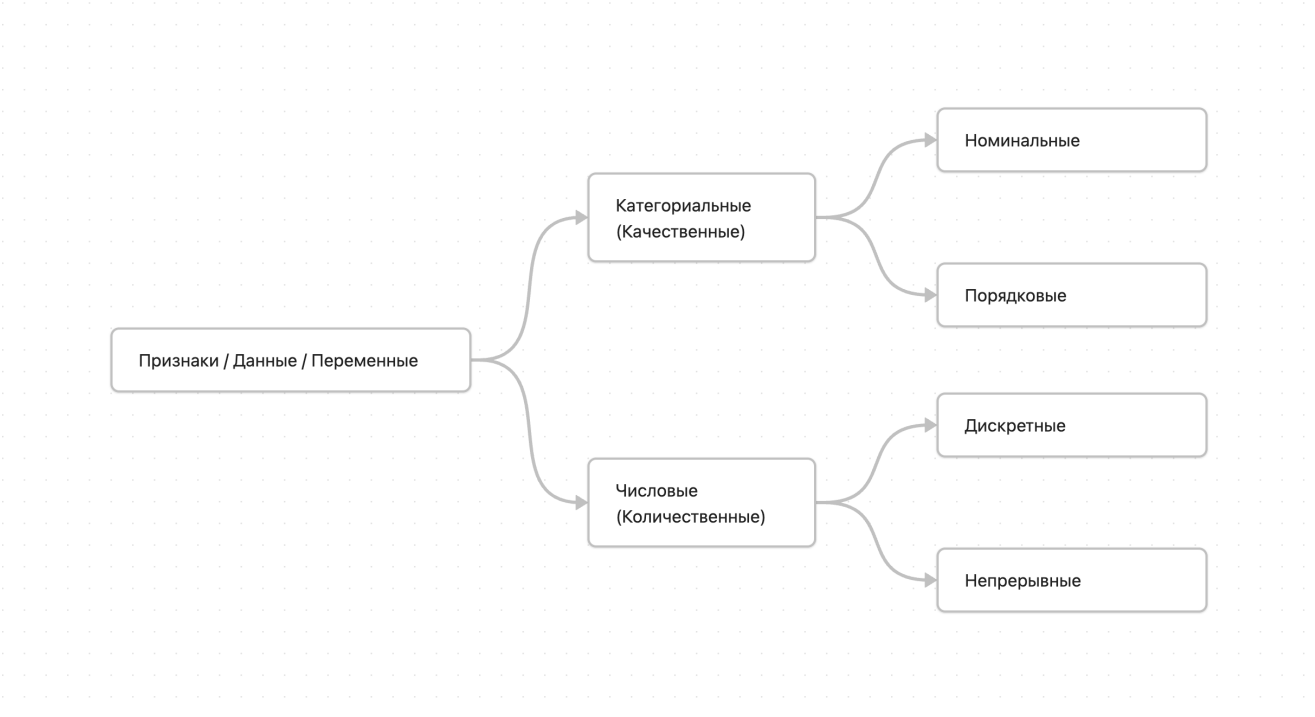
Типы признаков / данных

В мире анализа данных и статистики, разнообразие типов признаков и данных играет важную роль в понимании и интерпретации информации. Признаки представляют собой характеристики или переменные, которые описывают объекты или события, а данные представляют собой значения или наблюдения этих признаков. От правильного определения типов признаков зависит выбор подходящих статистических методов и моделей для анализа данных.

В данной главе мы рассмотрим различные типы признаков и данных, которые широко используются в анализе данных. В данном случае под данными имеются ввиду переменные и наоборот.



Категориальные / Качественные признаки

Категориальные / Качественные признаки - это признаки каждое значение которых указывает на принадлежность объекта к определенной группе. Цифры в данном случае обозначают свойства объекта, служат маркерами, и не имеют математического смысла.

Категориальные / Качественные признаки, в свою очередь, можно поделить на **номинальные** и **порядковые**.

Номинальные признаки

Номинальные признаки применяются для обозначения категорий или признаков, которые нельзя классифицировать по возрастанию или убыванию, т.е они только содержат информацию о принадлежности объекта к какому-то классу.

Примеры:

- Группы крови (Каждой группе крови можно дать число обозначающее ее.)
- Типы горных пород и т.п. (Каждой горной породе можно дать число обозначающее ее.)

Порядковые / Ранговые признаки

Порядковые / Ранговые признаки отличаются от номинальных тем, что в них появляется отношения порядка. То есть здесь у нас значения не только разделяют объекты на классы, но и определенным образом упорядочивают их.

Примеры:

- Список участников забега (Список отражает кто из участиков быстрее, а кто медленнее, но не показывает на сколько быстрее или медленнее)
- Другие примеры в которых подразумевается сравнивание результатов

Количественные / Числовые признаки

Количественные / Числовые признаки отображают количество чего-то.

Их в свою очередь, можно поделить на **дискретные** и **непрерывные**.

Непрерывные признаки

Непрерывные признаки - это признаки, которые могут принимать любое значение из диапазона возможных значений.

К примеру $1, 2.3, 3.3, 4, 8\frac{2}{3}, \dots$ или $[10, 30]$.

Дискретные признаки

Дискретные признаки - это признаки, которые могут принимать только целые значения из ряда возможных значений.

К примеру $1, 2, 3, 4, \ldots$

Такие переменные не могут быть дробными, то есть 3.5 не дискретная переменная.

Вариационный ряд

> Вариационный ряд представляет собой упорядоченный набор значений признака, а также соответствующие им частоты или количество наблюдений.

Вариационный ряд используется для выполнения следующих задач:

1. Изучение распределения данных: Вариационный ряд позволяет визуально представить распределение данных и понять, как они распределены по значению. Это особенно полезно при анализе больших наборов данных, где сложно сразу оценить распределение.
2. Вычисление статистических мер: Вариационный ряд позволяет легко вычислить различные статистические меры, такие как среднее значение, медиана, квартили, минимальное и максимальное значения. Эти меры помогают описать и понять свойства данных.
3. Оценка экстремальных значений: Вариационный ряд позволяет идентифицировать экстремальные значения данных, такие как выбросы или значения, находящиеся на краях распределения. Это помогает выявить необычные или потенциально ошибочные наблюдения.
4. Построение гистограммы и полигона частот: На основе вариационного ряда можно построить гистограмму или полигон частот, которые визуально отображают распределение данных и помогают исследовать их частоту или относительную частоту.

Вариационный ряд состоит из следующих элементов:

- **Варианты**
- **Абсолютные частоты** или **Относительные частоты**

Например, предположим, что у нас есть набор данных о возрасте людей, состоящий из следующих значений:

Возраст	**32**	**28**	**25**	**40**	**32**	**30**	**25**	**28**

Ряд отсортированный в порядке возрастания будет выглядеть следующим образом

```
data_sort = data.sort_values()
print(data_sort.tolist())
```

```
[25, 25, 25, 28, 28, 30, 32, 32, 40]
```

Варианты

> В контексте статистики, термин **"варианты"** относится к уникальным значениям признака, распределенным в порядке возрастания

Варианты значений для данных о возрасте людей будут выглядеть следующим образом:

```
unique = sorted(data.unique())
print(unique)
```

```
[25, 28, 30, 32, 40]
```

Метод `.unique()` используется для получения уникальных значений из массива данных. Результатом этого метода будет новый массив, содержащий только уникальные элементы из исходного массива `data`.

Функция `sorted()` применяется к полученному массиву уникальных значений для их сортировки в порядке возрастания.

Вариационный ряд с абсолютными частотам

> **Вариационный ряд с абсолютными частотами** представляет собой упорядоченный список значений признака вместе с соответствующими **абсолютными частотами**, которые показывают, сколько раз каждое значение встречается в наборе данных.

Для создания вариационного ряда с абсолютными частотами необходимо пройти по всем значениям признака и посчитать, сколько раз каждое значение появляется в данных. Затем значения признака упорядочиваются в порядке возрастания или убывания, а ряду присваиваются соответствующие абсолютные частоты.

```
freq = pd.Series(data_sort).value_counts().sort_index()
freq = pd.DataFrame(freq, index=unique, columns=["Абсолютные частоты"]).T
```

```
freq
```

	25	28	30	32	40
Абсолютные частоты	3	2	1	2	1

Вариационный ряд с относительными частотам

> **Вариационный ряд с относительными частотами** представляет собой вариационный ряд, где вместо абсолютных частот используются относительные частоты. **Относительная частота** указывает на долю каждого значения признака относительно общего количества наблюдений.

Для создания вариационного ряда с относительными частотами необходимо разделить абсолютные частоты на общее количество наблюдений. Это позволяет нам получить представление о том, как часто каждое значение признака встречается относительно всего набора данных.

```
rel_freq = pd.Series(data_sort).value_counts().sort_index() /
len(data_sort)
rel_freq = pd.DataFrame(rel_freq, index=unique, columns=["Относительные
частоты"]).T
rel_freq
```

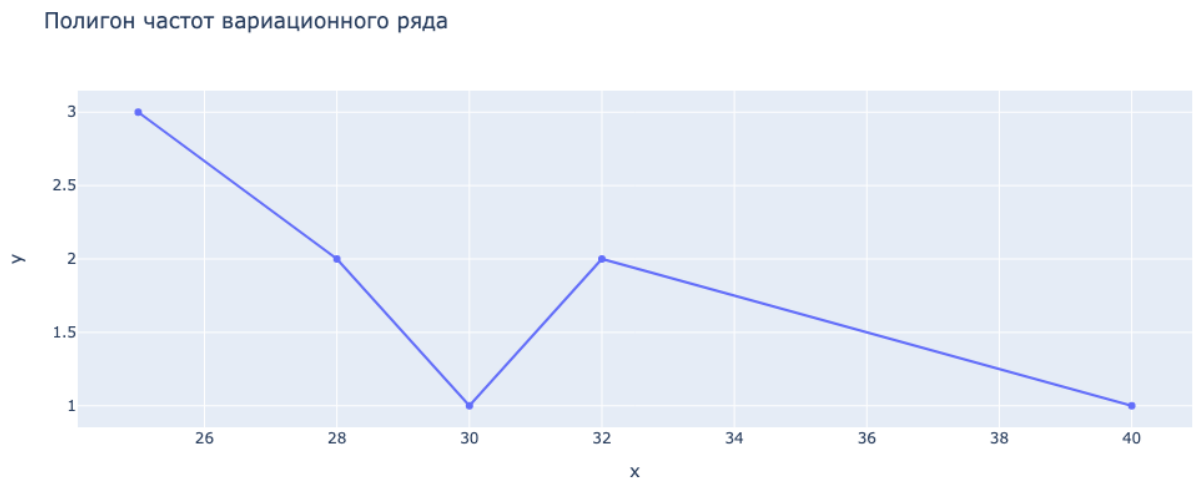
	25	28	30	32	40
Относительные частоты	0.333	0.222	0.111	0.222	0.111

Полигон абсолютных и относительных частот вариационного ряда

> Полигон абсолютных и относительных частот вариационного ряда является графическим представлением распределения значений признака и их соответствующих частот. Он помогает визуализировать данные и понять их распределение на графике.

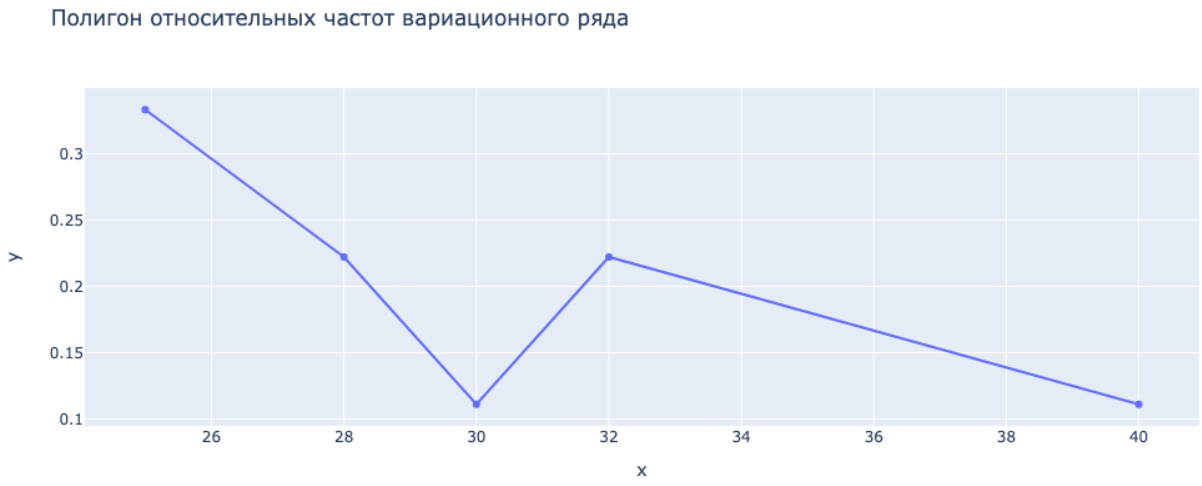
Для полигона **абсолютных частот**, на оси абсцисс откладываются значения признака, а на оси ординат - абсолютные частоты, которые показывают, сколько раз каждое значение встречается в наборе данных. Затем точки с заданными координатами соединяются линиями, образуя полигон, который отображает вариации абсолютных частот признака.

```
fig = px.line(x=unique, y=freq.iloc[0], markers=True, title='Полигон
частот вариационного ряда')
fig.show()
```



Для полигона **относительных частот**, на оси абсцисс также откладываются значения признака, а на оси ординат - относительные частоты, которые показывают, какую долю составляет каждое значение от общего числа наблюдений. Относительные частоты обычно выражаются в виде десятичных долей или процентов. Затем точки с заданными координатами соединяются линиями, образуя полигон относительных частот.

```
fig = px.line(x=unique, y=rel_freq.iloc[0], markers=True, title='Полигон
относительных частот вариационного ряда')
fig.show()
```



Построение полигона абсолютных и относительных частот позволяет визуально анализировать распределение значений признака и выявлять особенности, такие как

пики, моды, симметрию или асимметрию в данных. Это помогает лучше понять характеристики набора данных и делать выводы о его распределении.

Интервальный вариационный ряд

> Интервальный вариационный ряд является особым видом вариационного ряда. Вместо того, чтобы перечислять каждое отдельное значение признака, интервальный вариационный ряд группирует значения в заданные интервалы.

Интервальный вариационный ряд состоит из интервалов значений признака и их соответствующих частот (абсолютных или относительных). Каждый интервал представляет диапазон значений, в котором находятся наблюдаемые значения признака. Частота интервала указывает, сколько значений попадает в этот интервал.

Количество интервалов

> Количество интервалов в интервальном вариационном ряду определяет, сколько равных по ширине интервалов используется для группировки данных. Это важный параметр, который влияет на способ представления и анализа данных.

Когда имеется большое количество уникальных значений или широкий диапазон данных, интервальный вариационный ряд помогает упростить их представление. Вместо перечисления каждого значения в отдельности, данные группируются в интервалы, которые представляют диапазоны значений.

Выбор оптимального количества интервалов зависит от различных факторов, включая размер выборки, размах данных, природа переменной и цель анализа. Часто используются статистические методы, такие как формула Стерджесса для определения рекомендуемого количества интервалов.

$$k = \lceil \log_2 n \rceil + 1 \text{ или } k = \lceil 3.222 \lg n \rceil + 1 \qquad (\text{формула Стерджесса})$$

где

- n - количество элементов вариационного ряда,

```
k = int(np.ceil(np.log2(len(data_sort)) + 1))
print(f'Количество интервалов: {k}')
```

Шаг интервала

> Шаг интервала в интервальном вариационном ряду представляет собой разницу между верхними или нижними границами соседних интервалов. Он определяет

> ширину каждого интервала и используется для группировки данных в интервальный ряд.

$$h = \frac{max(x) - min(x)}{k}$$

где

- $max(x)$ - максимальное значение вариационного ряда,
- $min(x)$ - минимальное значение вариационного ряда,
- k - количество интервалов.

```
h = (data_sort.max() - data_sort.min()) / k
print(f'Шаг интервала: {h}')
```

Интервальный вариационный ряд

Реализация интервального вариационного ряда на питоне выглядит следующим образом:

```
intervals = pd.Series([data_sort.min() + i * h for i in range(k+1)])
print(intervals.tolist())
```

- `data_sort` : Это переменная, которая содержит отсортированный список данных.
- `k` : Это переменная, которая представляет количество интервалов, которые требуется создать в интервальном вариационном ряду.
- `h` : Это переменная, которая представляет шаг интервала.

Распределение абсолютных частот интервального вариационного ряда

```
freq_intervals =
pd.Series(data_sort).value_counts(bins=8).sort_index().values
freq_intervals = pd.DataFrame(freq_intervals, index=intervals_mean,
columns=["Частота"]).T
freq_intervals
```

	25.40-26.80	**26.80-28.20**	**28.20-29.60**	**29.60-31.00**	**31.00-32.40**	**32.40-33.80**	**33.80-35.20**	**35.20-36.60**
Абсолютная частота	6	16	20	26	19	8	3	2

В данном коде используются следующие функции и методы из библиотеки Pandas:

1. `pd.Series()`: Это функция, которая создает объект Series в Pandas. В данном коде, она используется для создания объекта Series `data_sort`, который содержит отсортированные значения данных.
2. `value_counts()`: Это метод, применяемый к объекту Series. Он подсчитывает количество уникальных значений в объекте Series и возвращает результат в виде нового объекта Series, где индексы - это уникальные значения, а значения - их абсолютные частоты. В данном коде, `value_counts()` используется для подсчета абсолютных частот значений в объекте Series `data_sort`.
3. `bins`: Параметр метода `value_counts()`, который указывает количество интервалов, на которые нужно разделить значения при подсчете частот. В данном коде, значение `bins=8` указывает, что данные должны быть разделены на 8 интервалов.
4. `sort_index()`: Метод, применяемый к объекту Series, который сортирует значения по индексу. В данном коде, он используется для сортировки объекта Series `value_counts()` по возрастанию индексов, чтобы значения частот соответствовали правильным интервалам.

Распределение относительных частот интервального вариационного ряда

Для получения распределения относительных частот интервального вариационного ряда, мы можем использовать следующий код, основанный на предыдущем коде:

```
rel_intervals_freq =
pd.Series(data_sort).value_counts(bins=8).sort_index().values /
len(data_sort)
rel_intervals_freq = pd.DataFrame(rel_intervals_freq,
index=intervals_mean, columns=["Относительная частота"]).T
rel_intervals_freq
```

	25.40-26.80	**26.80-28.20**	**28.20-29.60**	**29.60-31.00**	**31.00-32.40**	**32.40-33.80**	**33.80-35.20**	**35.20-36.60**
Относительная частота	0.06	0.16	0.2	0.26	0.19	0.08	0.03	0.02

В процессе получения распределения относительных частот интервального вариационного ряда используется аналогичный подход, но с небольшим отличием. Вместо простого подсчета абсолютных частот значений в объекте Series, мы выполняем дополнительное деление каждой абсолютной частоты на общее количество элементов, чтобы получить относительные частоты.

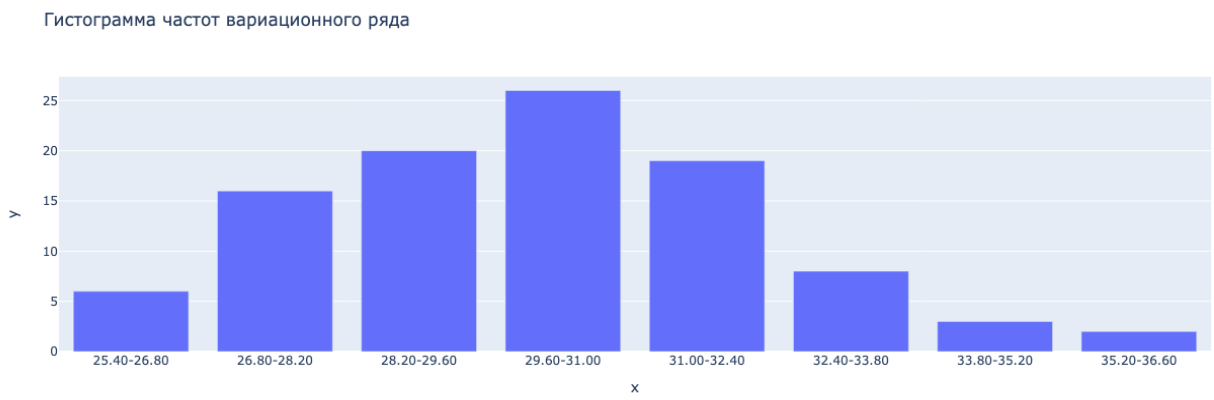
```
rel_intervals_freq =
pd.Series(data_sort).value_counts(bins=8).sort_index().values /
```

```
len(data_sort)
```

Полигон абсолютных и относительных частот интервального вариационного ряда

```
fig = px.bar(x=intervals_mean, y=freq_intervals.iloc[0],
title='Гистограмма частот вариационного ряда')

fig.show()
```




```
fig = px.bar(x=intervals_mean, y=rel_intervals_freq.values.tolist()[0],
title='Гистограмма относительных частот вариационного ряда')

fig.show()
```

Эмпирическая функция распределения (ECDF)

> Эмпирическая функция распределения (empirical cumulative distribution function, ECDF) - это статистическая функция, которая используется для описания и визуализации распределения вероятностей в наборе данных.

> ECDF представляет собой функцию, которая оценивает вероятность того, что случайная переменная принимает значение, меньшее или равное определенной точке. Она строится путем подсчета относительной частоты значений в наборе данных, которые меньше или равны данной точке.

Для построения ECDF сначала сортируются значения в наборе данных по возрастанию. Затем для каждого значения рассчитывается относительная частота, которая представляет собой долю значений, меньших или равных данному значению, от общего числа значений. Наконец, строится график, где по оси X откладываются значения, а по оси Y - соответствующие относительные частоты.

ECDF позволяет наглядно представить распределение данных и увидеть, как часто значения находятся в определенном диапазоне. Она также может использоваться для сравнения распределений разных наборов данных или для сопоставления с теоретическими распределениями.

Эмпирическая функция распределения является полезным инструментом для исследования данных и предоставляет информацию о вероятностях и квантилях распределения.

```
emp_func = rel_intervals_freq.iloc[0].cumsum()
emp_func.name = "F*"
emp_func = pd.DataFrame(emp_func, index=intervals_mean).T
emp_func
```

Метод `.cumsum()` используется для вычисления накопленной суммы значений. Он применяется к объекту Series или DataFrame и возвращает новый объект, содержащий накопленные суммы значений по каждой позиции.

Таким образом, код `emp_func = rel_intervals_freq.iloc[0].cumsum()` вычисляет накопленную сумму относительных частот для первого интервала в DataFrame `rel_intervals_freq` и сохраняет ее в переменную `emp_func`. Это позволяет построить эмпирическую функцию распределения на основе относительных частот.

Среднее (Mean)

> **Среднее арифметическое** (Mean) - Сумма всех элементов исследования $y_1, y_2, \cdots, y_n$ деленная на их количество. Среднее значение набора измерений определяет **только лишь** центр распределения данных. Два набора измерений могут иметь очень разные частотные распределения, но равные средние.

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

где

- n - объем выборки,
- x_i - i-й элемент выборки

Среднее значение по **Генеральной совокупности** обозначается буквой μ. Зачастую μ это неизвестная константа которую мы бы хотели найти.

Символ $\overline{x}$ в английском читается как "x bar" и обозначает **выборочное среднее**.

```
mean = np.mean(data)
print(f'Выборочное среднее: {mean:.2f}')
```

Данный код использует функцию `np.mean()` из библиотеки NumPy для вычисления среднего значения (mean) исходных данных.

- `np.mean()`: Это функция из библиотеки NumPy, которая принимает массив данных в качестве аргумента и вычисляет среднее значение. В данном случае, `data` представляет собой массив данных.

Таким образом, код `mean = np.mean(data)` вычисляет среднее значение исходных данных `data` и сохраняет результат в переменной `mean`.

Дисперсия (Variance)

> **Дисперсией случайной величины** называется математическое ожидание квадрата отклонения этой случайной величины от ее математического ожидания. Дисперсия (Variance) в статистике является мерой разброса или вариации данных относительно их среднего значения. Она показывает, насколько значения признака отклоняются от среднего значения.
>
> **Пояснения:** Зачастую математическое ожидание и среднее по совокупности это одно и то же, поэтому в данном определении используется термин математического ожидания.

Дисперсия по выборочной совокупности обозначается как s^2

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

в то время как дисперсия по генеральной совокупности обозначается как σ^2

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

где

- n - объем выборки,
- y_i - элемент выборки,

- $\overline{y}$ - выборочное среднее,
- $y_i - \overline{y}$ - отклонение элемента выборки от выборочного среднего.

Дисперсия характеризует разброс случайной величины вокруг ее математического ожидания.

Корень из дисперсии называется стандартным отклонением.

```
var = np.var(data, ddof=1)
print(f'Выборочная дисперсия: {var:.2f}')
```

В данном коде используется функция `np.var()` из библиотеки NumPy для вычисления дисперсии (variance) исходных данных.

- `np.var()` : Это функция из библиотеки NumPy, которая принимает массив данных в качестве аргумента и вычисляет дисперсию. В данном случае, `data` представляет собой массив данных.
- `ddof` : Это необязательный параметр функции `np.var()`, который определяет число степеней свободы. Значение `ddof=1` указывает, что используется поправка Бесселя для несмещенной оценки дисперсии, учитывающая размер выборки.

Стандартное / Среднеквадратическое отклонение (Standard deviation)

Стандартное отклонение, также известное как среднеквадратическое отклонение, является мерой разброса или вариации данных относительно их среднего значения. Оно позволяет оценить, насколько значения признака отклоняются от среднего и предоставляет информацию о степени разброса данных. Корень из дисперсии называется средним квадратичным отклонением.

- Для выборочной совокупности

$$s = \sqrt{s^2}$$

- Для генеральной совокупности

$$\sigma = \sqrt{\sigma^2}$$

```
std = np.std(data, ddof=1)
print(f'Среднеквадратическое отклонение: {std:.2f}')
```

- `np.std()` : Это функция из библиотеки NumPy, которая принимает массив данных в качестве аргумента и вычисляет стандартное отклонение. В данном случае, `data` представляет собой массив данных.

- `ddof` : Это необязательный параметр функции `np.std()`, который определяет число степеней свободы. Значение `ddof=1` указывает, что используется поправка Бесселя для несмещенной оценки стандартного отклонения, учитывающая размер выборки.

Таким образом, код `std = np.std(data, ddof=1)` вычисляет стандартное отклонение исходных данных `data` с использованием поправки Бесселя для несмещенной оценки стандартного отклонения, и результат сохраняется в переменной `std`.

Коэффициент вариации (Coefficient of Variation)

> Коэффициент вариации является статистической мерой относительной изменчивости данных, которая позволяет сравнивать степень изменчивости различных наборов данных, учитывая их среднее значение

$$CV = \frac{s}{\overline{x}} \cdot 100$$

```
cv = (std / mean) * 100
print(f'Коэффицент вариации: {cv:.2f}')
```

- `std` : Это значение стандартного отклонения, полученное ранее.
- `mean` : Это значение среднего значения, полученное ранее.

Таким образом, код `cv = (std / mean) * 100` вычисляет коэффициент вариации на основе стандартного отклонения и среднего значения, и результат сохраняется в переменной `cv`. Коэффициент вариации позволяет сравнивать относительную изменчивость между различными наборами данных, независимо от их единиц измерения, и может использоваться для оценки степени риска или неопределенности в данных.