Статистика вывода

Цель занятия

После освоения темы:

- вы сможете отличать и понимать базовые статистические концепции — генеральная совокупность, выборка;
- узнаете свойства нормального распределения;
- узнаете, как использовать библиотеки Python для анализа данных в некоторых задачах машинного обучения;
- сможете вычислить точечные оценки и доверительные интервалы, интерпретировать их;
- сможете проверять статистические гипотезы с использованием тестов на нормальность данных, равенство дисперсий, сравнение средних, взаимосвязь переменных.

План занятия

Используемые термины

Генеральная совокупность (*N*) — это весь набор объектов, о которых мы хотим получить информацию.

Выборка (*n*) — подмножество объектов из генеральной совокупности.

Доверительный уровень — вероятность того, что реальный параметр лежит в границах полученного доверительного интервала: значение параметра ± ошибка выборки (Δ).

Стратификация — выделение подгрупп (страт) на основе важных признаков.

Случайная величина — это такая величина, для которой возможное значение в результате эксперимента зависит от такого большого количества разных факторов, что предсказать ее с ходу невозможно.

Плотность распределения — это взаимоотношение между значением величины и вероятностью того, что величина примет именно это значение (первая производная от ее функции распределения).

Точечная оценка параметра — число, оцениваемое на основе наблюдений, предположительно близкое к оцениваемому параметру.

Надежность — вероятность того, что оценка параметра принадлежит доверительному интервалу.

Конспект занятия

1. Выборка и генеральная совокупность

Соотношение концепции выборки и генеральной совокупности являются ключевыми для многих научных сфер и прикладных аналитических направлений исследовательской деятельности.

Генеральная совокупность (*N*) — это весь набор объектов, о которых мы хотим получить информацию. Набор этих объектов зависит от тех или иных исследовательских задач.

Выборка (*n*) — подмножество объектов из генеральной совокупности. Как правило, это те объекты, которые могут быть непосредственными участниками исследования.

В большинстве случаев выборка и генеральная совокупность не совпадают по размеру.

При решении задач анализа данных мы чаще работаем с выборкой. Например, при исследовании благосостояния россиян мы не можем провести опрос 140+ миллионов человек. В то же время мы не располагаем публичной базой данных по гражданам России с такой информацией. Соответственно, нужно взять выборку и спроецировать те или иные закономерности на всю генеральную совокупность.

Примеры выборки для разных исследований:

- Респонденты маркетингового исследования привлекательности товара *X*.
- Совокупность посещений сайта Y во временные периоды $T_1 - T_n$, собранная для тестирования конверсии нового лендинга.
- База клиентов банка *S*, использованная для решения задачи кредитного скоринга.

Количество респондентов в выборке зависит от целей исследования.

Пример. Представим, что мы пытаемся предсказать целесообразность построения станции метро в микрорайоне и проводим уличный опрос около местного автовокзала.

Вопрос. Важно ли нам количество опрошенных респондентов?

Ответ. Конечно, ведь большинство методов количественного анализа данных просто не будут давать валидные оценки на слишком маленькой выборке.

Как выбрать достаточный размер выборки

Одна из общепринятых формул расчета необходимой выборки (для прикладных опросных исследований):

$$n = \frac{Z^2 pq}{\Delta^2}, \qquad (*)$$

где n — объем выборки;

Z — коэффициент, зависящий от доверительного уровня;

p — доля респондентов с наличием целевого признака;

$q = 1 - p$ — доля респондентов без целевого признака;

Δ — предельная ошибка выборки.

Вернемся к задаче предсказания целесообразности построения станции метро. Представим, что нам нужно определить размер выборки для исследования с доверительным уровнем в 95% и ошибкой не более 4%. При том что мы не знаем, как будет распределяться целевой признак среди респондентов.

Доверительный уровень — это вероятность того, что реальный параметр лежит в границах полученного доверительного интервала: значение параметра ± ошибка выборки (Δ). Доверительный уровень устанавливает сам исследователь в соответствии со своими требованиями к надежности полученных результатов. Чаще всего применяются доверительные уровни равные 0,95 или 0,99.

Перейдем к решению задачи в Python. **Scipy (Scientific Python)** — библиотека Python, специализирующаяся на научных инженерно-математических и статистических исследованиях. Ее функционал хорошо подходит для статистических задач, которые мы применяем в аналитике данных.

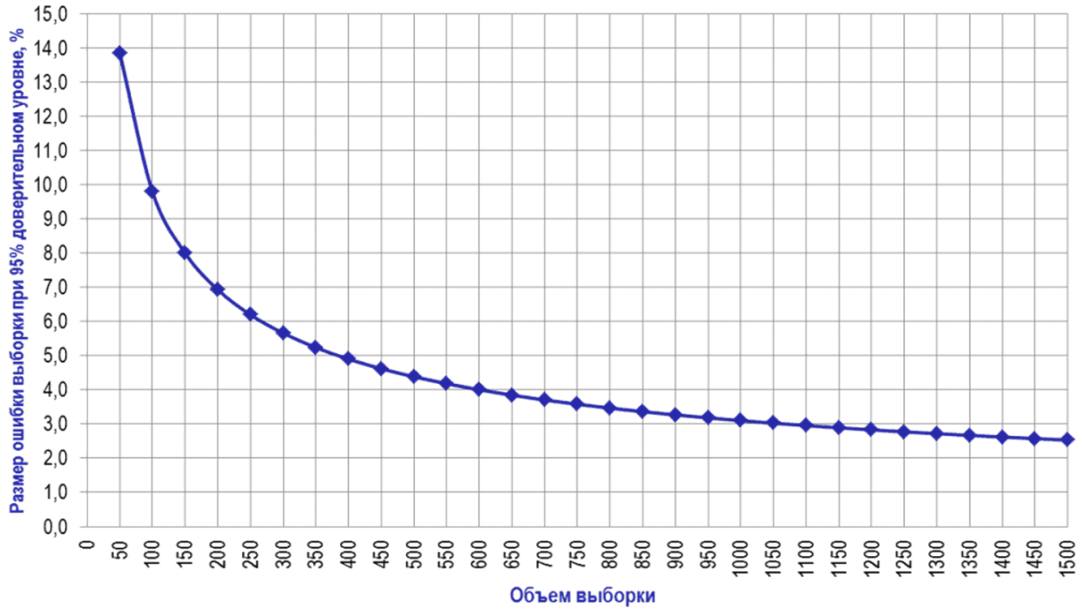
Импортируем модуль `stats` из библиотеки `scipy` и подставим исходные значения из условия в задачи в формулу для расчета размера выборки (*):

```
import scipy.stats as st
conf = 0.95 #доверительный уровень в долях
Z = st.norm.ppf(conf + (1-conf)/2)
p = 0.5
q = 1 - p
delta = 0.04 #ошибка в долях
print(round(((Z**2)*p*q)/delta**2))
```

Результат выполнения: 600 — минимальное количество респондентов в выборке для задачи.

Объем выборки vs Ошибка

Размер выборки — не единственный параметр, по которому мы можем улучшить наше предсказание и сделать его более точным. На графике показано соотношение размера ошибки выборки при 95% доверительном уровне от объема выборки.



С ростом выборки в области минимальных значений падение ошибки существенно. А далее работает **закон убывающей предельной полезности**: с каждым новым респондентом ошибка будет уменьшаться все меньше и меньше.

Как определить достаточность выборки

Часто на практике интересующие характеристики могут быть более сложными (их может быть больше).

Вопрос. Как охарактеризовать генеральную совокупность (ГС) «Все жители России» одним бинарным признаком?

Ответ. Никак. Нам необходима **репрезентативность** нашей выборки — отсутствие значимых различий в характеристиках выборки и ГС. В данном случае одного параметра будет недостаточно.

Вернемся к примеру, где мы пытаемся предсказать целесообразность построения станции метро в микрорайоне и проводим уличный опрос около местного автовокзала.

На первом этапе мы попробовали подсчитать оптимальный размер выборки. Получили, что нам нужно опросить 600 человек.

Однако этого мало для проведения исследования, так как исходный дизайн исследования не предполагает репрезентативности. Это во многом связано с методом сбора данных.

ГС людей, которые по разным причинам находятся у автовокзала, очень далека от ГС жителей микрорайона:

- разная причастность к микрорайону;
- разный образ жизни.

Идеально, если выборка формируется случайным алгоритмом. То есть вероятность попадания каждого члена ГС будет одинакова.

+ Возможно, когда мы имеем в распоряжении данные по всей ГС (например, A/B-тестирование на базе метрик нашего сайта).

- Невозможно, когда нет технической и финансовой возможности достигнуть каждого потенциального члена ГС с равной вероятностью.

В случаях, когда нет возможности сформировать случайную выборку, можно схитрить: обратиться к отдельной науке о том, как формировать выборки, из которых можно делать более-менее репрезентативные результаты.

Стратификация — выделение подгрупп (страт) на основе важных признаков.

Пример. Благосостояние жителей России зависит от множества параметров:

- От пола. Проблема гендерного равенства до сих пор существует.
- От образования. Как правило, более высокий уровень образования скоррелирован с более высоким уровнем благосостояния.
- От возраста.
- От дохода.

В соответствии с этими признаками мы можем разработать определенные страты и попытаемся в нашей выборке реплицировать все эти страты по долям.

Вернемся к задаче предсказания целесообразности построения станции метро. Мы можем выбрать стратификационные критерии:

- Наличие прописки, регистрации, фактического проживания в микрорайоне.
- Потребление транспортных услуг (личный автомобиль, общественный транспорт).
- Социально-демографические критерии (пол, возраст, образование).

Конечного списка здесь предложить нельзя. Нужно опираться на похожие исследования, здравый смысл и метод проб и ошибок.

2. Распределения

Пусть нам удалось сформировать выборку. Для того чтобы количественно анализировать собранные данные, понимать описательные статистики и правильно их интерпретировать, необходимо знать о концепте распределения.

Случайная величина

Случайная величина — это такая величина, для которой возможное значение в результате эксперимента зависит от такого большого количества разных факторов, что предсказать ее с ходу невозможно.

Примеры:

- Значение на верхней плоскости «правильного кубика» в результате его броска (6 значений).
- Рост случайного выбранного студента в аудитории (бесконечное множество всех возможных значений).

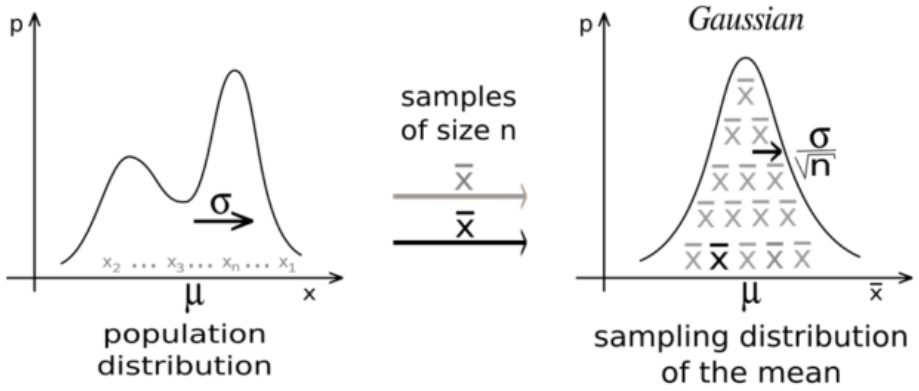
Существует два типа случайных величин (СВ):

- **Дискретная СВ** принимает отдельно взятые, изолированные значения (но их число может быть бесконечным).
- **Непрерывная СВ** может принимать абсолютно все числовые значения (на промежутке).

Центральная предельная теорема и нормальное распределение

На практике мы часто работаем с суммой сразу всего множества случайных величин, поэтому здесь важно поговорить о **центральной предельной теореме**.

Центральная предельная теорема (ЦПТ) гласит, что сумма независимых одинаково распределенных случайных величин имеет распределение, близкое к нормальному. Поэтому и распределение параметров этих величин будет нормальным.



Нормальное распределение описывает плотность распределения многих случайных непрерывных величин.

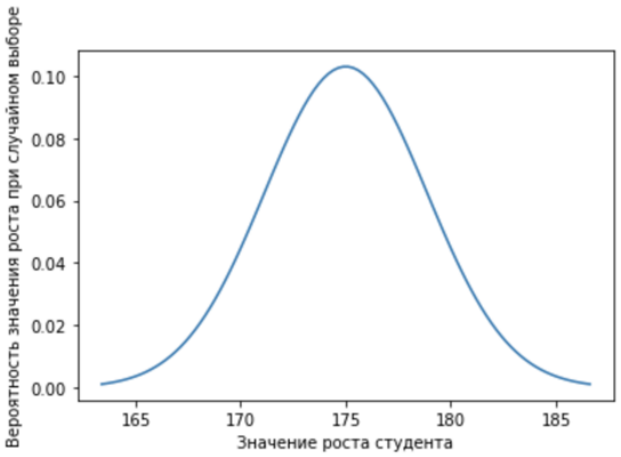
Плотность распределения — это взаимоотношение между значением величины и вероятностью того, что величина примет именно это значение (первая производная от ее функции распределения). То есть плотность распределения показывает частоту встречаемости того или иного значения для нашей случайной величины.

Функция плотности нормального распределения:

$$f(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{\frac{(-x-\mu_x)^2}{2\sigma_x^2}}$$

Пример. Представим, что мы хотим посмотреть на распределение роста студентов на потоке. Итоговое распределение — сумма множества случайных одинаково распределенных случайных величин (рост конкретного человека).

По ЦПТ получим следующую плотность распределения:



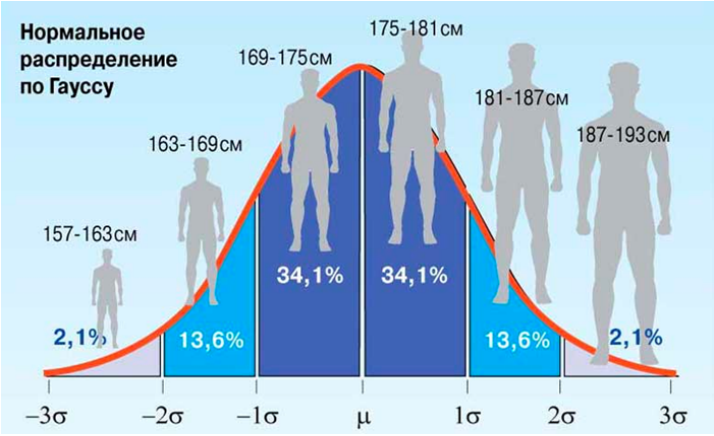
У нормального распределения есть ряд свойств:

1. Кривая нормального распределения выпукла (колоколообразная), с симметрией относительно среднего, с точками перегиба в значениях абсцисс суммы и разности среднего и стандартного отклонения.
2. Нормальное распределение определяется двумя параметрами: значением генерального среднего (μ) и генерального стандартного отклонения (σ).
3. Медиана и мода нормального распределения совпадают и равны μ.
4. Коэффициенты асимметрии и эксцесса нормального распределения равны нулю.

Таким образом, нормальное распределение — идеальная модель, под которую мы подстраиваем некоторые распределения эмпирических данных.

Нормальное распределение подчиняется **правилу «трех сигм»**: если непрерывная СВ распределена нормально, то практически все ее значения лежат в промежутке от разности среднего и трех стандартных отклонений до их суммы.

Пример. Распределение роста человека:



Примерно 68% значений всех наблюдений выборки заключено на промежутке $[-\sigma, \sigma]$.

3. Оценки

Пусть мы собрали некоторую выборку для нашего исследования. Смогли изучить характеристики, построить распределения. Далее нам необходимо вычленить определенные оценки и параметры.

Точечная оценка

Точечная оценка параметра — число, оцениваемое на основе наблюдений, предположительно близкое к оцениваемому параметру.

Цель нахождения точечной оценки — это восстановить параметр генеральной совокупности по выборке.

Генеральный параметр	Формула точечной оценки
Среднее μ	$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n}$
Дисперсия σ^2	$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} (x_i - \hat{\mu})^2}{n-1}$
Доля π	$\hat{\pi} = \frac{\sum_{i=1}^{n} x_i}{n}$

Для полного представления о качестве оценок необходимо знать три свойства, которыми они должны обладать.

1. **Состоятельность.** Оценка стремится к значению генерального параметра.
2. **Несмещенность.** Математическое ожидание оценки равно оцениваемому параметру генеральной совокупности.
3. **Эффективность.** Оценка, которая имеет при заданном объеме выборки *n* наименьшую дисперсию среди всех возможных несмещенных точечных оценок.

Доверительные интервалы

Любая точечная оценка будет иметь погрешность, поскольку в большинстве задач размер выборки и размер генеральной совокупности значимо отличаются.

Поэтому лучше в явном виде смоделировать эту погрешность, чтобы показать, насколько точечная оценка потенциально точно описывает возможный параметр генеральной совокупности.

Пример. Мы провели репрезентативный опрос по поводу строительства новой станции метро в микрорайоне на выборке в 600 человек и выяснили, что 75% жителей поддерживают строительство. Но мы опросили только часть жителей. Даже при идеальном дизайне не может быть 100% уверенности в точности параметра. Поэтому оценим интервал для параметра, чтобы смоделировать погрешность.

Доверительный интервал — это такой интервал, который покрывает неизвестный параметр с заданной надежностью. Чем меньше выборка, тем в большей степени доверительный интервал предпочтительнее точечной оценки.

Общая формула для доверительного интервала:

$CI = (\hat{\theta} - \epsilon, \hat{\theta} + \epsilon)$, где $\hat{\theta}$ — оцениваемый параметр, ϵ — ошибка

Надежность — вероятность того, что оценка параметра принадлежит доверительному интервалу. Часто также используют термин статистическая значимость, которая определяется как (*1 — надежность*).

Пример. Надежность 0.99 соответствует уровню значимости 0.01 (также могут задаваться в процентах).

При нахождении **доверительного интервала для среднего** нужно обратить внимание на ее размер.

Маленькая выборка (не более 30 наблюдений):

$$CI_{mean}^{n\leq 30} = (\hat{\mu} - t\frac{\hat{\sigma}}{\sqrt{n}}; \hat{\mu} + t\frac{\hat{\sigma}}{\sqrt{n}})$$

Большая выборка (более 30 наблюдений):

$$CI_{mean}^{n>30} = (\hat{\mu} - z\frac{\hat{\sigma}}{\sqrt{n}}; \hat{\mu} + z\frac{\hat{\sigma}}{\sqrt{n}})$$

Здесь:

$\hat{\mu}$ — точечная оценка среднего,
$\hat{\sigma}$ — точечная оценка стандартного отклонения,
n — размер выборки,
t — табличное значение, распределение Стьюдента,
z — табличное значение стандартного нормального распределения.

Пример. Дана выборка ростов студентов потока, которую мы поместили в список `data`:

```
data = [187, 185, 165, 145, 152, 168, 172, 179, 180, 195, 168,
168, 170, 172, 160]
```

Узнаем, в каком промежутке будет находиться среднее генеральной совокупности с 90, 95 или 99% надежностью.

```
#90% надежность
print(st.t.interval(alpha=0.90, df=len(data)-1,
loc=np.mean(data), scale=st.sem(data)))
```

Результат выполнения: `(165.11615700777816, 177.01717632555517)`

```
#95% надежность
print(st.t.interval(alpha=0.95, df=len(data)-1, loc=np.mean(data),
scale=st.sem(data)))
```

Результат выполнения: `(163.82059816147947, 178.31273517185386)`

```
#99% надежность
print(st.t.interval(alpha=0.99, df=len(data)-1, loc=np.mean(data),
scale=st.sem(data)))
```

Результат выполнения: `(161.00953301227102, 181.1238003210623)`

Можно заменить, что чем выше надежность, тем шире доверительный интервал. Это вполне логично. Чтобы быть более уверенными в результатах, необходимо сделать большую поправку при оценке интервалов.

Для случая большой выборки будет использоваться аналогичный метод:

```
st.norm.interval(alpha=a, loc=np.mean(data), scale=st.sem(data))
```

Доверительный интервал для доли рассчитывается по следующей форме:

$$CI_{proportion} = (\hat{\pi} - z\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}}; \hat{\hat{\pi}} + z\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}})$$

Пример. Мы провели репрезентативный корпоративный опрос в компании *X*. На выборке в 150 респондентов 93 респондента заявили о недовольстве введением фиксированного времени обеда. В каком промежутке с 95% надежностью будет находиться доля недовольных во всей компании:

```
from statsmodels.stats.proportion import proportion_confint
```

```
proportion_confint(93, 150, alpha=(1 - 0.95))
```

Результат выполнения: `(0.5423234184516537, 0.6976765815483463)`

4. Тестирование гипотез

Для более глубокого изучения данных точечных и интервальных оценок недостаточно. Тестирование гипотез — один из ключевых процессов анализа данных.

Источником гипотез может быть фантазия исследователя, а также устоявшиеся практики в области.

Важно!

- Даже самую креативную и свежую гипотезу нужно уметь переводить на язык статистики.
- Мы должны понимать условия фальсификации и верификации гипотезы.

Тестирование гипотез можно проводить по единому алгоритму.

Рассмотрим шаги приведенного алгоритма более подробно:

1. Необходимо определиться с исследовательским вопросом и сформулировать гипотезы на языке статистики. Всегда должна быть сформулирована **нулевая гипотеза** и **альтернативная**, они должны быть взаимоисключающими. Например, возьмем какой-нибудь параметр μ:
    - H_0: μ=x, H_1: μ≠x.

- H_0: $\mu \leq x$, H_1: $\mu > x$.
- H_0: $\mu \geq x$, H_1: $\mu < x$.

Сумма вероятностей условий происхождения нулевой гипотезы и альтернативной равна 1.

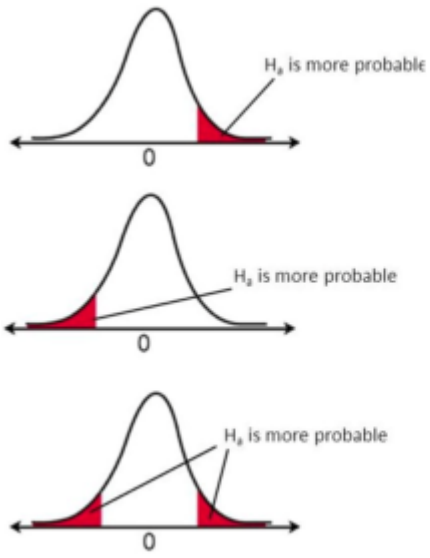
2. Чтобы решить, использовать ли параметрическую или непараметрическую версию теста, мы должны проверить конкретные требования, перечисленные ниже:
    - Наблюдения в каждой выборке независимы и одинаково распределены (IID — independent and identically-distributed).
    - Наблюдения в каждой выборке распределены нормально.
    - Наблюдения в каждой выборке имеют одинаковую дисперсию.
3. Выбор теста.
4. Имплементация теста с выбранными гиперпараметрами (в частности, степень надежности).
5. Интерпретация — статистический вывод.

Статистический вывод делается на основе статистики конкретного теста. Но тестов много, запомнить пограничные значения для каждой спецификации практически невозможно. Поэтому каждый тест предоставляет **p-value (p-значение)** — вероятностную оценку того, что статистика принимает то или иное значение случайно. То есть, чем меньше p-value, тем больше вероятность того, что какая-то закономерность не случайна.

P-value для теста необходимо сравнить с **пороговым значением значимости** (например, 0,05) или статистической надежности (0,95). Превышение порогового значения дает основания отвергнуть нулевую гипотезу в пользу альтернативной.

Для одного и того же уровня значимости может существовать множество алгоритмов проверки гипотез.

Гипотеза может быть правосторонней, левосторонней и двусторонней:



Right-tail test

$H_a: \mu > value$

Left-tail test

$H_a: \mu < value$

Two-tail test

$H_a: \mu \neq value$

Особенностью тестирования гипотез будет заключаться в том, что при двусторонней гипотезе значение статистической значимости будет делиться пополам.

Тестов огромное множество, поскольку есть очень много разных логик постановки гипотез.

Некоторые важные тесты, которые часто встречаются при анализе данных:

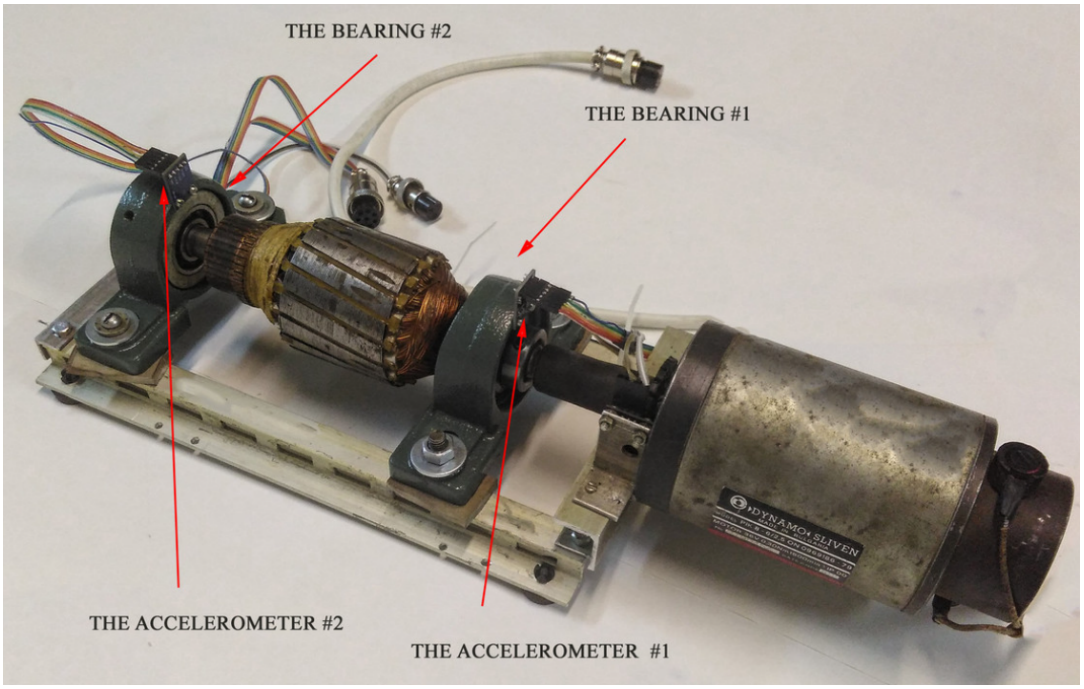
- на нормальность данных;
- на равенство дисперсий;
- на сравнение средних;
- на взаимозависимость переменных.

5. Определение неисправностей в подшипниках через анализ экспериментальных данных

Рассматриваемая задача относится к задачам предиктивного обслуживания технических систем.

Предиктивное обслуживание — это стратегия непрерывного мониторинга состояния оборудования при стандартных условиях эксплуатации и прогнозирования оставшегося срока его службы. Реактивное и превентивное техобслуживание помогает снижать количество сбоев или просто предотвращать их, в то время как предиктивное обслуживание использует модели для предсказания сбоев компонентов конкретной единицы. В предиктивном обслуживании решают задачу обнаружения аномалий для оценки работоспособности деталей узлов оборудования за счет выявления отклонения значений от нормального распределения.

Мы будем решать задачу, связанную с экспериментами на работоспособность подшипников. На рисунке ниже показан двигатель и два подшипника, которые соосно соединены с ним. Также имеются акселерометры, которые измеряют вибрации по трем осям для каждого из подшипников.



Задача заключается в анализе данных экспериментов с целью обнаружения исправных и неисправных подшипников.

Есть данные о работе подшипников (bearings):

- `a1_x` — ускорение для оси X для первого подшипника (м/с^2);
- `a1_y` — ускорение для оси Y для первого подшипника (м/с^2);
- `a1_z` — ускорение для оси Z для первого подшипника (м/с^2);
- `a2_x` — ускорение для оси X для второго подшипника (м/с^2);
- `a2_y` — ускорение для оси Y для второго подшипника (м/с^2);
- `a2_z` — ускорение для оси Z для второго подшипника (м/с^2);
- `hz` — скорость вращения двигателя (Гц).

Данные представлены по трем осям для первого подшипника и для второго. Первый подшипник фиксированный, второй подшипник мы меняем. Таким образом мы определяем, какой подшипник пригоден для эксплуатации.

Заранее задана разметка классов подшипников:

- Status = 1 — означает, что подшипник нормальный;

- Status = 0 — означает, что подшипник с дефектами.

В ходе решения задачи необходимо:

1. Загрузить данные.
2. Сделать выборку.
3. Проанализировать статистику показателей и разметки.
4. Сгенерировать дополнительные показатели (признаки).

Как правило, дополнительные признаки генерируются для ряда показателей и затем используются в машинном обучении для классификации.

Начинаем с загрузки данных:

```
import numpy as np
import pandas as pd

import plotly.express as px # library for plotting
import matplotlib.pyplot as plt

dir = '/content/drive/MyDrive/Advanced ML course/Module
1/Video/input'

import os
for dirname, _, filenames in os.walk(dir):
    for filename in filenames:
        print(os.path.join(dirname, filename))

signals = pd.read_csv(dir+'/bearing_signals.csv',
low_memory=False)
labels = pd.read_csv(dir+'/bearing_classes.csv',sep=";")
```

В `signals` хранятся данные по сигналам из экспериментов по подшипникам, а в `labels` так называемая разметка.

Таблица `signals` содержит временные метки, ускорения и скорость вращения двигателя:

	experiment_id	bearing_1_id	bearing_2_id	timestamp	a1_x	a1_y	a1_z	a2_x	a2_y	a2_z	hz
0	1	0	1	0.000000	0.113269	0.149706	-0.110275	-0.186030	0.194450	0.454299	0.000000
1	1	0	1	0.000333	-0.367713	-0.228832	0.177821	0.285992	0.002226	-0.043930	0.000000
2	1	0	1	0.000667	0.113269	0.149706	-0.398371	-0.091625	0.002226	0.454299	0.000000

Получаем количество уникальных экспериментов по каждому из подшипников:

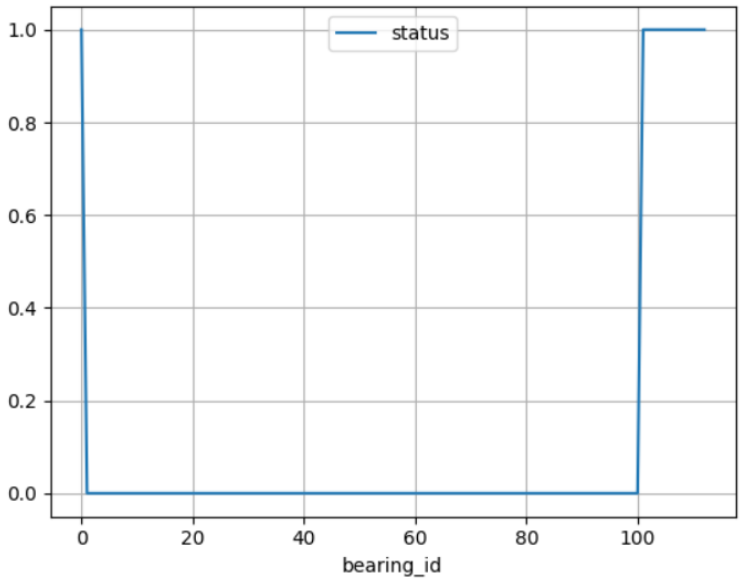
```
signals['experiment_id'].unique()
```

Количество экспериментов — 112.

Далее отобразим статус:

```
labels.plot(x = 'bearing_id', y  = "status", grid = "on", kind =
"line")
```

На полученном графике показано, что «1» соответствуют нормальные подшипники, а «0» — неисправные:



Видно, что работоспособных подшипников меньше. Как правило, в практических задачах часто бывает наоборот. Этот момент важно отметить, так как он далее влияет на баланс классов и используется в машинном обучении.

Удаляем из списка статуса подшипников нулевой, так как он точно нормальный:

```
labels = labels.drop(index= 0)
```

Далее выведем номера нормальных подшипников:

```
labels_status_norm =
labels['bearing_id'][labels['status']==1].values
labels_status_norm
```

Аналогично выводим номера дефектных подшипников:

```
labels_status_defect =
labels['bearing_id'][labels['status']==0].values
labels_status_defect
```

Проведем анализ показателей одного эксперимента:

```
id_experiment = 105
```

```
experiment = signals[signals['experiment_id']==id_experiment]
experiment.info()
```

С помощью функции `info()` получаем информацию о количестве точек эксперимента по времени, ускорению и по вращению двигателя:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 98100 entries, 9501600 to 9599699
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   experiment_id  98100 non-null  int64
 1   bearing_1_id   98100 non-null  int64
 2   bearing_2_id   98100 non-null  int64
 3   timestamp      98100 non-null  float64
 4   a1_x           98100 non-null  float64
 5   a1_y           98100 non-null  float64
 6   a1_z           98100 non-null  float64
 7   a2_x           98100 non-null  float64
 8   a2_y           98100 non-null  float64
 9   a2_z           98100 non-null  float64
 10  hz             98100 non-null  float64
dtypes: float64(8), int64(3)
memory usage: 9.0 MB
```

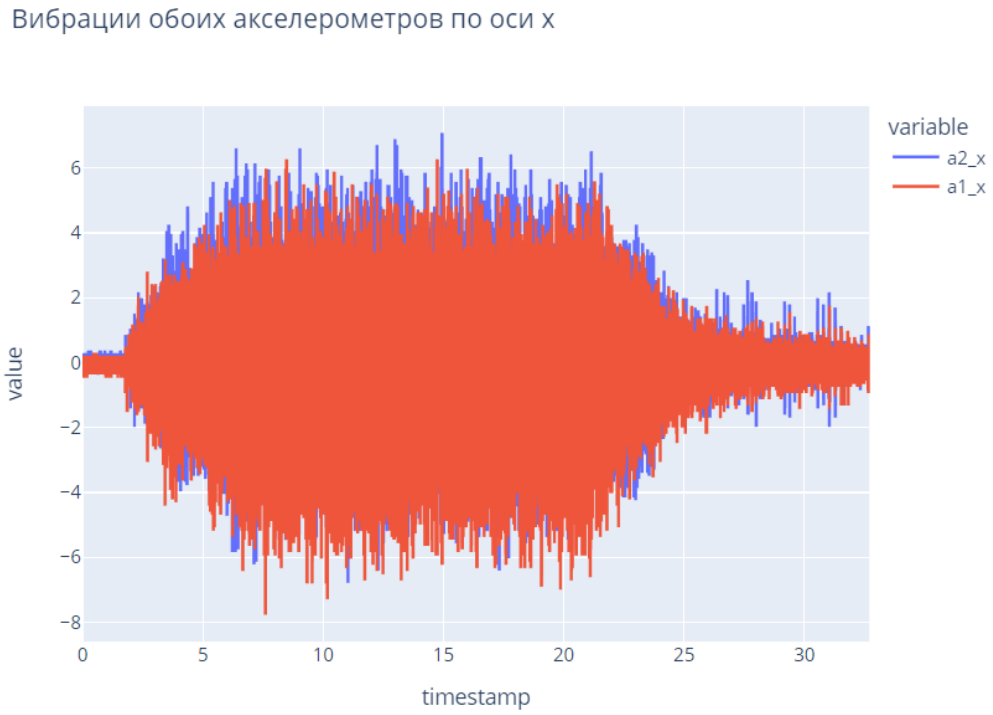
С помощью функции `describe()` получаем усредненные показатели — количество точек, среднее значение, минимум, максимум и другие:

```
experiment.decsribe()
```

Далее перейдем к визуализации эксперимента. Будем использовать специальную библиотеку `plotly.express`, которую ранее импортировали под именем `px`:

```
fig = px.line(experiment, x="timestamp", y=["a2_x","a1_x"],
title='Вибрации обоих акселерометров по оси x')
fig.show()
```

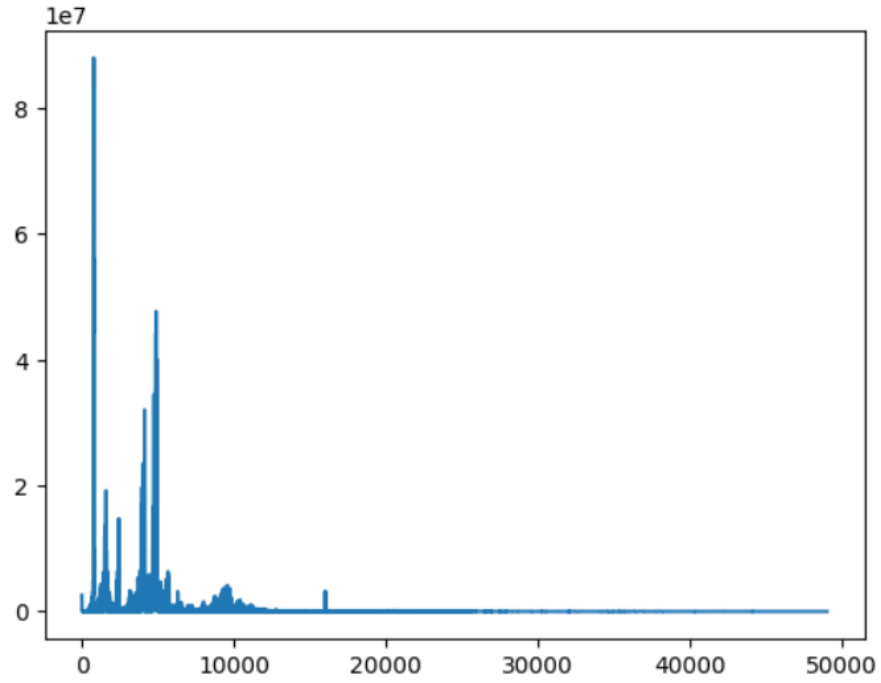
Для первого и второго подшипников мы вывели значения акселерометров, которые соответствуют колебаниям на графике, за период времени чуть больше 30 секунд. Причем видно, что амплитуда сигнала сначала нарастает, затем держится приблизительно одинаковой, а далее падает:



Можно построить спектр мощности сигнала:

```
fftData=abs(np.fft.rfft(experiment['a2_x']))**2
plt.plot(fftData[1:])
```

Результат:

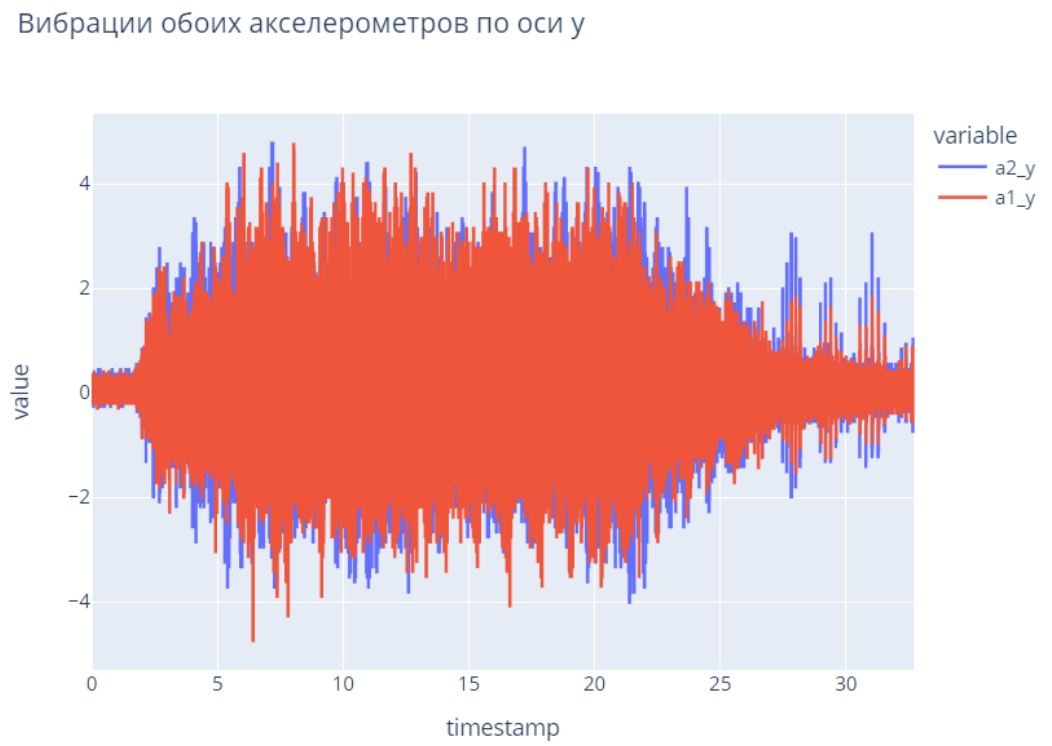


Целесообразно для дальнейшего анализа рассматривать не только пиковое значение в полученном спектре, но и несколько других.

Аналогичным образом был построен график по другой оси:

```
fig = px.line(experiment, x="timestamp", y=["a2_y","a1_y"],
title='Вибрации обоих акселерометров по оси y')
fig.show()
```

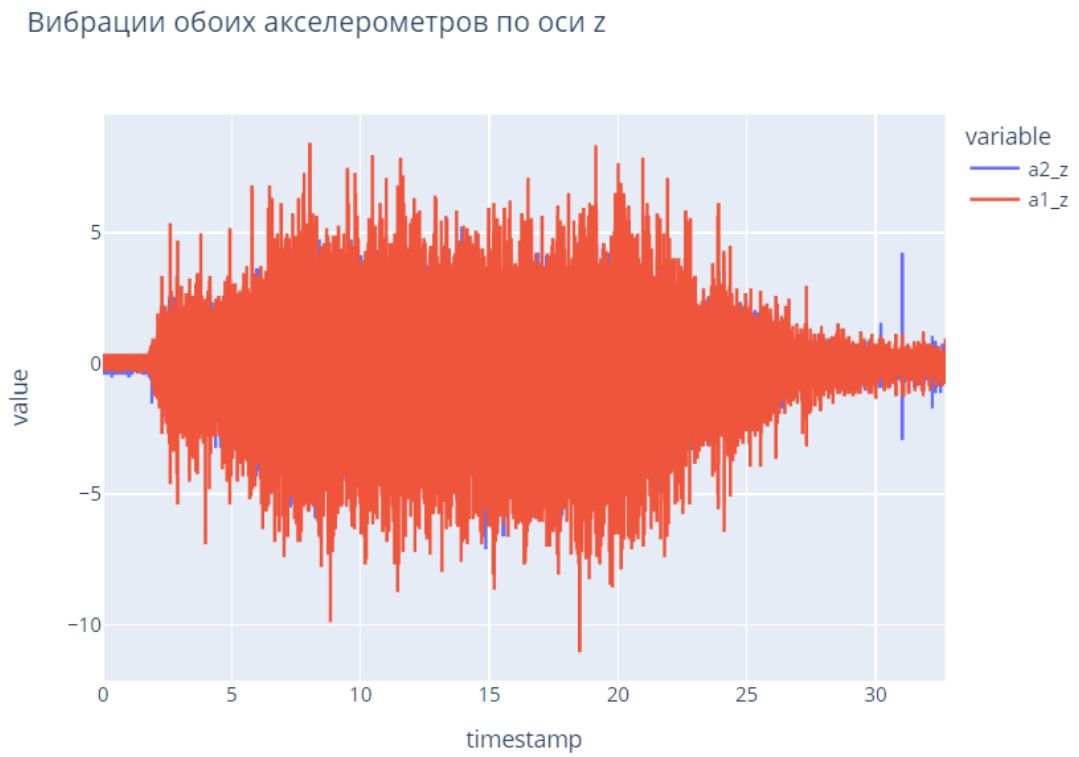
Результат:



Построение графика по оси z:

```
fig = px.line(experiment, x="timestamp", y=["a2_z","a1_z"],
title='Вибрации обоих акселерометров по оси z')
fig.show()
```

Результат:

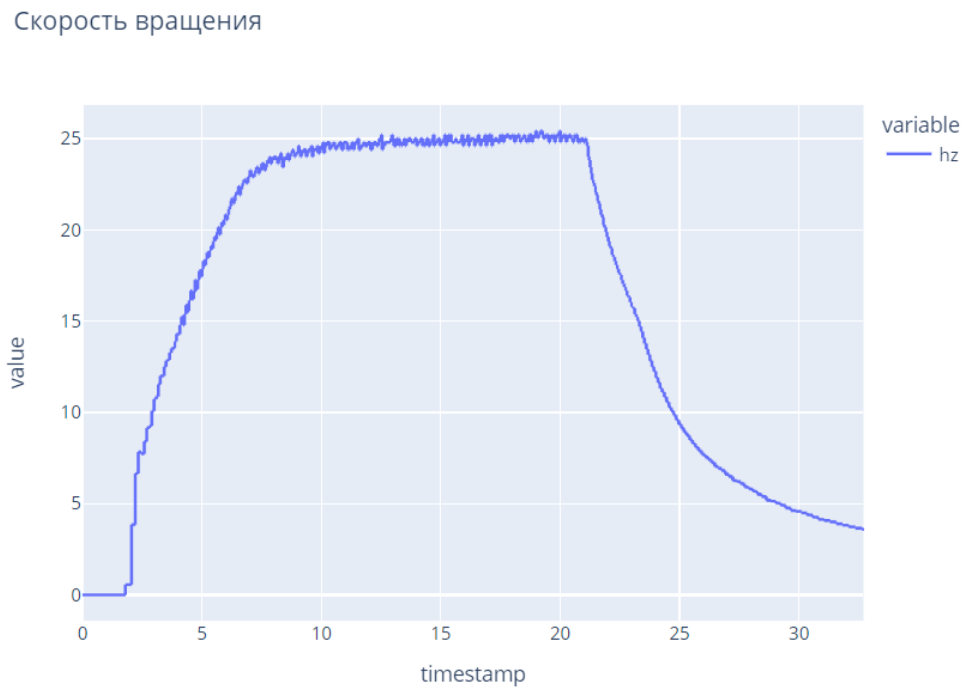


Мы получаем подобные картинки, но вместе с тем можем видеть, что, например, для оси z амплитуда имеет несколько другой характер.

Визуализируем скорость вращения двигателя:

```
fig = px.line(experiment, x="timestamp", y=["hz"], title='Скорость
вращения')
fig.show()
```

Результат:



Из графика видно, что скорость вращения двигателя сначала нарастает, затем идет некоторая планка, далее идет убывание.

Для проведения анализа целесообразно делать выборки. Причем эти выборки могут быть сделаны не статистически, а исходя из физических принципов. Например, можно анализировать значения, приходящиеся на «планку», также представляют интерес значения соответствующие участкам нарастания и убывания.

Важно! Выбор данных должен быть физически обоснован. В первую очередь это касается технических систем.

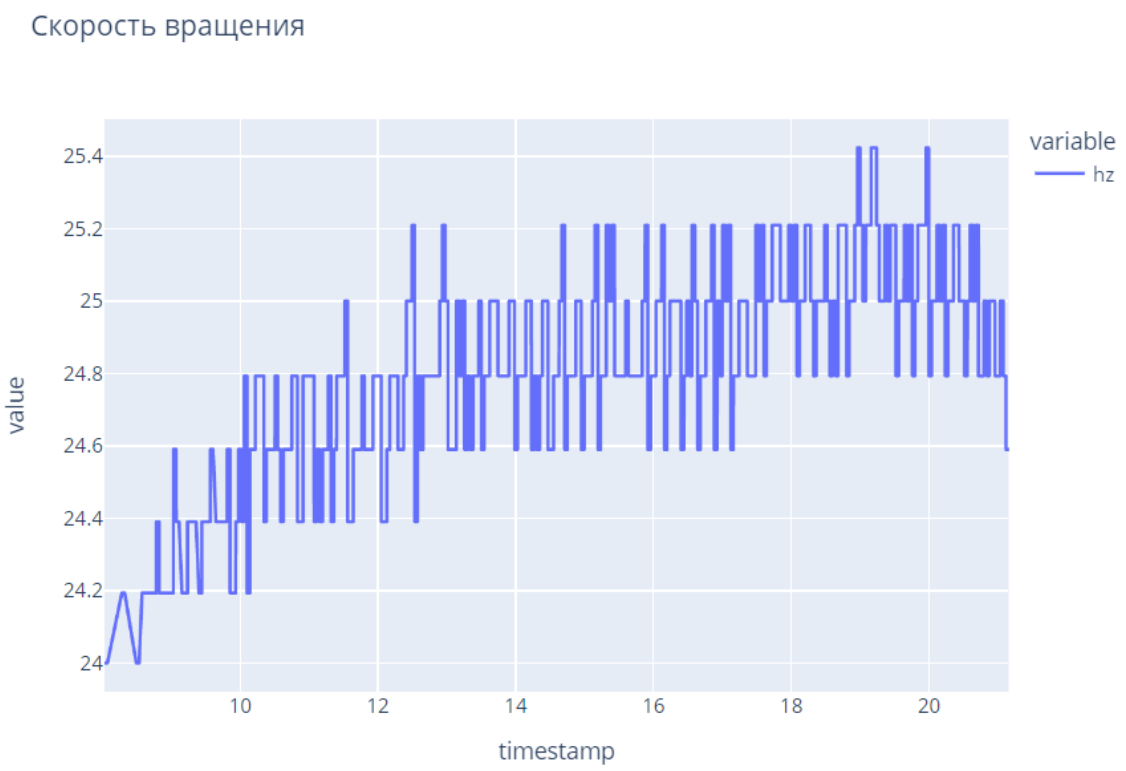
Перейдем к анализу показателей нескольких экспериментов. Берем данные эксперимента, где скорость вращения двигателя более стабильна:

```
steady_filter = (experiment['hz'] > 24) & (experiment['hz'] < 27)
experiment_steady_speed = experiment[steady_filter]
```

Визуализируем скорость вращения:

```
fig = px.line(experiment_steady_speed, x="timestamp", y=["hz"],
title='Скорость вращения ')
fig.show()
```

Результат:



На графике мы видим разброс значений. Скорее всего, он обусловлен ошибками датчика.

С помощью функции `describe()` мы можем посмотреть на краткую статистическую сводку по данным:

```
experiment_steady_speed['a2_y'].describe()
```

Результат:

```
count    37371.000000
mean         0.081133
std          1.222642
min         -3.842249
25%         -0.766669
50%          0.098338
75%          0.963345
max          4.711709
Name: a2_y, dtype: float64
```

Далее создадим функции для генерации признаков. В примере представлены 6 основных функций. На практике таких функций может быть больше. При анализе

признаков выделяют, какие признаки подходят больше, а какие меньше. Таким образом выполняют так называемый анализ значимости. Но сам подход при этом не меняется.

Обычно из данных временных рядов генерируют различные признаки начиная с простых — максимум, минимум, стандартное отклонение и т. д. В примере дополнительно добавлена функция, которая вычисляет частоту (отсчет), соответствующую максимуму амплитуды:

```
#функция определения максимального абсолютного значения
def get_peak_acceleration(signal):
    return pd.DataFrame.max(signal.abs())

#функция определения стандартного отклонения
def get_std(signal):
    return signal.std()

#функция определения дисперсии
def get_variance(signal):
    return signal.var()

#функция определения асимметрии распределения
def get_skewness(signal):
    return signal.skew()

#функция определения эксцесса распределения
def get_kurtosis(signal):
    return signal.kurtosis()

#функция определения частоты основного тона
def get_tone_frequency(signal):
    fftData=abs(np.fft.rfft(signal))**2
    which = fftData[1:].argmax() + 1
    return which/100
```

В следующем коде отображен весь перечень функций-признаков для дальнейшего получения признаков по каждому эксперименту:

```
list_features_function = [get_peak_acceleration, get_std,
                          get_variance,get_skewness,
                          get_kurtosis,get_tone_frequency]
```

Теперь для получения признаков обработаем все эксперименты с помощью полученных выше функций:

```
experiments = signals['experiment_id'].unique()
data_stationary_features = []
for exp in experiments:
    experiment = signals[(signals['experiment_id']==exp)]
    steady_filter = (experiment['hz'] > 24) & (experiment['hz'] <
27)
```

```
    experiment_steady = experiment[steady_filter]
    feature_a1_y = []
    feature_a2_y = []
    feature_a2_x = []
    for func in list_features_function:
        a1 = func(experiment_steady['a1_y'])
        a2 = func(experiment_steady['a2_y'])
        a3 = func(experiment_steady['a2_x'])
        if type(a1) == list:
            feature_a1_y+=a1
            feature_a2_y+=a2
            feature_a2_x+=a3
        else:
            feature_a1_y.append(a1)
            feature_a2_y.append(a2)
            feature_a2_x.append(a3)

data_stationary_features.append((feature_a1_y,feature_a2_y,feature
_a2_x))
```

В примере кода взяты только три признака. Но мы можем взять все шесть признаков и анализировать, какой из них наиболее полезен. Всего может быть получено 6 x 6 = 36 признаков. Все эти признаки в дальнейшем можно использовать для задачи классификации.

```
data_stationary_features =
np.array(data_stationary_features,ndmin=3)
df_first = pd.DataFrame(data_stationary_features[:,0,:]) #
df_first - это данные признаков первого подшипника в каждом
эксперименте по ускорению a1_y
df_second = pd.DataFrame(data_stationary_features[:,1,:])#
df_second - это данные признаков второго подшипника в каждом
эксперименте по ускорению a2_y
df_third = pd.DataFrame(data_stationary_features[:,2,:]) #
df_third- это данные признаков второго подшипника в каждом
эксперименте по ускорению a2_x
```

После того как мы создали датафрейм, выведем значения признаков по первому подшипнику:

```
df_first
```

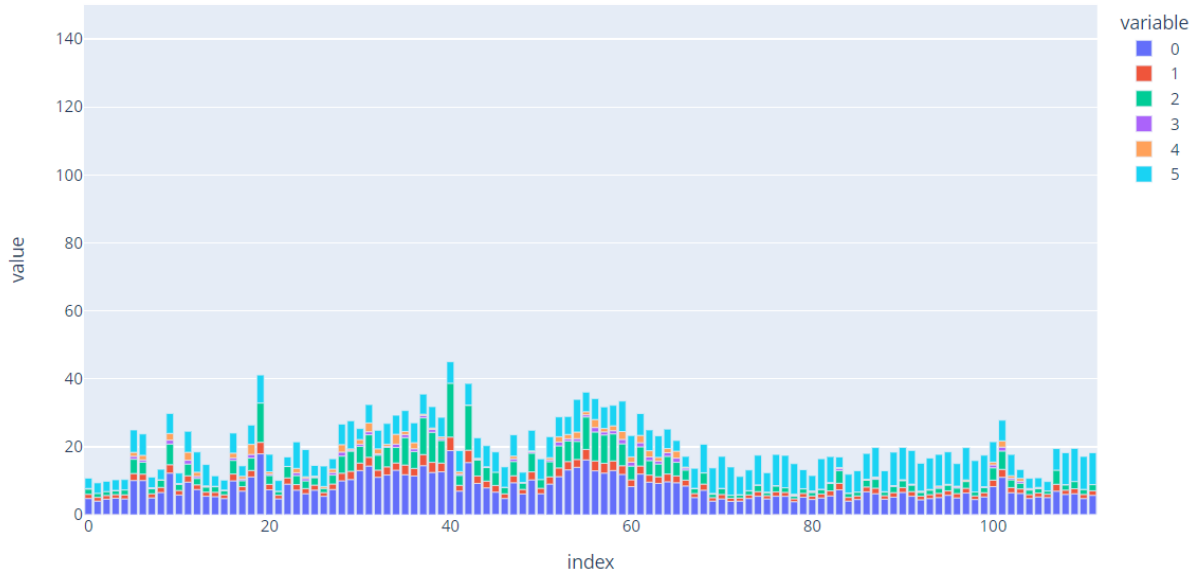
Результат — первые пять признаков относятся к статистике, а последний — к частоте:

	0	1	2	3	4	5
0	4.786797	1.240943	1.539939	0.171874	-0.211842	3.04
1	4.029721	1.089269	1.186506	-0.084968	-0.187444	3.09
2	4.597528	1.076600	1.159067	-0.052527	-0.062624	3.03
3	4.881431	1.053053	1.108920	0.135117	-0.077723	3.12
4	4.676653	1.194776	1.427489	-0.079056	-0.148830	3.07

Для всех подшипников мы можем сгенерировать диаграмму, отображающую значение всех признаков:

```
fig =px.bar(df_first)
fig.update_yaxes(range=[0,150])
fig.show()
```

Результат:



Выведем значения по второму подшипнику:

```
df_second
```

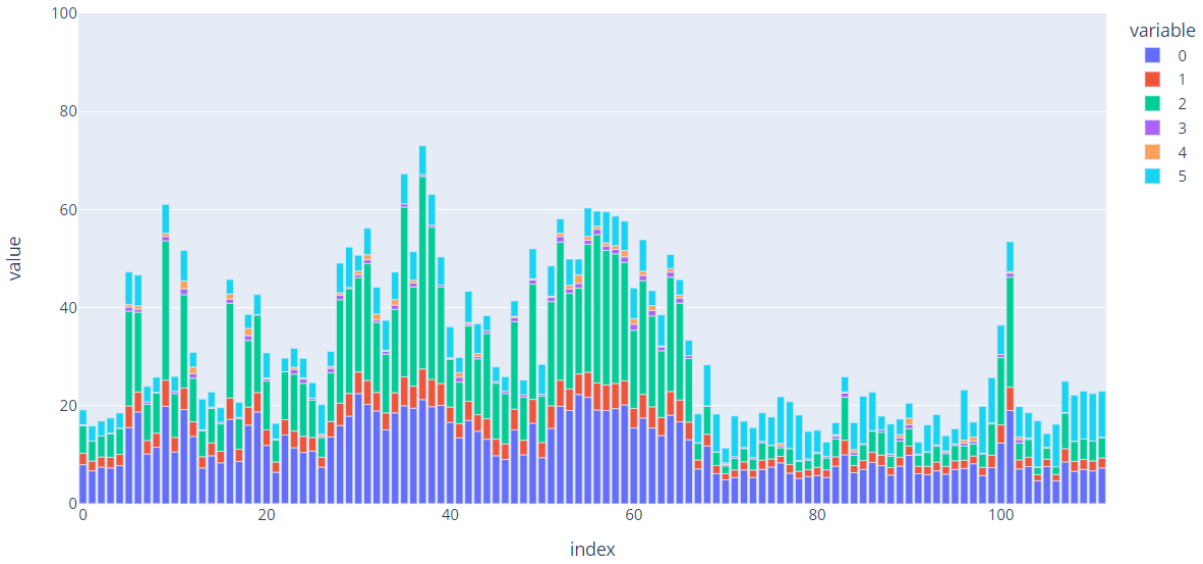
Результат:

	0	1	2	3	4	5
0	7.979513	2.362640	5.582067	0.211537	-0.559279	3.04
1	6.730058	2.010370	4.041588	-0.127512	-0.461153	3.09
2	7.494501	2.070274	4.286034	-0.257122	-0.461904	3.03
3	7.306729	2.185033	4.774368	0.122575	-0.433994	3.12
4	7.787289	2.303893	5.307921	0.045110	-0.411461	3.07

Визуализация:

```
fig =px.bar(df_second)
fig.update_yaxes(range=[0,100])
fig.show()
```

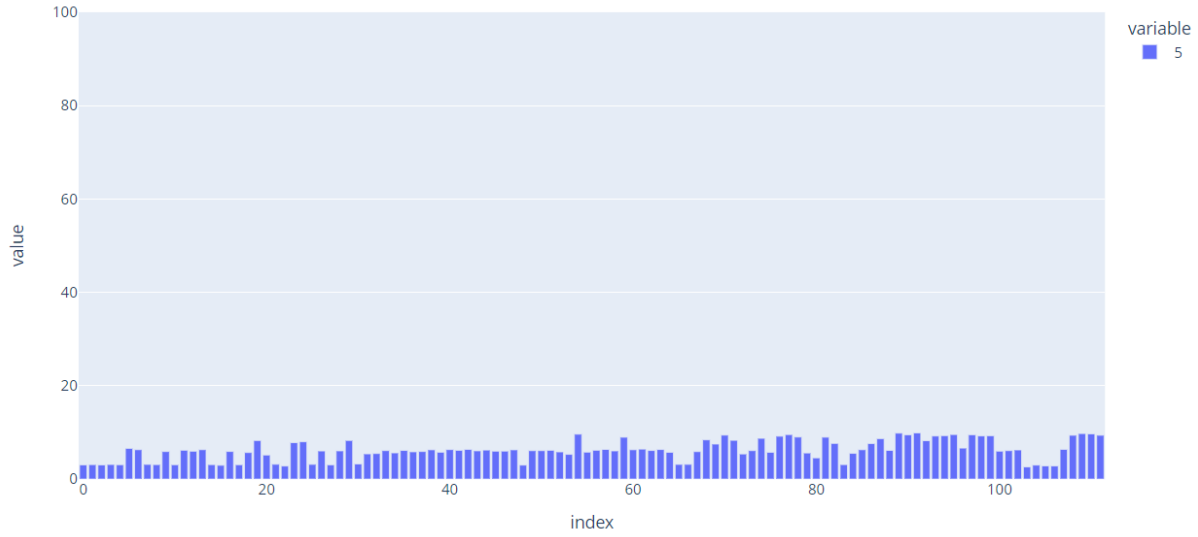
Результат:



Отдельно визуализируем признак, отвечающий за частоту:

```
fig =px.bar(df_first[5])
fig.update_yaxes(range=[0,100])
fig.show()
```

Результат визуализации:



Посмотрим номера экспериментов среди полученных признаков с выбросами:

```
data = df_second[5] # в качестве признака для проверки берем 5 признак - основной тон
```

В целом может быть построена функция, которая будет анализировать все 36 признаков.

Функция определения экспериментов с неисправными подшипниками:

```
def outliers(data):
    data_std = data.std()
    data_mean = data.mean()
    anomaly_cut_off = data_std*2

    lower_limit  = data_mean - anomaly_cut_off
    upper_limit = data_mean + anomaly_cut_off

    lower_limit, upper_limit

    outliers = data.where((data >upper_limit) | (data<lower_limit)).dropna()

    return outliers
```

В созданной функции мы получаем среднее значение, берем стандартное отклонение и вычисляем так называемую функцию отрезания, которая определяет аномалии — то, как мы разделяем данные. В примере мы берем значение стандартного отклонения и умножаем его на коэффициент, например, на 2. В данном случае

уровень отклонения от среднего может являться гиперпараметром, который можно варьировать.

Посмотрим, что получилось:

```
out = outliers(df_second[5])
out
```

Результат:

```
85      9.72
96     10.25
109     9.75
110     9.68
Name: 5, dtype: float64
```

Видим, что для функции идет отклонение по частоте для четырех подшипников с номерами 85, 96, 109 и 110.

В исходных данных мы определили, что подшипники со 101 по 110 неисправны. Получается, что мы правильно определили неисправные подшипники 109 и 110. То есть возникает перспектива развития этих признаков. Как правило, признаки используются для дальнейших методов машинного обучения, например, линейной и логистической регрессии.

В качестве дополнительного эксперимента можно попробовать «поиграть» гиперпараметром и посмотреть на полученный результат. Например, если установить гиперпараметр равным 3, результата мы не получаем. При значении 1,5 получаем намного больше неисправных подшипников. Сам по себе ответ `out` можно использовать как признак в машинном обучении.

Дополнительные материалы для самостоятельного изучения

1. Statistical functions (scipy.stats) — SciPy v1.10.1 Manual
2. Predictive Maintenance⚙️ | Kaggle
3. Understanding Confidence Intervals | Easy Examples & Formulas (scribbr.com)
4. 6 Steps to Evaluate a Statistical Hypothesis Testing (enago.com)
5. 17 Statistical Hypothesis Tests in Python (Cheat Sheet) - MachineLearningMastery.com