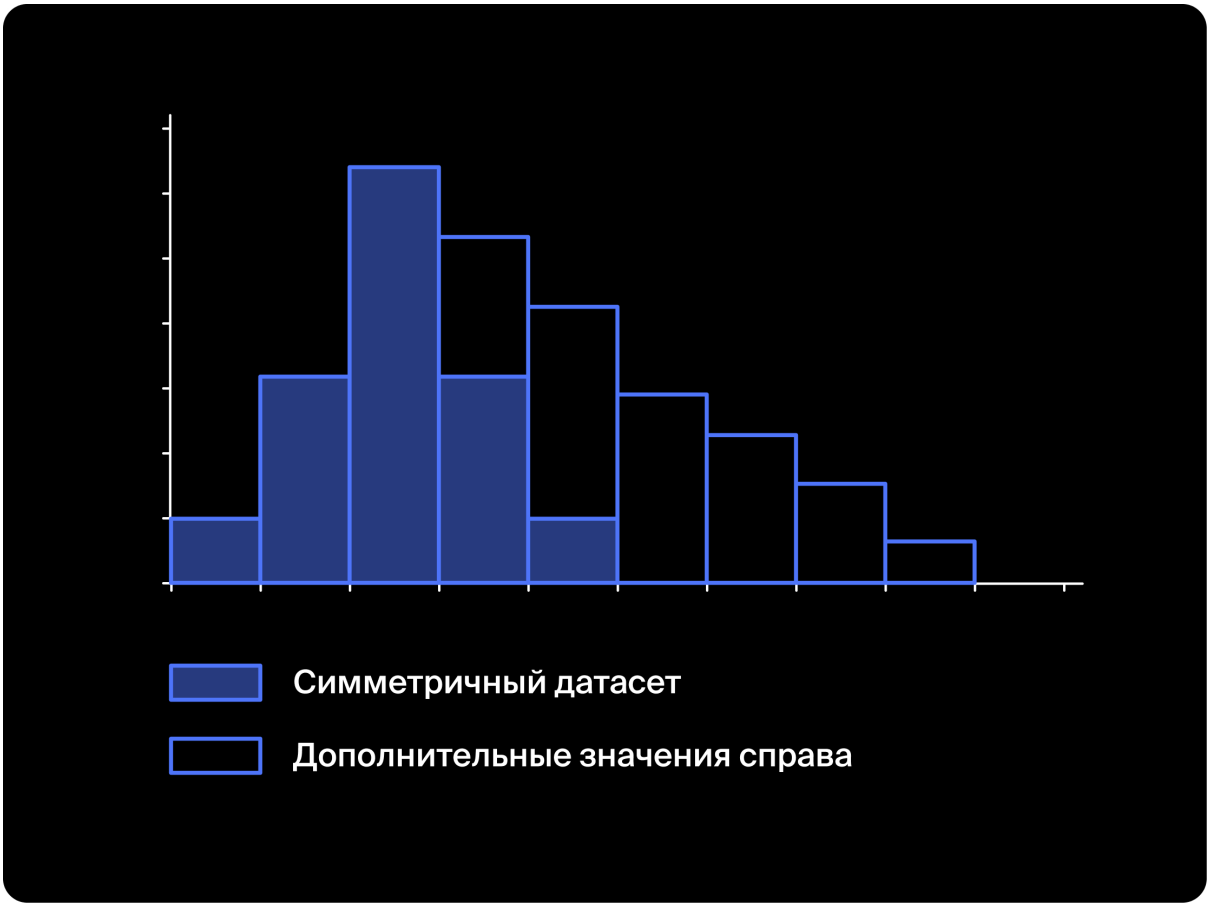
Для большинства распределений верно правило трёх стандартных отклонений, или **правило трёх сигм.** Оно гласит — практически все значения (около 99%) находятся в промежутке:

$$(\mu - 3\sigma, \mu + 3\sigma)$$

Это правило позволяет не только находить интервал, в который наверняка попадут практически все значения интересующей нас переменной, но и искать значения вне этого интервала — часто их называют **выбросами**.

Скошенность наборов данных

Многие данные «из жизни», распределены нормально, или симметрично. Однако датасеты могут быть ассиметричными, то есть, иметь **скошенность** в положительную или отрицательную сторону. Определить скошенность легко по гистограмме. Для этого нужно представить асимметричную гистограмму как симметричную с «дополнительными» значениями.



Такая гистограмма с *дополнительными значениями справа* отображает частоту значений в **скошенном вправо наборе данных.** Его также называют датасетом **с положительной скошенностью**, ведь дополнительные значения находятся со стороны положительного направления оси.

Скошенный влево датасет получится, если добавить к симметричному набору данных значений слева. Если влево идёт отрицательное направление оси, такой набор данных назовут датасетом **с отрицательной скошенностью**.

Скошенность данных также хорошо иллюстрирует диаграмма размаха.

Чтобы понять, в какую сторону скошен датасет, необязательно строить графики. Достаточно взглянуть на метрики локации: медиану и среднее. Помня о том, что медиана в отличие от среднего устойчива к выбросам, легко сделать вывод, что для скошенных вправо данных медиана будет меньше среднего, а для скошенных влево — больше.