Статистика на Python

Вспомним такие базовые понятия, как выборка и генеральная совокупность.

Генеральная совокупность – это множество абсолютно всех объектов, которые используются для исследования.

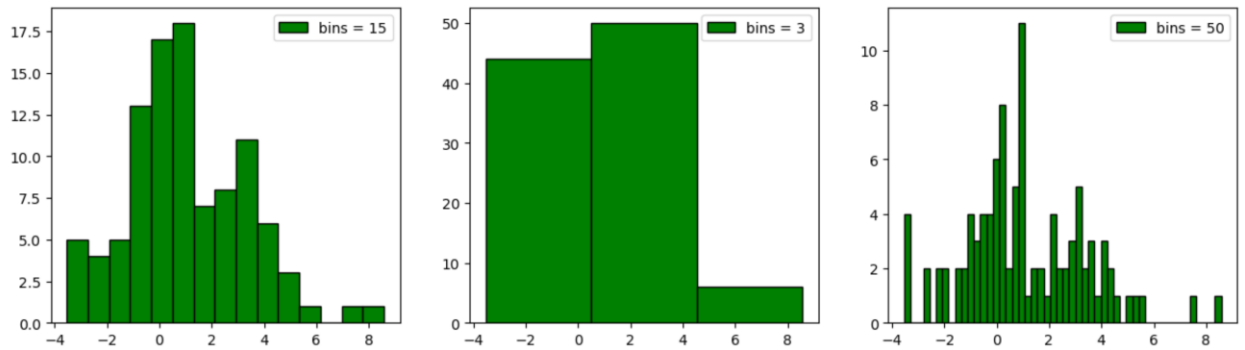
Выборка – это часть генеральной совокупности, которая выбирается для изучения. Очень важно, чтобы выборка была максимально похожа на генеральную совокупность, отражала ее свойства. Степень такой похожести называется **репрезентативностью.**

Статистика занимается тем, что исследует выборки и старается аппроксимировать полученные знания на уровне генеральной совокупности.

Например: показание температуры тела. Собрать показания температуры тела у абсолютно всех жителей России достаточно сложно. Намного проще работать с выборкой, то есть узнать показатель температуры тела у части населения России. Мы пытаемся выявить локальные закономерности исследования, которые могут быть отражены в генеральной совокупности.

Для того, чтобы исследовать форму распределения выборки, используется **гистограмма частот**. По оси абсцисс откладывается значение переменной, а по оси ординат указывается как часто значение этой переменной встречается на определенном интервале. Интервалы можно выбрать разной длины, если интервалы будут слишком большими, то гистограмма будет очень грубой и малоинформативной, если интервалы очень малы, то гистограмма будет очень разреженной. С помощью аргумента bins можно регулировать длину интервалов. Пример с использованием matplotlib:

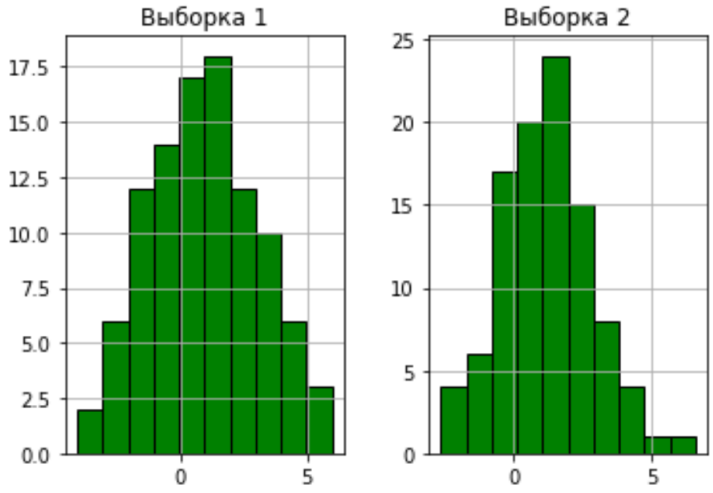
```
fig, ax = plt.subplots(1,3, figsize = (15,4))
ax[0].hist(s2, edgecolor = 'black', color = 'green',bins = 15,label = 'bins = 15')
ax[0].legend()
ax[1].hist(s2, edgecolor = 'black', color = 'green',bins = 3,label = 'bins = 3')
ax[1].legend()
ax[2].hist(s2, edgecolor = 'black', color = 'green',bins = 50,label = 'bins = 50')
ax[2].legend()
plt.show()
```



Аналогичный результат будет получен с использованием серий и датафреймов **pandas**.

```
print(dataframe)
dataframe.hist(color = 'green',edgecolor = 'black')
plt.show()
```

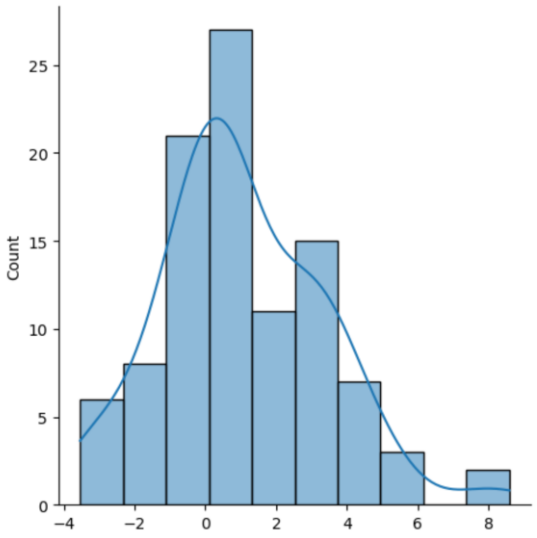
```
    Выборка 1  Выборка 2
0    1.052810   1.813691
1   -1.533349   2.382516
2   -0.635700  -0.419679
3    1.708966   1.652885
4   -1.546923   0.720153
..        ...        ...
95   3.312986   0.930325
96   0.607443  -2.639953
97  -2.633400  -0.004969
98   0.421884   1.600639
99   3.661231  -2.097273

[100 rows x 2 columns]
```



Очень часто для анализа используется библиотека визуализации **seaborn** (import seaborn as sns).

```
1 sns.displot(s2, kde = True)
```

```
<seaborn.axisgrid.FacetGrid at 0x1f6e2e68e20>
```



Непрерывную кривую, которая аппроксимирует гистограмму, можно убрать, используя kde = False. kde – это ядерное сглаживание, которое используется для гладкой оценки плотности распределения.

Одним числом данные можно описать несколькими способами:

– Меры центральной тенденции:

- **Мода.** Это значение, которое наиболее часто встречается в выборке.
- **Медиана.** Для нечетного количества элементов медиана равна центральному элементу в отсортированном массиве $(\text{sort(x)}\left[\frac{n+1}{2}\right])$.

Для четного количества элементов медиана равна среднему двух центральных элементов в отсортированном массиве $(\frac{\text{sort(x)}\left[\frac{n}{2}\right]+\text{sort(x)}\left[\frac{n+1}{2}\right]}{2})$.

- **Среднее.** Сумма значений всех элементов выборки, деленное на их количество.

Стоит отметить, что самой неустойчивой к выбросам мерой является среднее. И наоборот, самой надежной или робастной является мода.

Пример:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as sts
```

```
sp = np.array([1,2,3,2,1,2,4,5,4,100])
```

```
mean = np.mean(sp)
moda = sts.mode(sp)
med = np.median(sp)
```

```
print('Среднее = %f'%mean)
print('Мода: ',moda)
print('Медиана = %f'%med)
```

```
Среднее = 12.400000
Мода:  ModeResult(mode=array([2]), count=array([3]))
Медиана = 2.500000
```

Большая часть выборки лежит на интервале от 1 до 5, соответственно, среднее в данном примере неверно характеризует центральную тенденцию. Мода и медиана более робастны.

Библиотека **scipy** предназначена для решения различных математических задач (решение интегральный, дифференциальных

уравнений, интерполяция, оптимизация и численное решение уравнений и т.д.). Пакет **stats** содержит статистические распределения и функции.

– Меры изменчивости:

- **Размах.** Разница между максимальным и минимальным значением выборки. Очень простая мера, но она использует только два значения из всей выборки. Правильнее использовать каждое значение из выборки для расчета изменчивости данных.
- **Стандартное отклонение.** Это корень из дисперсии, которая вычисляется по формуле $\frac{\sum_1^n (x_i - \bar{x})^2}{n-1}$, где $\bar{x}$ – это среднее выборки. Это оценка для выборки. Считается, что стандартное отклонение выборки немного недооценивается, поэтому ее чуть-чуть увеличивают, делив на n-1, а не на n, как для генеральной совокупности. Для генеральной совокупности такой показатель называется **среднеквадратическим отклонением**. Этот показатель позволяет оценить, как сильно меняются данные относительно их среднего. Стандартное отклонение не устойчиво к выбросам.
- **Межквартильный размах (IQR).** Для всех выборок существуют такие отсечки, которые называются «квартили», их всего три: Q1, Q2 и Q3. Межквартильный размах – разность между Q3 (75%) и Q1 (25%), это ширина интервала, который содержит 50% данных. Это метрика полезна для описания данных, она устойчива к выбросам.

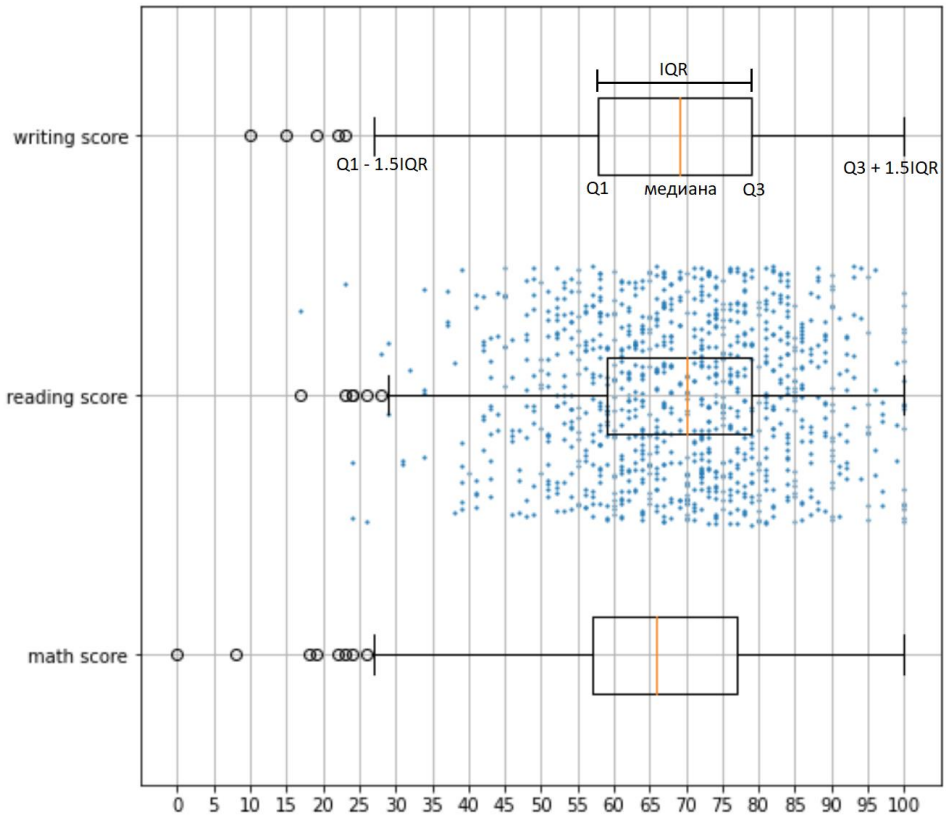
Пример:

```
std = st['writing score'].std()
raz = st['writing score'].max() - st['writing score'].min()
q1 = np.percentile(st['writing score'], 25 , interpolation = 'midpoint' )
q3 = np.percentile(st['writing score'], 75 , interpolation = 'midpoint' )
iqr1 = q3 - q1
iqr2 = sts.iqr(st['writing score'], interpolation = 'midpoint' )

print('Стандартное отклонение: ', std)
print('Размах: ', raz)
print('Межквартильный размах через numpy: ', iqr1)
print('Межквартильный размах через scipy:: ', iqr2)
```

```
Стандартное отклонение:  15.195657010869642
Размах:  90
Межквартильный размах через numpy:  21.5
Межквартильный размах через scipy::  21.5
```

Полезным графиком является «ящик с усами» или box-plot. Это диаграмма, которая используется для отображения случайной величины и несет в себе много полезной информации. Пример диаграммы для данных оценок студентов, которые содержатся в датафрейме result:

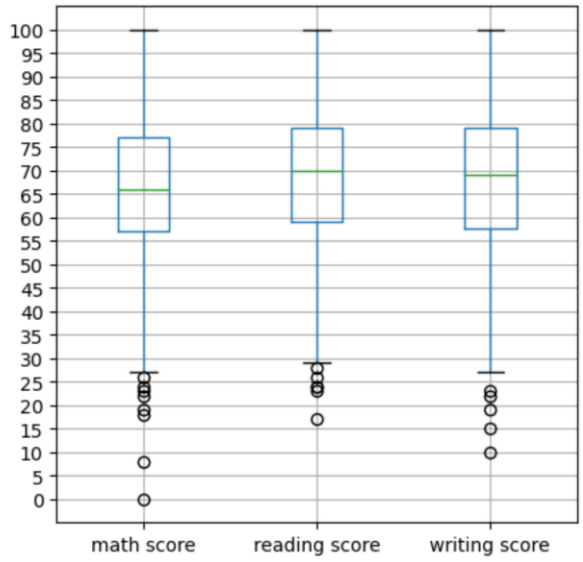
```
plt.figure(figsize=(8,8))
plt.boxplot([result['math score'],result['reading score'],result['writing score']],
            labels=['math score','reading score','writing score'],vert = False)
plt.xticks(np.arange(0,105,5))
plt.scatter(result['reading score'],rand,s=1.5)
plt.grid()
plt.show()
```



Оранжевая линия – это медиана или Q2. Границы коробки – это квартили Q1 и Q3, то есть 50 процентов выборки находится в этом диапазоне. Точки за пределами «усов» – это выбросы. Границы усов – это Q1 – 1.5IQR и Q3 + 1.5IQR в matplotlib. Еще один способ указания границ усов – это максимум и минимум выборки, тогда выбросов на такой диаграмме нет. Синие точки – это reading score, которые изображены для того, чтобы наглядно посмотреть, как box-plot соотносится с распределением данных. Если прямоугольник и усы симметричны, то данные распределяются симметрично, без перекоса.

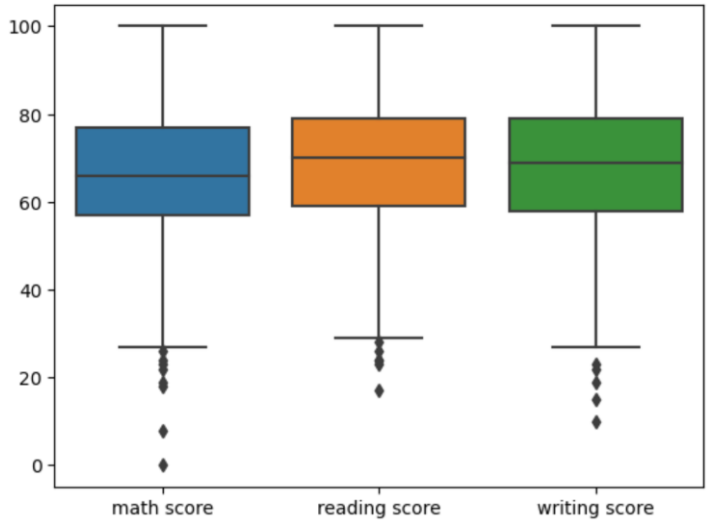
Аналогичный результат будет получен при использовании датафрейма: result.boxplot(). Для каждого столбца данных будет построен boxplot. Также можно использовать библиотеку **seaborn**.

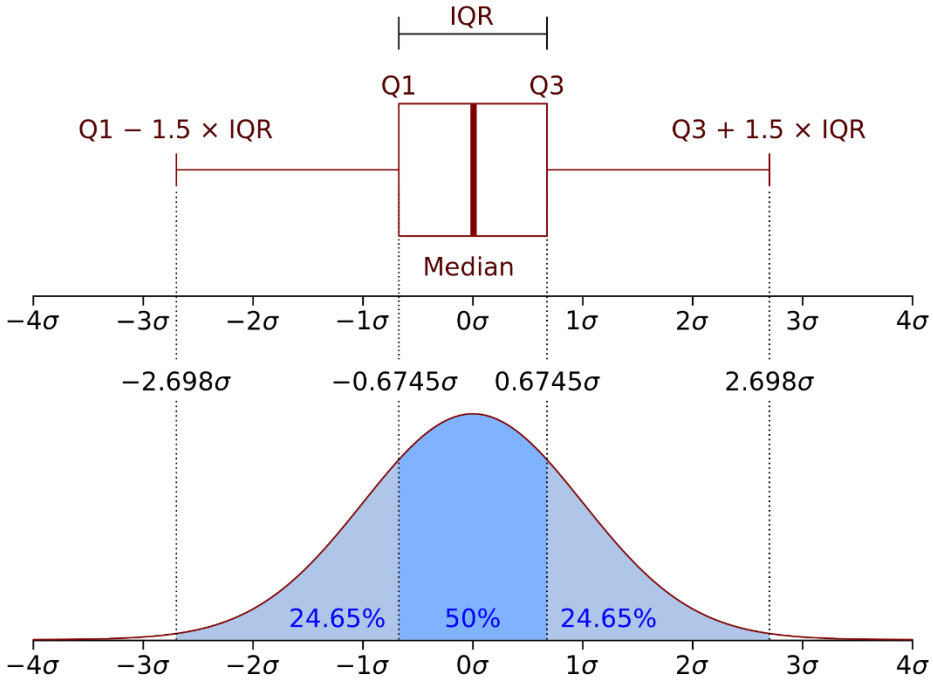
```
plt.figure(figsize=(5,5))
result.boxplot()
plt.yticks(np.arange(0,105,5))
plt.show()
```




```
sns.boxplot(data=result)
```

<AxesSubplot:>

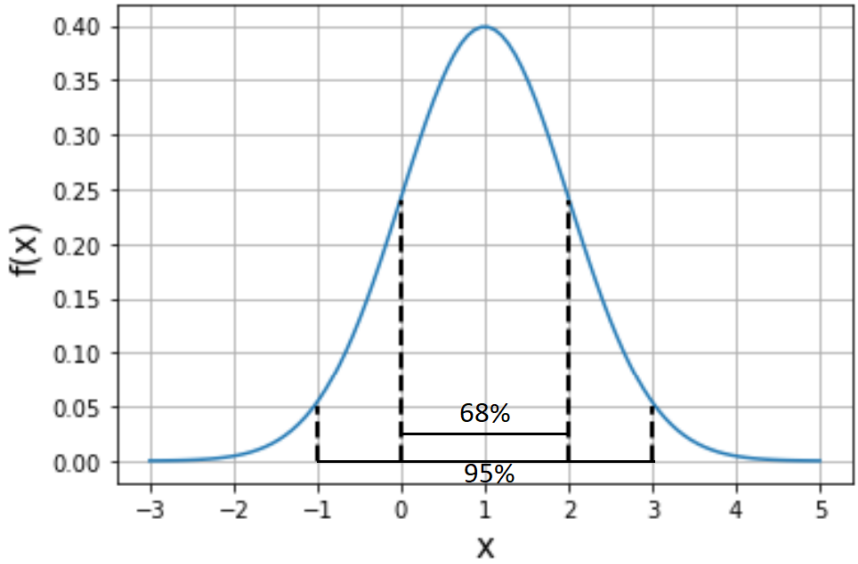




Наиболее популярным является **нормальное распределение или Гауссово распределение,** которое хорошо моделирует результат взаимодействия большого количества слабо зависимых случайных факторов. Это симметричное, унимодальное распределение, которое наиболее часто встречается в различных природных явлениях. Оно имеет два параметра – среднее и стандартное отклонение. Функция плотности вероятности нормального распределения выглядит следующим образом:

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

Функция плотности нормального распределения, где среднее и стандартное отклонение равно 1, представлено ниже.



Как правило, большинство, а именно около 68,3% выборки находятся в пределах одного среднеквадратического отклонения от среднего. В данном примере это интервал от 0 до 2. 95,45 % результатов наблюдений находятся в пределах двух среднеквадратических отклонений от среднего. И в пределах трех среднеквадратических отклонений находится 99,7 % всех наблюдений.

Одна из важнейших теорем статистики – **центральная предельная теорема**. Допустим, у нас есть некоторая генеральная совокупность с распределением F. Из этой генеральной совокупности мы получаем N выборок. Если для каждой такой выборки мы посчитаем выборочное среднее, то распределение этих средних будет **нормальным**. Чем больше будет длина выборок n, тем больше распределение средних будет унимодальным, тем лучше такое распределение будет аппроксимироваться нормальным распределением со следующими параметрами:

$$\tilde{X}_n = N\left(\mu_X, \frac{\sigma_X^2}{n}\right)$$

где μ_X – это среднее генеральной совокупности;

σ_X^2 – это дисперсия генеральной совокупности.

Среднее значение всех средних будет очень близко к среднему значению исходной генеральной совокупности.

Стандартное отклонение $\frac{\sigma_X}{\sqrt{n}}$ полученного распределения называется **стандартной ошибкой среднего (SE)**. Оно показывает на сколько в среднем выборочное значение отличается от среднего генеральной совокупности. Чем больше длина выборок n и чем меньше дисперсия исследуемых данных, тем

меньше будет стандартная ошибка среднего. Если количество элементов выборки n > 30 и выборка является репрезентативной, то мы можем вместо среднеквадратического отклонения генеральной совокупности σ_x использовать стандартное отклонение выборки sd_x для оценки стандартной ошибки. Благодаря этому, мы можем узнать стандартную ошибку среднего, взяв только одну выборку длины n из генеральной совокупности и найти ее стандартное отклонение sd_x.

Это работает не только с непрерывными распределениями, но и с дискретными.

Построение доверительных интервалов для среднего значения

Доверительный интервал – это интервал, в пределах которого с заданной вероятностью лежат оценки некоторых статистических характеристик. Две статистики $\hat{\theta}_1$ и $\hat{\theta}_2$ определяют границы доверительного интервала для параметра θ с коэффициентом доверия $1-\alpha$. Вероятность того, что θ лежит между этих двух статистик, больше или равна $1-\alpha$.

$$P\left(\hat{\theta}_1 \le \theta \le \hat{\theta}_2\right) \ge 1-\alpha,$$

где θ – это параметр, который оценивается с помощью интервала;

$1-\alpha$ – уровень доверия;

$\hat{\theta}_1$– нижний доверительный предел;

$\hat{\theta}_2$ – верхний доверительный предел.

Часто на практике коэффициент α принимают равным 0.05. Как правило, длина доверительного интервала возрастает при увеличении коэффициента доверия $1-\alpha$ и стремится к нулю с ростом размера выборки n.

Если повторять эксперимент по построению интервала бесконечно, то в $100(1-\alpha)\%$ случаев, этот интервал будет накрывать истинное значение θ. Это называется 95 % доверительный интервал.

Очень часто исследователей интересует среднее значение исследуемого признака во всей генеральной совокупности.

Например, у нас имеется выборка баллов по экзамену 1000 студентов. Ее среднее $\bar{x}$ равно 68.054 балла, стандартное отклонение *sd* равно 15.19. Но нам интересно узнать, чему равно среднее не в этой выборке, а во всей генеральной совокупности. Но собрать данные всех студентов мы не можем, поэтому абсолютно точное значение среднего генеральной совокупности получить невозможно. Но можем получить интервал, в который будет включено истинное среднее значение. Тут нам помогает центральная предельная теорема. Множество раз извлекаем из генеральной совокупности выборки длины n. Для каждой выборки рассчитываем среднее значение и свой

доверительный интервал, которые рассчитывается по следующей формуле: $\bar{x} \pm 1{,}96\ SE$. 95 % построенных доверительных интервалов содержат истинное значение среднего генеральной совокупности.

Рассчитаем стандартную ошибку среднего для примера с баллами студентов:

$$SE = \frac{sd_x}{\sqrt{n}} = \frac{15.19}{\sqrt{1000}} = 0.48.$$

Далее рассчитаем доверительный интервал, нижняя граница = $\bar{x} - 1{,}96\ SE$, верхняя граница $\bar{x} + 1{,}96\ SE$. Тогда доверительный интервал равен $[67.11, 68.99]$. Мы на 95 % уверены, что такой интервал содержит среднее значение генеральной совокупности. Так же мы можем рассчитать 99% доверительный интервал, где нижняя граница = $\bar{x} - 2.58\ SE$, верхняя граница $\bar{x} + 2.58\ SE$, такой интервал будет шире.

Статистическая проверка гипотез

Допустим, исследователи произвели новый препарат, который позволяет спортсменам улучшить свои результаты. Для проведения эксперимента было отобрано 50 спортсменов. Пусть их средний лучший результат $\bar{x}$ составляет 20 условных единиц. А стандартное отклонение *sd* равно 3. После приема препарата среднее значение результата составило 18 единиц. Возникает вопрос, действительно ли новый препарат позволяет улучшить результаты, или же это статистическая случайность и для всей генеральной совокупности спортсменов улучшения результатов не будет.

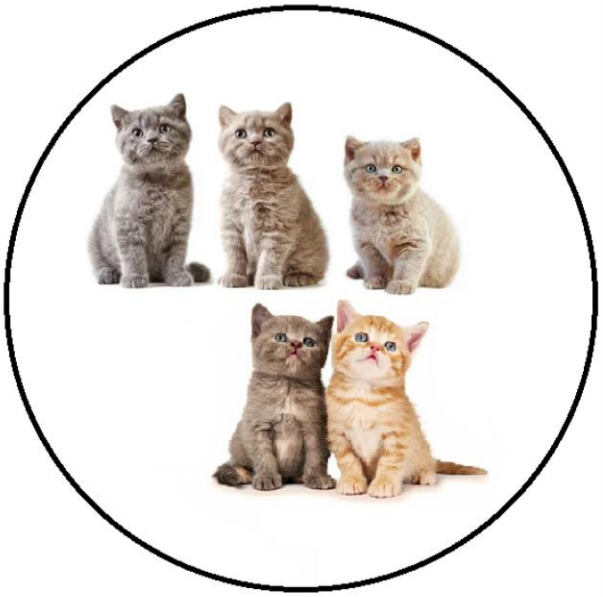
Для начала считается, что результаты никак не отличаются друг от друга, то есть прием нового препарата не дает никаких результатов – это предположение является **нулевой гипотезой (H_0, гипотеза отсутствия различий)**. С другой стороны, есть гипотеза о том, что прием нового препарата дает некие результаты, это **альтернативная гипотеза (H_1, гипотеза о значимости различий).**

Далее рассчитывается так называемый **p-уровень значимости** с помощью различных статистических критериев. Это минимальный уровень, при котором гипотеза отвергается. По сути, p-уровень значимости – это вероятность получить такие же или большие отклонения при условии нулевой гипотезы. То есть чем меньше p-уровень значимости, тем больше оснований отклонить нулевую гипотезу. Как правило, если **p-уровень значимости** менее 0.05, то нулевая гипотеза отвергается и принимается **альтернативная** гипотеза, что выборки все-таки различны. Если **p-уровень значимости** больше 0.05, то нулевая гипотеза не отвергается. То есть наши данные неплохо

согласуются с нулевой гипотезой и недостаточно оснований для ее отброса. Низший уровень статистической значимости: $p \leq 0{,}05$; достаточный: $p \leq 0{,}01$; высший: $p \leq 0{,}001$. Бывают исследования, когда этот уровень может варьироваться.

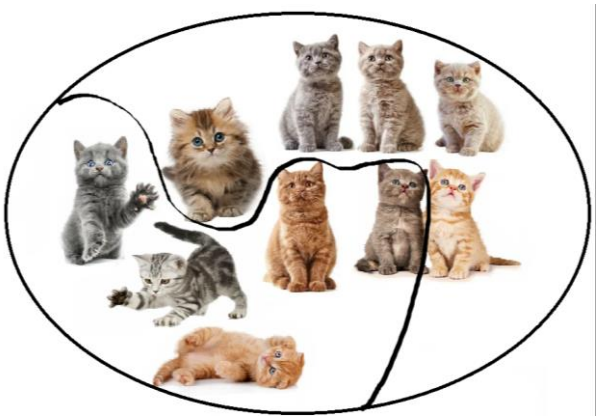
Пример. Имеется две группы котов. Одним давали новый корм, другим нет. Средний вес котов, которым не давали корм, равен 8 кг. Средний вес котов, которым давали корм, равен 5.5 кг, разница 2.5 кг. Вопрос, действительно ли при употреблении нового корма заметно снижается вес животного?

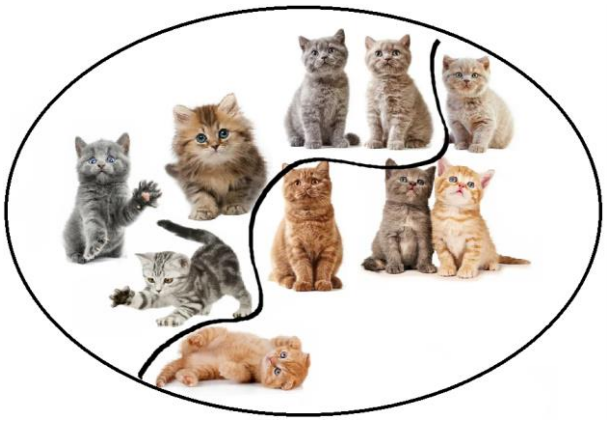
Средний вес = 8



Средний вес = 5.5

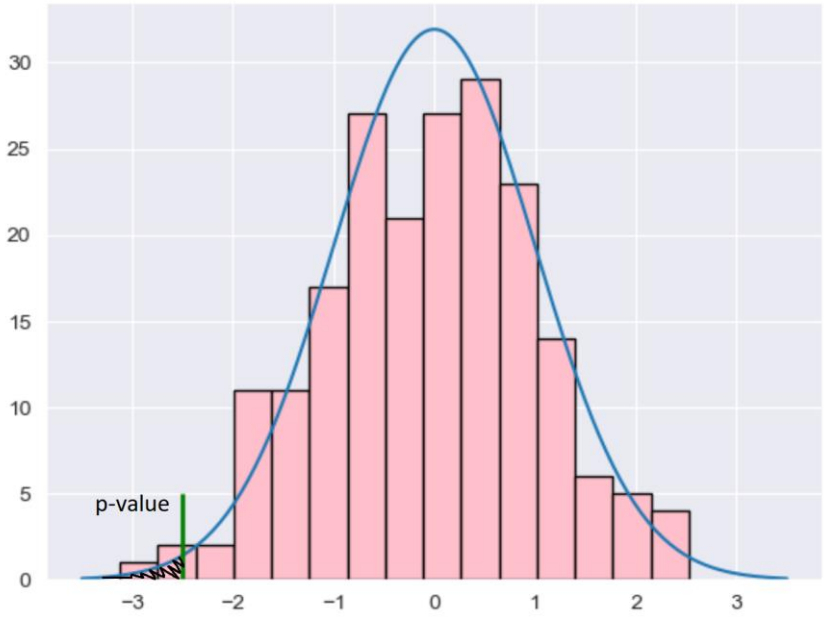
Нулевая гипотеза – прием корма никак не влияет на вес животного. Альтернативная гипотеза – потребление нового корма приводит к изменению веса. Если выполняется условие того, что корм никак не влияет на вес, то мы получили такие результаты только потому, что разделили определенным образом котов на две выборки.





Если мы поделим котов на две другие выборки, то получим разницу средних, равную 1 кг., если разделим другим образом, то получим 1.5 кг. То есть прием корма не влияет на вес, это просто разные выборки из одной группы.

Таким образом мы можем разделить исходные данные n раз и изобразить распределение разниц средних значений веса.



Имея ввиду нулевую гипотезу о том, что наша выборка изначально не имеет различий, то есть нет влияния нового корма на котов, мы смотрим, какова вероятность того, что при случайных разбиениях выборки мы получим отклонение большее или равное -2.5. Это и есть p-значение или p-уровень значимости. Если получается p-уровень значимости меньше 0.05, то мы можем отклонить нулевую гипотезу и принять альтернативную. То есть корм оказывает влияния на вес котов.

Гипотезы бывают следующих видов:

- гипотезы о законах распределения;
- гипотезы о параметрах распределения.

Статистический вывод – это утверждение, сделанное о параметрах генеральной совокупности, которое основывается на результатах исследования выборки из генеральной совокупности.

Существует два рода ошибок статистического вывода при проверке статистической гипотезы:

- **ошибка I рода (**α-ошибка**)** – отклоняется нулевая гипотеза, но она была верна. Вероятность ошибки первого рода называют уровнем значимости и обозначают α.

- **ошибка II рода** – нулевая гипотеза не отклоняется при том, что она не верна, а альтернативная гипотеза является верной. Вероятность ошибки второго рода обозначается буквой β.

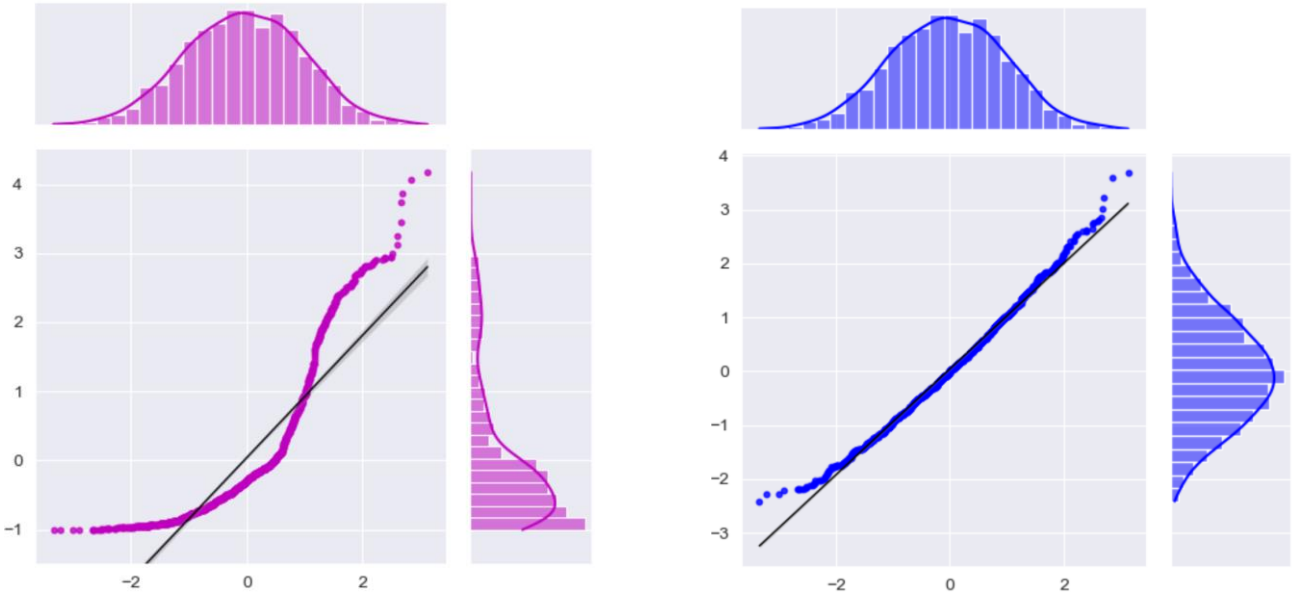
Проверка на нормальность

Достаточно часто в статистике присутствует требование к нормальному распределению при использовании различных методов. Посмотрим, как можно определить, насколько сильно распределение исследуемых данных отличается от нормального теоретического распределения.

Первый способ – это поверх гистограммы частот построить теоретическую кривую нормального распределения. Также можно воспользоваться box-plot для оценки симметричности распределения. Если медиана находится в центре прямоугольника и «усы» симметричны, то это нормальное распределение. Можно оценить моду, медиану и среднее выборки, как известно, у нормального распределения меры центральной тенденции равны.

Еще один способ – это график Q-Q plot (Quantile-Quantile plot). Представляет собой зависимость исходных значений выборки и значений идеального нормального распределения. Если наблюдается идеальная прямая, то данные следуют нормальному закону, если наблюдается отклонение выше прямой, то исходные значения выше, чем нормальные, и наоборот. Удобно использовать Q-Q plot, когда данных немного. Также Q-Q plot позволяет определить асимметрию в данных. Пример с использованием библиотеки seaborn:

```
g = sns.jointplot(x=q2, y=q12_s,
                  kind="reg", truncate=True,
                  color="b", height=5,ratio=3,
                  scatter_kws={"s": 10,}, line_kws={"lw": 1, 'color':'black'})
```



По оси абсцисс откладываются значения стандартного нормального распределения, по оси ординат – распределение исследуемой выборки. Левый график показывает, что распределение исследуемой выборки сильно отличается от нормального. На правом графике середина распределения следует нормальному закону, но его концы отклоняются от него.

Существует множество тестов для проверки распределения на нормальность, некоторые из них и наиболее часто используемые представлены ниже.

Тест Колмогорова-Смирнова (KS-тест)

Это непараметрический тест, который позволяет оценить существенность различий между распределениями двух выборок, например, оценка соответствия распределения исследуемой выборки закону нормального распределения. Критерий Колмогорова-Смирнова определяет расстояние между эмпирической функцией распределения выборки и функцией распределения эталонного распределения (это не обязательно распределение Гаусса). В случае проверки на нормальность распределения, выборки стандартизуются и сравниваются со стандартным нормальным распределением. Данный тест эффективен при размере выборки ≥ 50.

Данный тест можно использовать с помощью scipy.stats.kstest. Пример:

```
11 test_sk = stats.kstest(ch, 'norm')
12 print(test_sk)
```

```
KstestResult(statistic=0.18846450965981876, pvalue=4.381949677777384e-42)
```

ch – это стандартизированная исследуемая выборка. Статистика теста Колмогорова-Смирнова – это максимальная абсолютная разница между двумя кумулятивными распределениями. Значение статистики необходимо сравнивать с критическим значением из таблицы. Если полученное значение выше критического, то нулевая гипотеза может быть отброшена.

p-значение намного меньше 0.05, следовательно нулевая гипотеза отвергается и выборка не имеет нормального распределения.

Тест Андерсона-Дарлинга. Позволяет проверить, получена ли выборка данных из заданного распределения вероятностей.

Тест Лиллифорса. Это тест на нормальность, который основан на тесте Колмогорова–Смирнова.

Тест Шапиро-Уилка

Гипотеза о нормальности распределения отбрасывается, если значение p-уровня значимости меньше выбранного уровня α. Нулевая гипотеза – распределение выборки НЕ отличается от нормального, альтернативная

гипотеза – распределение отличается от нормального. Данный тест дает отличные результаты на небольших размерах выборок (≤ 50).

Данный тест можно использовать с помощью scipy.stats.shapiro. Пример:

```
12 test = stats.shapiro(ch)
13 print(test)
```

```
ShapiroResult(statistic=0.814687967300415, pvalue=1.150477698013898e-36)
```

statistic – это статистика критерия W, которая вычисляется по следующей формуле:

$$w = \frac{1}{s^2}[\sum_{i=1}^{n} a_{n-i+1}(x_{n-i+1} - x_i)]^2$$

$$s^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2$$

Коэффициенты a_{n-i+1} берутся из таблицы. Значение тестовой статистики сравнивается с критическим значением для данного размера выборки и ранее определенного уровня значимости. Для критических значений существуют готовые таблицы. Если значение тестовой статистики больше критического значения, нулевая гипотеза не отклоняется. Статистику теста можно интерпретировать как коэффициент корреляции, который может принимать значения от 0 до 1. Чем ближе статистика теста к 1, тем меньше отклонений фактической дисперсии от гипотетической дисперсии при нормальном распределении.

На скриншоте выше p-значение намного меньше 0.05, это высший уровень статистической значимости, следовательно, нулевая гипотеза может быть отброшена, распределение выборки не является нормальным. Если размер выборки достаточно велик, этот тест обнаруживает незначительные отклонения от нулевой гипотезы, то есть p-значение будет очень маленьким и нулевая гипотеза будет отброшена, при том, что она верна.

Предобработка данных и статистические тесты

Предобработка данных

Данные, которые содержат пропуски, дубликаты, неверный формат хранения и т.д. называются «грязными». Прежде чем начинать работу с данными, необходимо их очистить, то есть обработать пропуски, дубликаты, выбросы, и при этом не потерять важную информацию.

Далее рассмотрим основные методы поиска и исправления:

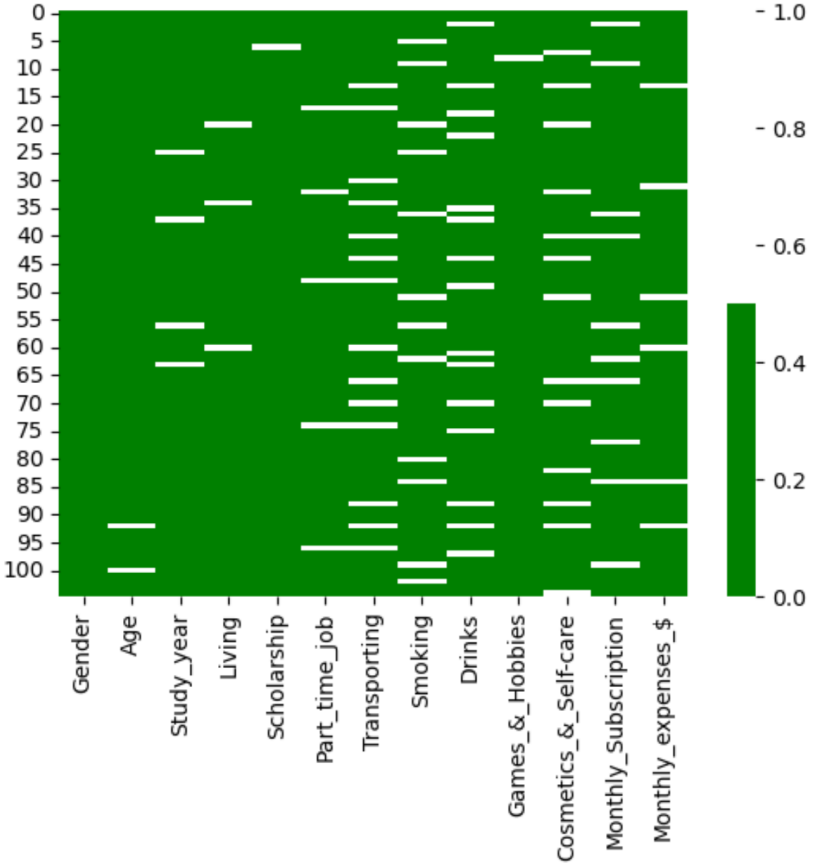
- отсутствующих данных;
- нетипичных данных – выбросов;
- неинформативных данных – дубликатов;
- несогласованных данных – одних и тех же данных, представленных в разных регистрах или форматах.

Рассмотрим три метода обнаружения пропусков в данных.

Уже используемый нами в предыдущих работах – функция isna(), которая возвращает True, если значение пропущено. Обратной ей является функция notna(), которая возвращает True, если значение не пропущено.

Еще один метод обнаружения пропусков – это построение тепловой карты. Пример с использованием библиотеки seaborn:

```
colors = ['green','white'] # пропущенные значения - белые
sns.heatmap(data.isna(), cmap=sns.color_palette(colors))
plt.show()
```



Тепловую карту удобно использовать, когда признаков в данных не очень много.

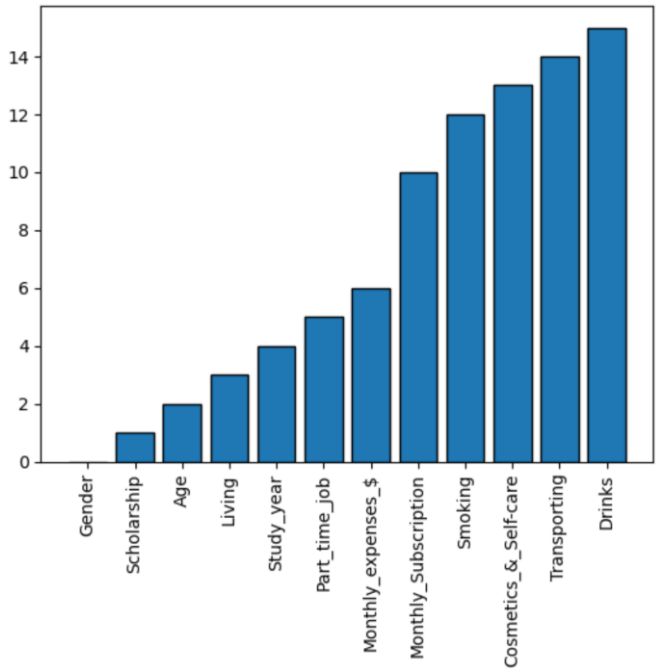
Следующий способ – это процентное содержание пропусков. Удобно использовать, когда данных очень много.

```
for column in data.columns:
    missing = np.mean(data[column].isna()*100)
    print(f" {column} : {round(missing,1)}%")
```

```
Gender : 0.0%
Age : 1.9%
Study_year : 3.8%
Living : 2.9%
Scholarship : 1.0%
Part_time_job : 4.8%
Transporting : 13.3%
Smoking : 11.4%
Drinks : 14.3%
Games_&_Hobbies : 1.0%
Cosmetics_&_Self-care : 12.4%
Monthly_Subscription : 9.5%
Monthly_expenses_$ : 5.7%
```

Еще один хороший способ визуализации для наборов с большим количеством признаков – столбчатая диаграмм пропусков. Можно изобразить количество пропусков по признакам, а можно наоборот – количество значений без пропусков. Пример вывода пропущенных значений в признаках:

```
d = dict()
for column in data.columns: #проход по столбцам
    missing = data[column].isna().sum() #кол-во пропущенных значений в столбце
    without_missing = len(data[column]) - missing #кол-во значений в столбце
    d[column] = missing #словарь признак : кол-во значений
sorted_dict = sorted([(value, key) for (key, value) in d.items()])
sort = dict(sorted_dict)
plt.bar(sort.values(),sort.keys(), edgecolor = 'black')
plt.xticks(rotation = 90)
plt.show()
```



Для каждого конкретного набора данных используются подходящие именно ему методы работы с пропусками или используются комбинации методов.

Далее рассмотрим самые распространенные методы работы с пропущенными значениями в данных.

Самый простой, но не самый лучший способ – это просто удалить все объекты, которые содержат пропуски. В таком случае можно потерять большое количество полезной информации. Удалить записи можно с помощью метода dropna().

Так же можно удалить признак, который содержит много пропусков, только если признак неинформативен для решаемой задачи. Пример:

```
data.drop('Drinks', axis = 1,inplace=True)
```

Следующий способ – это заполнение пропусков в данных. Это можно сделать несколькими способами, например: для числовых признаков можно заполнить пропуски средним значением или медианным, для категориальных данных пропуски можно заполнить самым часто встречающимся значением.

Заполнить пропуски можно с помощью метода fillna(). Пример заполнения признака возраста медианным значением:

```
median_age = data.Age.median()
data.Age.fillna(median_age, inplace=True)
```

Пример заполнения самым часто встречающимся значением категориального признака:

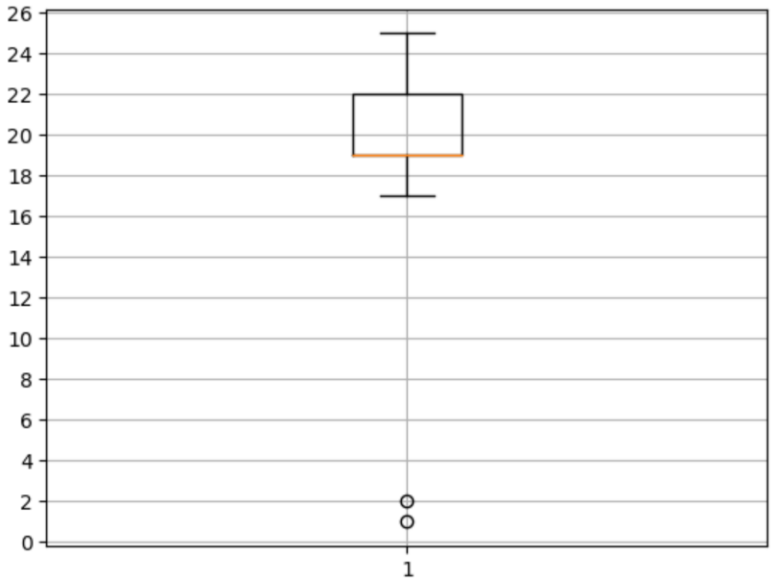
```
living = data.Living.describe().top
data.Living.fillna(living, inplace=True)
```

Можно заменить недостающие значения некоторым значением по умолчанию, таким образом мы сохраняем информацию о пропусках, она может иметь свою ценность.

Еще один метод заполнения пропусков – интерполяция. Интерполировать данные можно с использованием метода датафрема interpolate().

Далее рассмотрим методы поиска выбросов в данных. Выбросы – это значения, которые сильно отличаются от остальных объектов. Выбросы могут бать как реальными значениями, например, как рост самого высокого человека в мире, а могут быть просто ошибками.

Для нахождения выбросов можно использовать гистограмму и boxplot. Используем boxplot для определения выбросов признака возраста у студентов.



Определяются два выброса, которые являются ошибками, так как исходные данные – это данные о студентах, студенту не может быть 1 или 2 года. Можно использовать метод describe(). Для категориальных признаков можно построить гистограмму и по ней выяснить, есть ли выбросы.

Так же для поиска выбросов используется кластеризация. Алгоритмы кластеризации будут рассмотрены на следующих практических занятиях.

Работа с выбросами похожа на работу с пропущенными значениями. Все зависит от специфики задачи. Можно выбросы удалить, можно заменить на некоторое значение.

Для нахождения дублированных строк в наборе данных можно использовать метод duplicated(), который возвращает True, если строка дублируется. При необходимости удалить дублированные строки можно методом drop_dublicates().

Форматы записей в наборе данных могут отличаться. Например, форма записей дат, опечатки при вводе данных, разные регистры. Все это необходимо приводить к единому формату.

Найти такие ошибки можно с помощью метода value_counts(). Он подсчитывает количество уникальных значений.

```
1  data.Gender.value_counts()
```

```
Male       32
Female     29
Mal         2
Femal       1
```

Также используется метод unique(). Он возвращает уникальные значения.

```
1  data.Gender.unique()
```

```
array(['Female ', 'Male ', 'Femal ', 'Mal '], dtype=object)
```

Очевидно, что гендера всего два, это Female и Male. Два оставшихся значения являются опечатками, их необходимо заменить на верные варианты записи. Также с помощью метода unique() видим, что присутствует в записи лишний пробел в конце. Для удаления пробелов можно использовать метод str.strip(). Можно указать набор символов, которые необходимо удалить. Если не передавать в эту функцию аргументы, то он удалит пробелы в начале и в конце строки.

```
1 data['Gender']=data['Gender'].str.strip()
```

```
1 data.Gender.unique()
```

```
array(['Female', 'Male', 'Femal', 'Mal'], dtype=object)
```

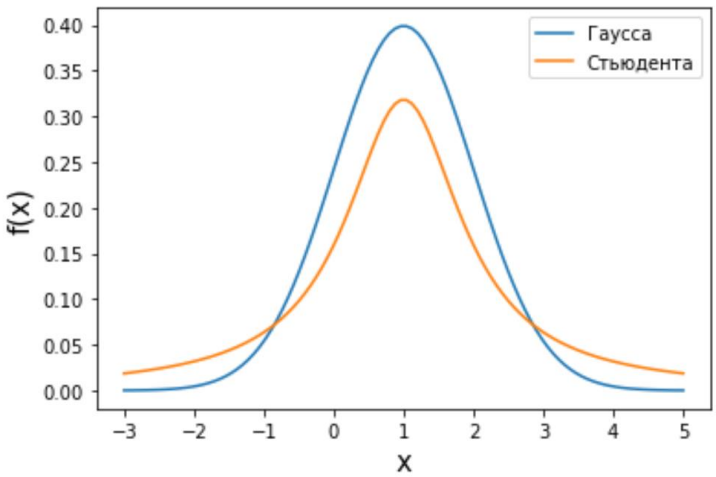
Еще используется .str.replace(‘ ‘,’’), в котором указывается какой символ на какой можно заменить.

Когда очень много уникальных элементов в признаке, удобно использовать поиск опечаток с помощью расстояния между словами. Чем меньше расстояние между словами, тем больше они похожи.

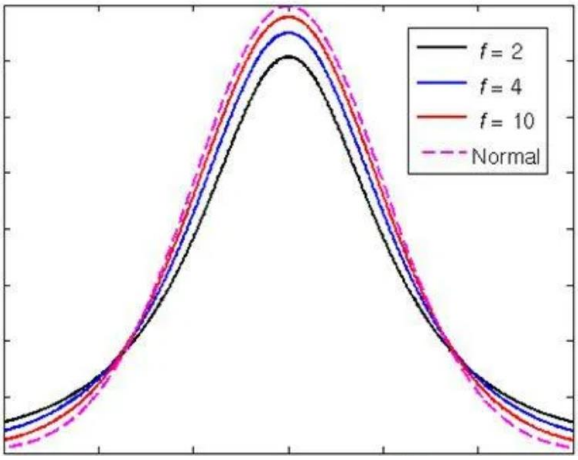
Тесты

Основываясь на центральной предельной теореме, мы узнали, как ведет себя распределение выборочных средних для разных длин выборок n. Чем больше элементов в выборке, тем лучше среднее выборочных средних будет приближено к среднему генеральной совокупности. Чем меньше выборка, тем чаще мы получаем выборочные средние, которые далеко отклоняются от среднего генеральной совокупности. Если $n < 30$, то нарушается предположение о том, что выборочные средние будут иметь нормальное распределение.

Когда $n < 30$, используется распределение Стьюдента. Распределение Стьюдента было представлено в предыдущей практической работе, рассмотрим его чуть более подробно.



t-распределение имеет более высокие хвосты, это значит, что наблюдения с большей вероятностью попадают за пределы $\pm 2\sigma$ (2 стандартных отклонения для выборок) от среднего генеральной совокупности, чем в нормальном распределении. Когда берем средние для маленьких выборок, большие отклонения от среднего генеральной совокупности мы получаем с большей вероятностью. Важным параметром распределения Стьюдента является число степеней свободы df, которое зависит от длины выборки n и вычисляется по формуле $df = n - 1$. Чем больше df (т.е. чем больше длина выборки), тем больше распределение Стьюдента стремиться к нормальному.



Если мы используем распределение Стьюдента, то значения p-уровня значимости будут выше, нежели при нормальном распределении, что не дает нам отклонять нулевую гипотезу в различных исследованиях. Число степеней свободы – это количество элементов выборке, которые могут варьироваться при расчете некоторого статистического показателя. Например, мы имеем выборку из 5 элементов и знаем ее среднее значение, тогда нам достаточно знать среднее значение и только 4 элемента выборки, чтобы однозначно определить, чему равно 5-е значение. То есть 4 элемента могут варьироваться, а 5 значение всегда будет однозначно определено. Важно понимать, сколько независимых элементов мы использовали для расчета того или иного показателя. Например, для t-распределения с 30 степенями свободы и для t-распределения с 3 степенями свободы результаты проверки гипотез будут кардинально отличаться.

Очень частый метод, применяемый в статистическом анализе – сравнение средних значений двух выборок. Целью данного метода является понимание того, действительно ли средние значения выборок отличаются друг от друга или же это статистические колебания. Самая популярная мера – это **t-критерий Стьюдента**, которая оценивает, есть ли разницы между средними значениями двух выборок. При сравнении средних выборок длины n < 30,

важным условием является нормальность распределения двух выборок. Так же необходимо, чтобы дисперсии двух выборок были примерно одинаковы (гомогенность дисперсий). Если длина выборок большая, то t-критерий Стьюдента дает достаточно точные результаты, не смотря на то, что распределение выборок отличается от нормального.

Существует три основных типа **t-теста,** которые применяются в зависимости от исходных данных:

1) **Одновыборочный t-критерий**, где сравнивается среднее одной выборки с эталонным значением.
2) **t-критерий Стьюдента для независимых выборок.** Сравнивается среднее значение двух несвязанных выборок. Например, мы взяли 50 котов и разделили их на две группы по 25 котов, одна группа будет есть новый корм, а другая нет. Это пример несвязанных выборок.
3) **Парный t-критерий Стьюдента (для зависимых выборок).** Сравниваются средние значения двух зависимых выборок. Например, у нас есть 25 котов в момент времени t, далее этим котам дается корм и в момент времени t+dt измеряется их средний вес. Получаем две выборки, одну в момент времени t, а вторую в момент времени t+dt. Эти выборки зависимы.

Рассмотрим пример сравнения среднего веса групп котов по критерию t-Стьюдента, где одна группа потребляла новый корм, а вторая нет.

Нулевая гипотеза – новый корм не работает, исследуемые выборки принадлежат к одной генеральной совокупности.

Альтернативная гипотеза – прием корма повлиял на котов, выборки принадлежат к разным генеральным совокупностям.

	Среднее	Стандартное отклонение	Кол-во котов в группе
Группа 1	8	2.1	25
Группа 2	5.5	1.9	25

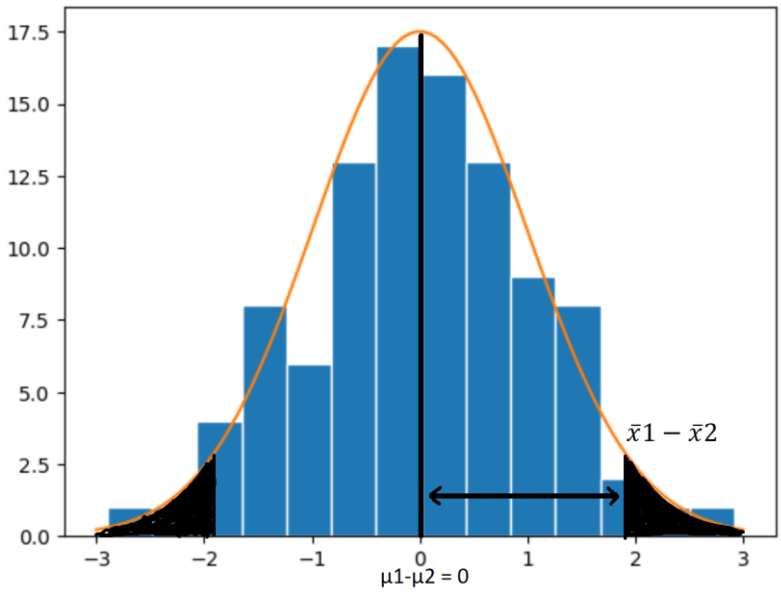
Являются ли эти различия статистически значимыми?

Помня правило центральной предельной теоремы, мы можем построить разности выборок и получить нормальное распределение средних весов котов. Но мы учитываем, что выборки у нас маленькие, тогда правильнее будет сказать, что мы получим распределение Стьюдента с числом степеней свободы $df = n_1 + n_2 - 2$. Учитывая, что изначально принимается нулевая гипотеза и выборки принадлежат к одной генеральной совокупности, тогда среднее такого распределения будет равно 0. Так как разность средних выборок стремится к разности средних двух генеральных совокупностей, а так как это одна и та же генеральная совокупность, то разность будет равна 0. Стандартное отклонение sd или стандартная ошибка SE такого распределения рассчитывается по следующей формуле:

$$\sqrt{\frac{sd_1^2}{n_1} + \frac{sd_2^2}{n_2}}$$

Это стандартное отклонение первой выборки в квадрате, деленное на количество элементов в выборке, плюс стандартное отклонение второй выборки в квадрате, деленное на количество элементов во второй выборке, и все это под корнем. То есть обе выборки влияют на стандартную ошибку.

Основываясь на всем этом, мы можем рассчитать, на сколько разность средних наших выборок отклонилась от разности средних генеральной совокупности. То есть найти p-уровень значимости – вероятность того, что мы получим такие или еще более выраженные различия разности средних значений при условии того, что верна нулевая гипотеза.



Формула t-значения выглядит следующим образом:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{sd_1^2}{n_1} + \frac{sd_2^2}{n_2}}} = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{sd_1^2}{n_1} + \frac{sd_2^2}{n_2}}}$$

где μ_1 и μ_2 – среднее значение генеральных совокупностей. Так как это одна и та же генеральная совокупность, то $\mu_1 - \mu_2 = 0$.

Эта формула аналогична формуле стандартизации выборки:

$$z_i = \frac{x_i - \bar{x}}{sd}$$

В стандартизированной выборке стандартное отклонение равно 1. Если внимательно посмотреть формулу t-значения, можно увидеть, что мы также отнимаем от значения выборки среднее, и делим на стандартное отклонение, которое является стандартной ошибкой на основе двух выборок. То есть в результате мы получаем отклонение от среднего значения в стандартных отклонениях.

Основываясь на числе степеней свободы и на полученном t-значении, мы можем рассчитать p-уровень значимости, который скажет нам какова вероятность получить такое или еще более выраженное различие между средними выборок, если верна нулевая гипотеза.

Вернемся к котам. Рассчитаем число степеней свободы и t-значение.

$$df = 25 + 25 - 2 = 48$$

$$t = \frac{8 - 5.5}{\sqrt{\frac{2.1^2}{25} + \frac{1.9^2}{25}}} = \frac{2.5}{\sqrt{0.177 + 0.14}} \approx 4$$

Получается, что мы отклонились от разности средних генеральной совокупности на 4 сигмы. Для того, чтобы найти p-уровень значимости можно

воспользоваться сайтом https://gallery.shinyapps.io/dist_calc/. p-значение равно 0.000218, следовательно, мы обнаружили статистически значимые различия, нулевая гипотеза отклоняется и коты принадлежат к разным группам, то есть корм оказывает влияние на их вес. Если разность средний достаточно большая, а стандартная ошибка маленькая, то значение t-критерия будет весьма большим. Это говорит о том, что средние выборок сильно отличаются друг от друга.

Проверка распределений на нормальность (критерий Шапиро-Уилка) с использованием scipy.stats:

```
import scipy.stats as sts
```

```
res1 = sts.shapiro(a)
res2 = sts.shapiro(b)
print(res1,'\n',res2)
```

```
ShapiroResult(statistic=0.9675695896148682, pvalue=0.5842580795288086)
 ShapiroResult(statistic=0.9567145705223083, pvalue=0.35293325781822205)
```

Тест дает p-значение выше 0.05, следовательно, выборки имеют нормальное распределение.

Проверка гомогенности дисперсии. Критерий Бартлетта – статистический критерий, позволяющий проверять равенство дисперсий нескольких (двух и более) выборок. Нулевая гипотеза предполагает, что рассматриваемые выборки получены из генеральных совокупностей, обладающих одинаковыми дисперсиями. Так же предполагается, что выборки распределены нормально.

```
res = sts.bartlett(a,b)
print(res)
```

```
BartlettResult(statistic=0.1991075475776091, pvalue=0.6554421742428356)
```

p-уровень превышает 0.05, следовательно, дисперсии выборок примерно одинаковы. Можем переходить к t критерию Стьюдента.

```
t_res = sts.ttest_ind(a,b)
print(t_res)
```

```
Ttest_indResult(statistic=-11.320889622349442, pvalue=3.739392839235914e-15)
```

p-значение намного ниже 0.05, следовательно нулевая гипотеза отвергается, выборки принадлежат к разным генеральным совокупностям и их средние значения различны.

Если распределение выборки очень сильно отличается от нормального, то можно использовать непараметрический критерий Манна-Уитни.

Биномиальный тест

Биномиальный тест показывает вероятность выполнения предполагаемой гипотезы при двух возможных исходах. Одно из распространенных применений биномиального теста – это случай, когда нулевая гипотеза

состоит в том, что две категории имеют одинаковую вероятность (например, подбрасывание монеты).

Рассмотрим пример.

Мы хотим узнать, знает ли о нашем магазине 50% населения города? Половину всех жителей опросить мы не можем. Но мы можем провести опрос небольшой выборки людей, например, 20. Из 20 человек, 6 человек посещали магазин (0.3), 14 человек впервые о нем слышат (0.7). Вопрос, является ли доля 0.3 нормальным результатом при условии, что о магазине знают 50 % населения?

Нулевая гипотеза – 50 % населения города знают о магазине.

Воспользуемся библиотекой

```
1 bin_test = sts.binom_test(6, n = 20, p = 0.5)
2 print(bin_test)
```

```
0.11531829833984375
```

p-значение больше, чем 0.05, следовательно, нулевая гипотеза не отклоняется.

Критерий хи-квадрат (χ^2) Пирсона

Ранее мы работали с нормальным распределением, но на практике точно не известно по какому закону распределены данные. В связи с этим решается вопрос о соответствии эмпирического распределения (полученная выборка) к теоретическому распределению вероятностей (которые мы определяем самостоятельно) и называется такое соответствие **критерием согласия**.

Одним из первых был изобретён критерий $\boldsymbol{\chi^2}$ **(хи квадрат)**, который остаётся популярным до сих пор. В основном представленный метод применяется для анализа таблиц сопряжённости, которые содержат категориальные данные (пол, производитель смартфона и т.д.).

Введём обозначение частот:

- Наблюдаемые частоты (**O**bserved) — количество объектов в каждой категории (данные из выборки)
- Ожидаемые частоты (**E**xpected) — количество наблюдений, при условии выполнения нашего предположения о распределении.

Критерий $\boldsymbol{\chi^2}$ Пирсона — это непараметрический метод, который позволяет оценить статистическую значимость различий двух или нескольких относительных показателей (частот, долей).

Ниже представлена формула Хи-квадрат для набора значений с набором эталонов:

$$\chi_n^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i},$$

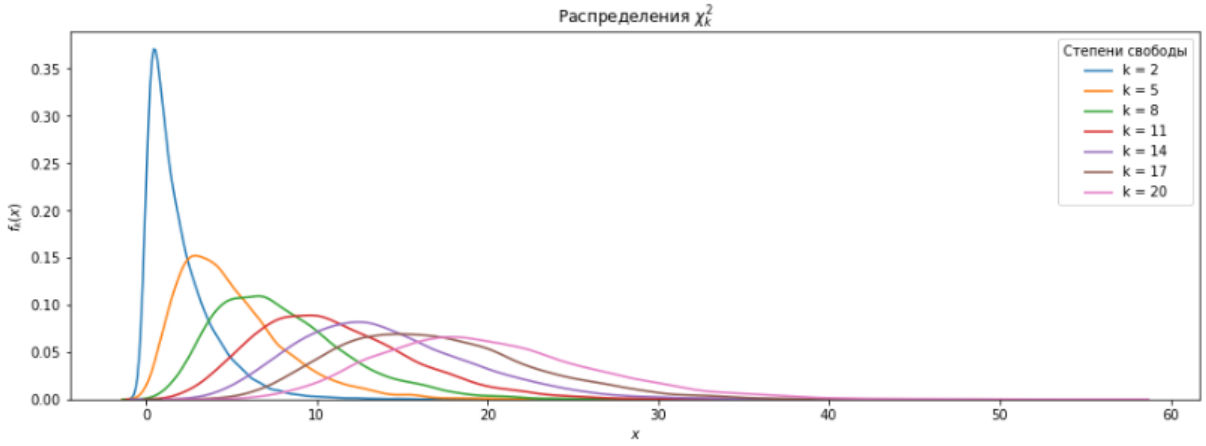
где O — наблюдаемое значение, E — ожидаемое значение.

С её помощью мы можем охарактеризовать одним числом на сколько сильно наблюдаемые частоты признака отклоняются от ожидаемых частот признака — это называется **расстояние Пирсона**.

Если частоты действительно соответствуют ожидаемым, то значение статистики Хи-квадрат будет относительно не большим (отклонения находятся близко к нулю). Большое значение статистики свидетельствует в пользу существенных различий между частотами.

Распределение хи квадрат – это семейство распределений, каждое из которых зависит от параметра степеней свободы. Или более формально: **распределение хи-квадрат с k степенями свободы** — это распределение суммы квадратов k независимых стандартных нормальных случайных величин.

На графике ниже представлено распределение хи-квадрат при многократном вычислении расстояния Пирсона и откладыванием его по оси абсцисс, а по оси ординат записывается частота встречаемости подсчитанного значения. С увеличением степеней свободы распределение хи-квадрат стремится к нормальному. Это объясняется действием центральной предельной теоремы, согласно которой сумма большого количества независимых случайных величин имеет нормальное распределение.



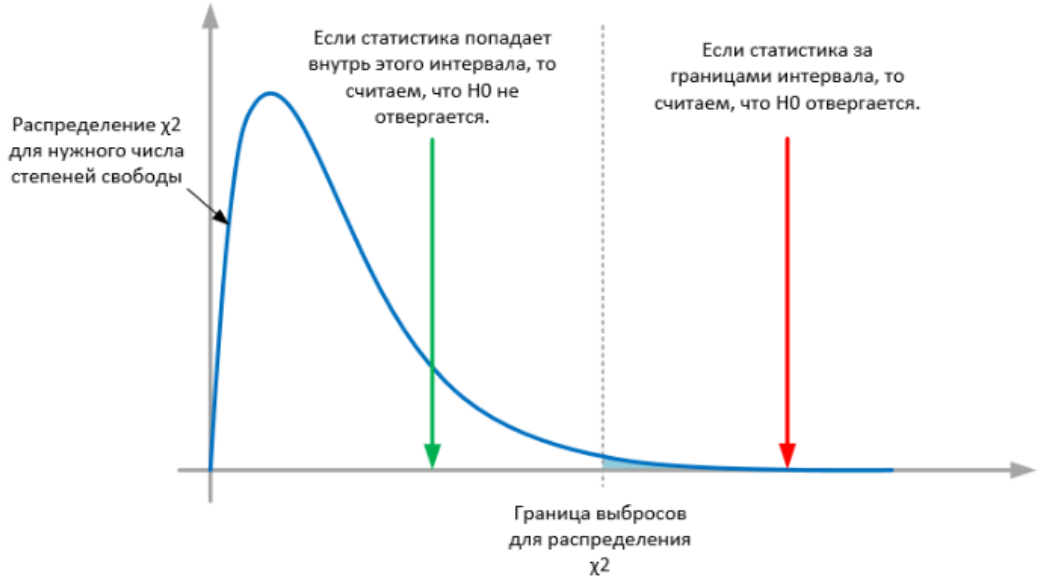
Критерий Пирсона применяется в тестах:

- На **гомогенность** — непараметрический, одновыборочный тест, который сопоставляет эмпирическое распределение признака с теоретическим распределением;
- На **независимость** — непараметрический, одновыборочный тест, который проверяет наличие связи между двумя категориальными переменными.

В тесте на гомогенность ставятся следующие гипотезы:

- H_0 между наблюдаемым распределением и эталонным различий нет
- H_1 между наблюдаемым распределением и эталонным есть различия

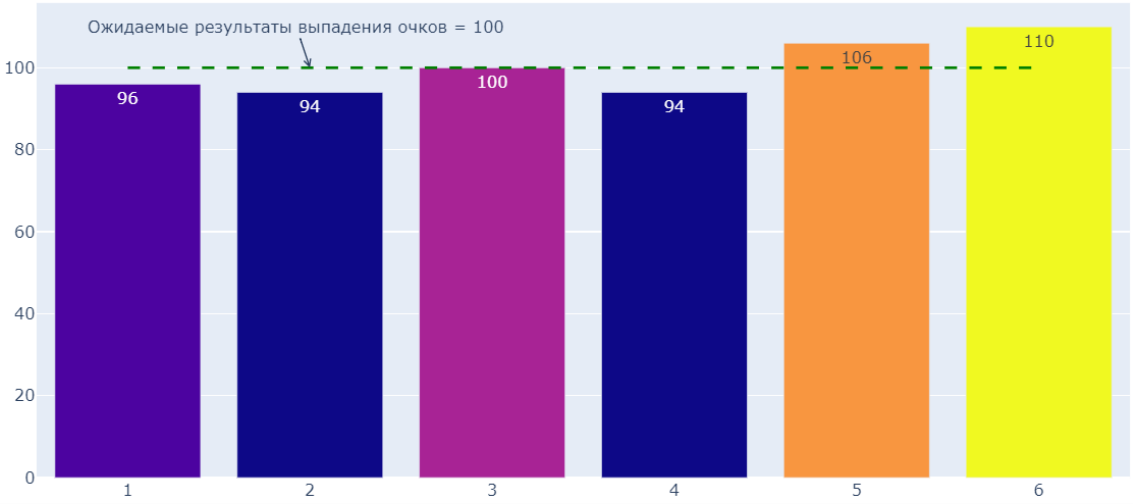
Уже на данном этапе можно воспользоваться специальной таблицей значений хи-квадрат для различных p, в зависимости от числа степеней свободы. То есть подсчитать расстояние Пирсона, сопоставить его с числом степеней свободы и отбросить или применить нулевую гипотезу. Или рассчитать p-уровень значимости.



Рассмотрим простой пример с игральной костью. К примеру, мы подбрасываем кубик 600 раз, тогда вероятность выпадения любой стороны равна 1/6, следовательно ожидание выпадения каждой из сторон равна 100.

```python
attempts = 600
np.random.seed(2718281828)
throws = pd.DataFrame(np.random.randint(1,7,attempts))
print(throws.head(),'\n') #вывод первых 5 бросков кубика
throws = throws.groupby(throws.columns.tolist(),as_index=False).size()
throws = throws.rename({0:'Points','size':'Observed'}, axis = 'columns')
throws['Expected'] = 100
throws.head(6) #количество выпавших значений кубика
```

	Points	Observed	Expected
0	1	96	100
1	2	94	100
2	3	100	100
3	4	94	100
4	5	106	100
5	6	110	100




```python
import scipy.stats as st
#попробуем расчитать критерий Хи-вадрат самостоятельно:
#Рассчитываем число степеней свободы
k = len(throws.Points) - 1 #так как в кубике 6 вариантов выпадения значений
#то число степеней свободы 6-1 = 5

#следуя формуле Хи-квадрат:
difference = throws.Observed - throws.Expected
throws['Difference'] = difference #записываем разность между полученными и ожидаемыми значениями
throws['SquaredDif'] = throws['Difference']**2 #возводим их в квадрат
throws['SquaredDif/Epxected'] = throws['SquaredDif']/throws.Expected #делим полученное значение на ожидание
print(throws)
statistic = throws['SquaredDif/Epxected'].sum() #суммируем последний столбец, получая статистику
print('\nСтатистика: ',statistic)

#воспользуемся функцией распределения cdf и получим значение
#вероятности получить похожее значение или выше найденной статистики
pval = 1 - st.chi2.cdf(statistic, k)
print('Вероятность получить похожее значение или выше, исходя из полученной статистики: ',pval)

st.chisquare(throws['Observed'], throws['Expected']) #встроенный метод Хи квадрат
```

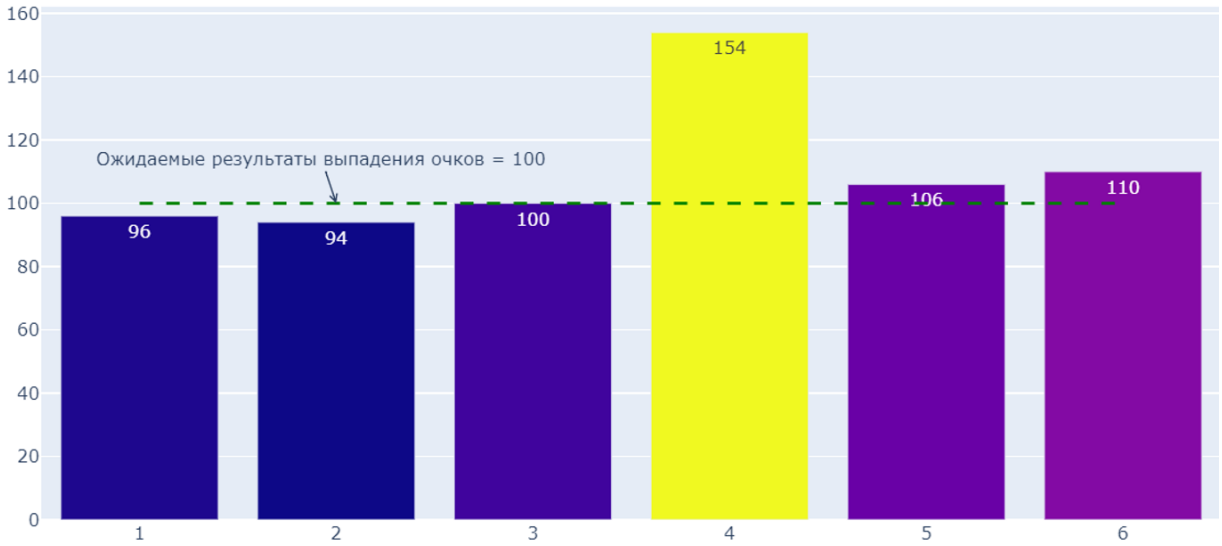
```
   Points  Observed  Expected  Difference  SquaredDif  SquaredDif/Epxected
0       1        96       100          -4          16                 0.16
1       2        94       100          -6          36                 0.36
2       3       100       100           0           0                 0.00
3       4        94       100          -6          36                 0.36
4       5       106       100           6          36                 0.36
5       6       110       100          10         100                 1.00

 Статистика:  2.24
Вероятность получить похожее значение или выше, исходя из полученной статистики:  0.8150376319793067

Power_divergenceResult(statistic=2.24, pvalue=0.8150376319793067)
```

Основываясь на p-value делаем вывод, что распределение равномерное.

Представим, что кубик «заговорён» на более частое выпадение числа 4:

```
#представим, что кубик "заговорён" на выпадение  4 в большей степени.
throws4 = throws.copy(deep=True)
throws4 = throws4.drop(['Difference','SquaredDif','SquaredDif/Epxected'], axis = 1)
throws4.at[3,'Observed'] = throws4['Observed'][3] + 60
throws4['Expected'] = 110
print(throws4)
```

```
   Points  Observed  Expected
0       1        96       110
1       2        94       110
2       3       100       110
3       4       154       110
4       5       106       110
5       6       110       110
```




```
import scipy.stats as st

st.chisquare(throws4['Observed'], throws4['Expected'])
#встроенный метод Хи квадрат
```

```
Power_divergenceResult(statistic=22.763636363636365, pvalue=0.0003745547927853111)
```

В таком случае p-value оказывается 0,0003, что гораздо меньше, чем 0,05. Исходя из этого мы видим, что распределение становится не равномерным и даёт повод сомневаться в достоверности игральной кости.

Хи-квадрат работает корректно в случае, когда количество всех частот превышает 50, а минимальное ожидаемое значение частоты не меньше 5. Если в какой-либо категории ожидаемая частота менее 5, но при этом сумма всех частот превышает 50, то такую категорию объединяют с ближайшей, чтобы их общая частота превысила 5. Если это сделать невозможно, или сумма частот меньше 50, то следует использовать более точные методы проверки гипотез.

Тест χ^2-тест на независимость отличается от предыдущего теста постановкой гипотез:

- H0: категориальные переменные A и B независимы;
- H1: категориальные переменные A и B связаны между собой.

Пример:

```
#χ²-тест на независимость
#составим таблицу сопряжённости
'''На четырёх видах оборудования разработаны детали трёх уровней качества «высокое», «среднее», «низкое».
Данные приведены в таблице'''

data = pd.DataFrame({'Низкое': [2, 1, 3, 2],
                     'Среднее': [10, 10, 15, 13],
                     'Высокое': [20, 21, 22, 20]})
data.index = ['№1','№2','№3','№4']

#Нужно проверить зависит ли распределение качества деталей от вида оборудования, на котором оно было сделано.
data.head(6)
```

	Низкое	Среднее	Высокое
№1	2	10	20
№2	1	10	21
№3	3	15	22
№4	2	13	20

```
st.chi2_contingency(data)[:3]
```

```
(1.3971398988812402, 0.9660321696141388, 6)
```

p-значение больше 0.05, качество детали не зависит от станка, на котором деталь сделана.

Точный критерий Фишера

Если выборка очень мала, и нет возможности набрать еще данных, например, если исследуется очень редкое заболевание, то некорректно использовать критерий χ^2. Для таких выборок используется точный критерий Фишера. Особое место отводится точному критерию Фишера в медицине. Это важный метод обработки медицинских данных, нашедший свое применение во многих научных исследованиях. В данном тесте рассчитывается вероятность получить таблицу сопряженности с такими или с еще более выраженными отклонениями при условии нулевой гипотезы. Если p-значение больше критического, принимается нулевая гипотеза и делается вывод об отсутствии статистически значимых различий частоты исхода в зависимости от наличия некоторого фактора.

Точный критерий Фишера можно реализовать, используя функцию scipy.stats.fisher_exact().

Практическая работа

1. Загрузить данные из файла “insurance.csv”.
2. С помощью метода describe() посмотреть статистику по данным. Сделать выводы.
3. Построить гистограммы для числовых показателей. Сделать выводы.
4. Найти меры центральной тенденции и меры разброса для индекса массы тела (bmi) и расходов (charges). Отобразить результаты в виде текста и на гистограммах (3 вертикальные линии). Добавить легенду на графики. Сделать выводы.
5. Построить box-plot для числовых показателей. Названия графиков должны соответствовать названиям признаков. Сделать выводы.
6. Используя признак charges или imb, проверить, выполняется ли центральная предельная теорема. Использовать различные длины выборок n. Количество выборок = 300. Вывести результат в виде гистограмм. Найти стандартное отклонение и среднее для полученных распределений. Сделать выводы.
7. Построить 95% и 99% доверительный интервал для среднего значения расходов и среднего значения индекса массы тела.
8. Проверить распределения следующих признаков на нормальность: индекс массы тела, расходы. Сформулировать нулевую и альтернативную гипотезы. Для каждого признака использовать KS-тест и q-q plot. Сделать выводы на основе полученных p-значений.

9. Загрузить данные из файла “ECDCCases.csv”.
10. Проверить в данных наличие пропущенных значений. Вывести количество пропущенных значений в процентах. Удалить два признака, в которых больше всех пропущенных значений. Для оставшихся признаков обработать пропуски: для категориального признака использовать заполнение значением по умолчанию (например, «other»), для числового признака использовать заполнение медианным значением. Показать, что пропусков больше в данных нет.
11. Посмотреть статистику по данным, используя describe(). Сделать выводы о том, какие признаки содержат выбросы. Посмотреть, для каких стран количество смертей в день превысило 3000 и сколько таких дней было.
12. Найти дублирование данных. Удалить дубликаты.
13. Загрузить данные из файла “bmi.csv”. Взять оттуда две выборки. Одна выборка – это индекс массы тела людей с региона northwest, вторая

выборка – это индекс массы тела людей с региона southwest. Сравнить средние значения этих выборок, используя t-критерий Стьюдента. Предварительно проверить выборки на нормальность (критерий Шопиро-Уилка) и на гомогенность дисперсии (критерий Бартлетта).

14. Кубик бросили 600 раз, получили следующие результаты:

N	Количество выпадений
1	97
2	98
3	109
4	95
5	97
6	104

С помощью критерия Хи-квадрат проверить, является ли полученное распределение равномерным. Использовать функцию scipy.stats.chisquare().

15. С помощью критерия Хи-квадрат проверить, являются ли переменные зависимыми.

Создать датафрейм, используя следующий код:

```
data = pd.DataFrame({'Женат': [89,17,11,43,22,1],
                     'Гражданский брак': [80,22,20,35,6,4],
                     'Не состоит в отношениях': [35,44,35,6,8,22]})
data.index = ['Полный рабочий день','Частичная занятость','Временно не работает','На домохозяйстве','На пенсии','Учёба']
```

Использовать функцию scipy.stats.chi2_contingency().

Влияет ли семейное положение на занятость?

16. Оформить отчет о проделанной работе, написать выводы.