СБЕР УНИВЕРСИТЕТ

# Школа Data scientist

## Занятие 4

# Анализ данных в Python

Тема 1

# Disclaimer

Все формулировки далее нестрогие, за более строгими определениями обращайтесь к специализированной литературе

# План занятия

- Библиотека Pandas
  - Чтение файлов
  - Индексы, операции с колонками и строками, мультииндекс, навигация по таблице
- Базовые понятия теории вероятность и математической статистики
- Основные виды распределений случайных величин


# Библиотека Pandas

# Что такое Pandas?

- Библиотека Python для обработки и анализа данных
- Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временны́ми рядами
- Основные возможности библиотеки:
  - Объект DataFrame для манипулирования индексированными массивами двумерных данных
  - Объект Serries  для манипулирования индексированными массивами одномерных данными
  - Инструменты для обмена данными между структурами в памяти и файлами различных форматов
  - Встроенные средства совмещения данных и способы обработки отсутствующей информации
  - Переформатирование наборов данных, в том числе создание сводных таблиц
  - Срез данных по значениям индекса, расширенные возможности индексирования, выборка из больших наборов данных
  - Вставка и удаление столбцов данных
  - Возможности группировки «разделение, изменение, объединение»
  - Слияние и объединение наборов данных
  - Иерархическое индексирование (работа с данными высокой размерности в структурах меньшей размерности)
  - Работа с временными рядами (формирование временных периодов и изменение интервалов и т. п.)

# Что такое Pandas?

- **import pandas as pd**
- **pd.DataFrame** (data=None, index=None, columns=None,  dtype=None, copy=False)

| | color | object | price |
|---|---|---|---|
| **0** | blue | ball | 1.2 |
| **1** | green | pen | 1.0 |
| **2** | yellow | pencil | 0.6 |
| **3** | red | paper | 0.9 |
| **4** | white | mug | 1.7 |

- **pd.Series** (data=None, index=None, dtype=None, name=None, copy=False)

| | |
|---|---|
| **0** | blue |
| **1** | green |
| **2** | yellow |
| **3** | red |
| **4** | white |

# Pandas: чтение файлов

| Format Type | Data Description | Reader | Writer |
|---|---|---|---|
| text | CSV | read_csv | to_csv |
| text | JSON | read_json | to_json |
| binary | MS Excel | read_excel | to_excel |
| text | HTML | read_html | to_html |
| text | XML | read_xml | to_xml |

https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html

# Pandas: индексы, операции с колонками и строками, мультииндекс, навигация по таблице

| | subject | Bob | | Guido | | Sue | |
|---|---|---|---|---|---|---|---|
| | type | HR | Temp | HR | Temp | HR | Temp |
| year | visit | | | | | | |
| 2019 | 1 | 31.0 | 38.7 | 32.0 | 36.7 | 35.0 | 37.2 |
| | 2 | 44.0 | 37.7 | 50.0 | 35.0 | 29.0 | 36.7 |
| 2021 | 1 | 30.0 | 37.4 | 39.0 | 37.8 | 61.0 | 36.9 |
| | 2 | 47.0 | 37.8 | 48.0 | 37.3 | 51.0 | 36.5 |

СБЕР

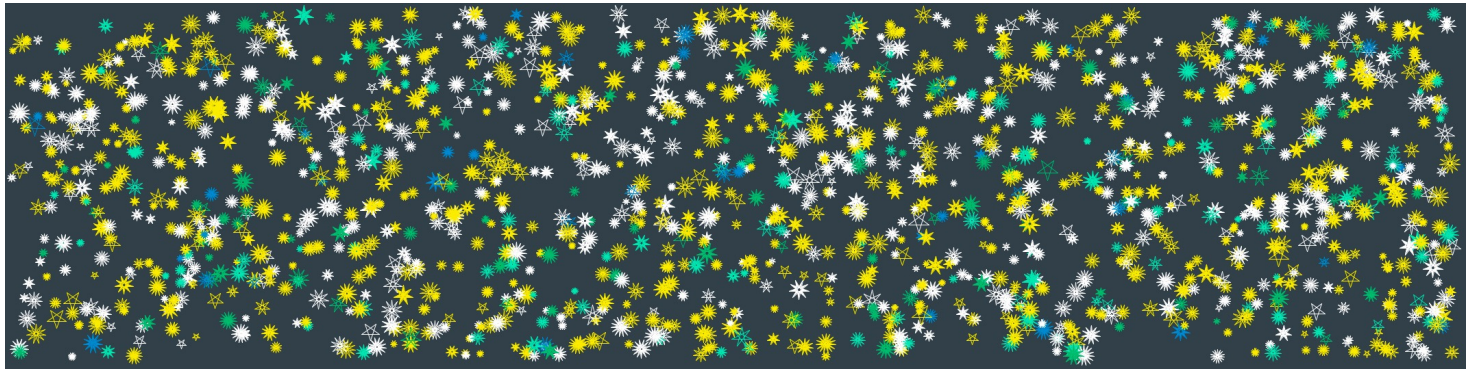
# Практика? Практика!

# Вероятность и случайные величины

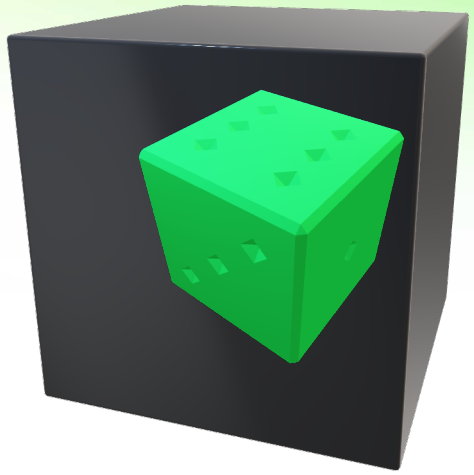
# Случайность в теории вероятностей и статистике

Ввиду того, что окружающий мир сложен, очень часто невозможно описать то или иное явление простым детерминированным законом

Когда на результат эксперимента влияет множество случайных, трудно описываемых факторов на помощью приходит теория вероятностей и статистика



# Пример случайной величины



Подбрасывание кубика.

Чёрный ящик, который по неизвестным законам возвращает нам числа от 1 до 6. Этот чёрный ящик – **случайная величина**. А числа, которые он нам возвращает (или генерируемые события) – **реализации случайной величины**. Набор реализаций случайной величины - **выборка**

# Пример случайной величины

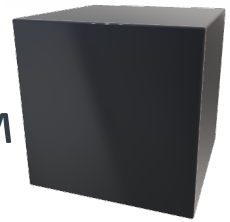
Подбрасывание кубика.

В каждое следующее подбрасывание мы не можем предугадать исход. Однако, если продолжать этот процесс достаточно долго, то мы можем обнаружить определенные закономерности. Например, что каждое число на кубике будет выпадать примерно одинаковое количество раз.

Если мы продолжим этот процесс до бесконечности, то каждому событию можно будет сопоставить его **вероятность*** – долю испытаний завершившихся соответствующей реализацией случайной величины.

*степень уверенности, возможность реализации, частота проявления…

# Случайность в теории вероятностей и статистике



**Теория вероятностей** изучает модели случайных величин и свойства этих моделей, а также позволят выводы от какие события нас ожидают в будущем

**Статистика и анализ данных** пытаются по свойствам конечных выборок определить свойства случайной величины, чтобы понять, как эта случайная будет вести себя в будущем

Осуществить такой переход позволяет **закон больших чисел**: на большой выборке частота события хорошо приближает его вероятность.

# Свойства вероятности

1) 0 <= P($A$) <= 1, то есть вероятность любого события лежит на отрезке от нуля до единицы.

2) P(∅) = 0 — событие ∅, вероятность которого равна нулю, называется невозможным. **Но** если $\mathrm{P}(A) = 0 \not\Rightarrow A = \emptyset$ !!!

3) P($\bar{A}$) + P($A$) = 1. Для события А всегда можно определить событие «не А», которое соответствует событию «А не произошло». Вероятности таких событий в сумме дают единицу.

# Реализация в Python

| Random | Описание | Numpy |
|---|---|---|
| **random.randrange** (start, stop, step) | возвращает случайно выбранное число из последовательности | **numpy.random.randint** (*low*, *high=None*, *size=None*, *dtype=int*) |
| **random.randint**(A, B) | случайное целое число N, A ≤ N ≤ B | |
| **random.choice**(sequence) | случайный элемент непустой последовательности | **numpy.random.choice** (*a*, *size=None*, *replace=True*, *p=None*) |
| **random.sample**(population, k) | список длиной k из последовательности population | |
| **random.shuffle**(sequence, [rand]) | перемешивает последовательность | **numpy.random.shuffle**(*x*) |
| **random.random**() | случайное число от 0 до 1 | **numpy.random.sample**(*size=None*) |

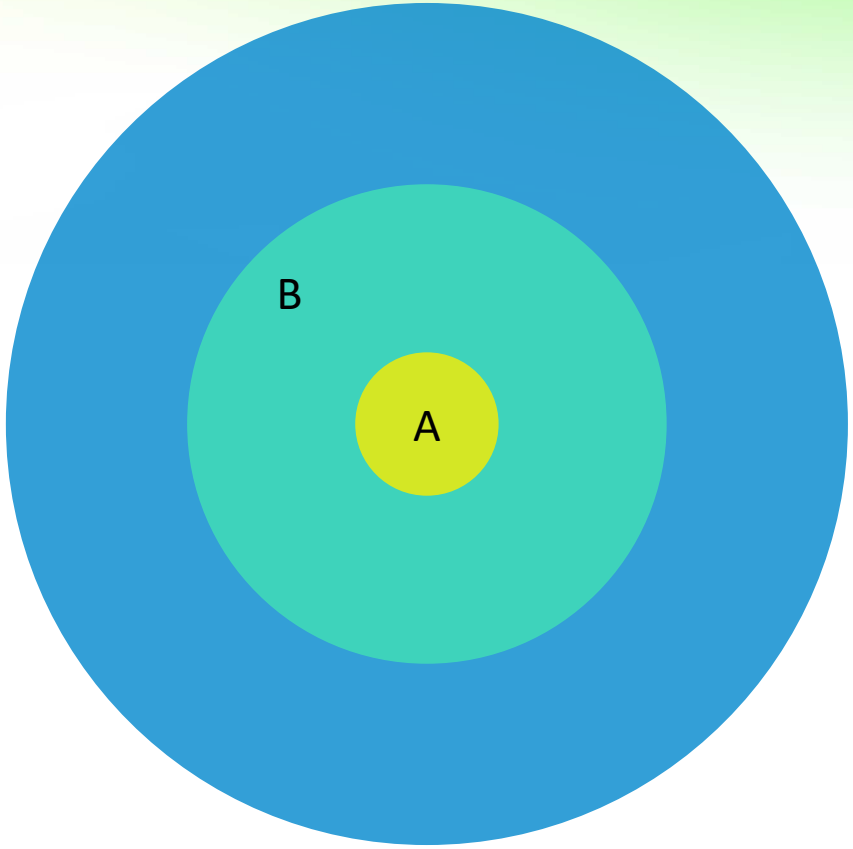
СБЕР

# Практика? Практика!

# Вложенность событий

$A \subseteq B$ – событие A вложено в событие B.

$P(A) \leq P(B)$



# Сумма и произведение событий

AB – произведение.

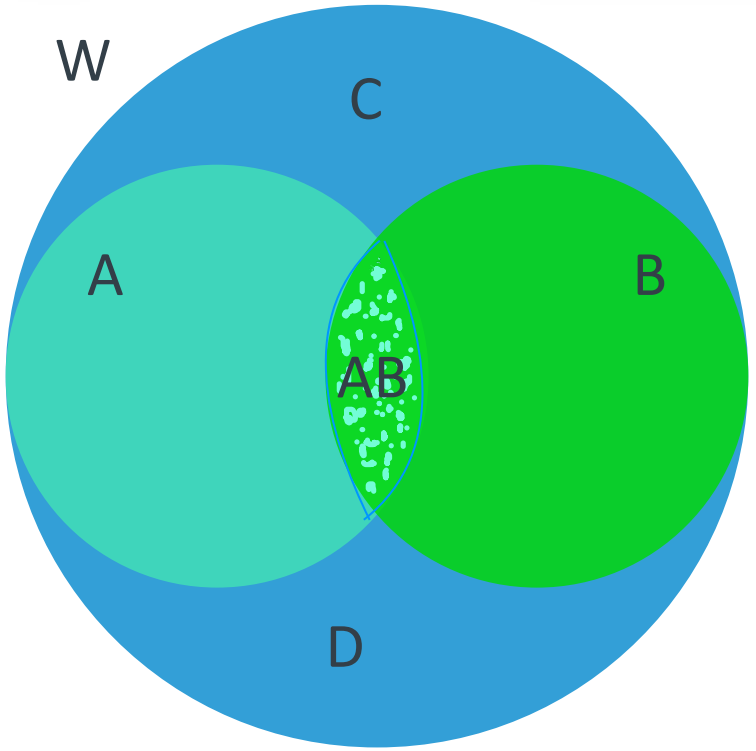
P(AB) = P(A) * P(B)

A + B – сумма.

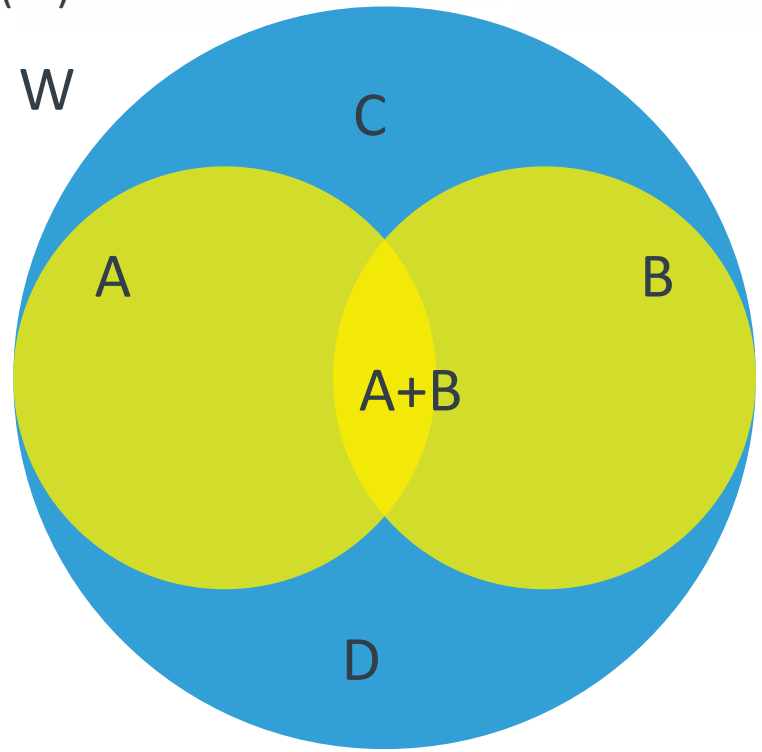
P(A+B) = P(A) + P(B) - P(AB)

| | |
|---|---|
| A | B |
| C | D |

P(AB) = S(AB)/S(W)




P(A)+P(B)+P(C)+P(D) – P(AB) = 1



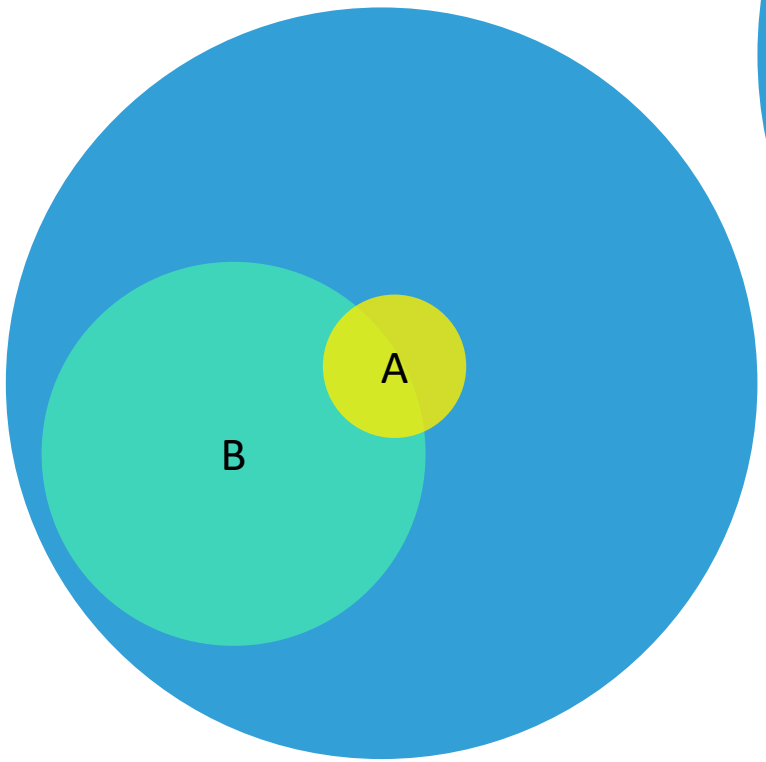

P(A) = S(A) / (S(A)+S(B)+S(C)+S(D)-S(AB))

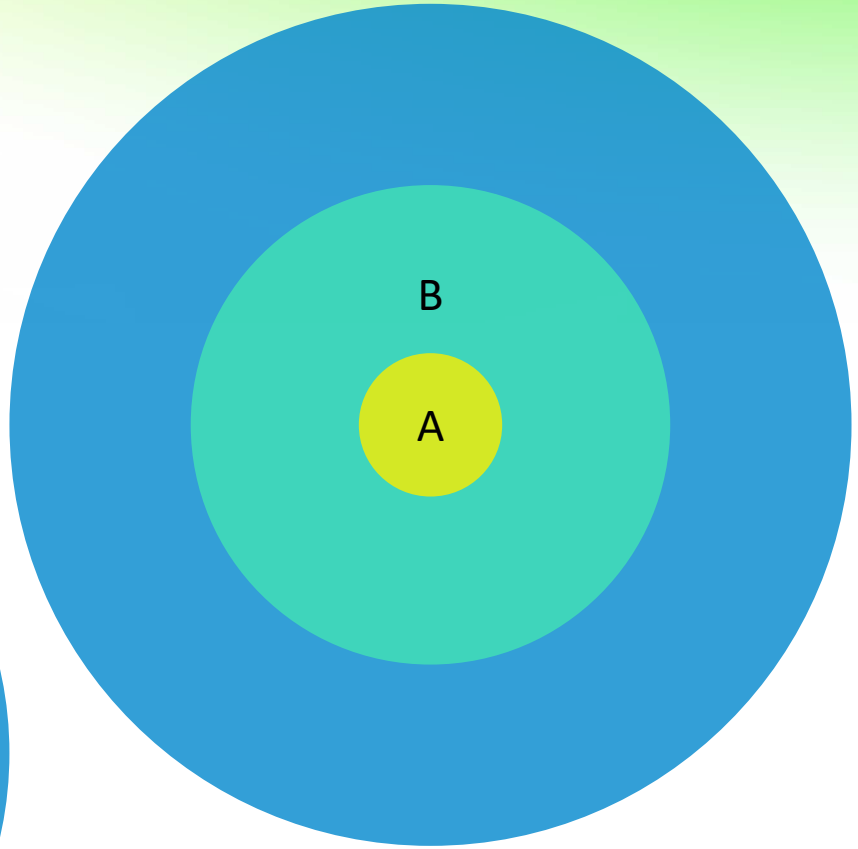
# Дополнение

B \ A – происходит событие B,
но не происходит событие A

P(B \ A) = P(B) - P(AB)

Если A полностью содержится
в B, то P(B \ A) = P(B) - P(A)



# Независимость

Если P(AB) = P(A)P(B)

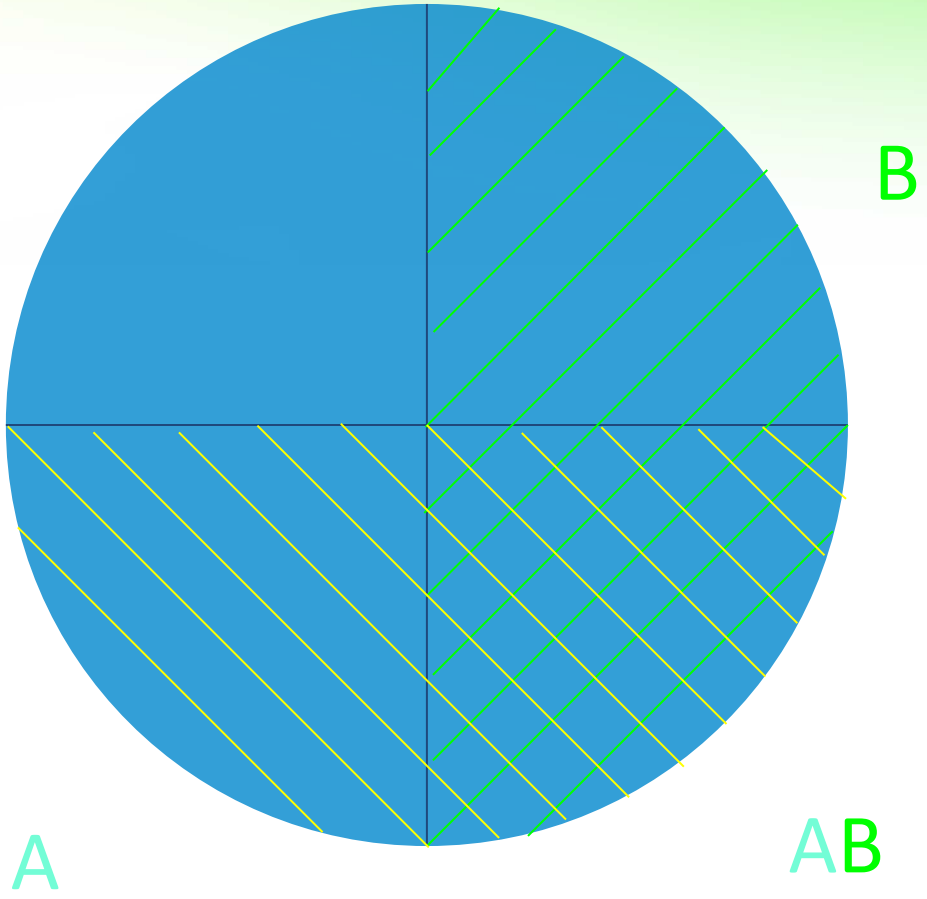
P(A) = 0.5

P(B) = 0.5

P(AB) = 0.25 = P(A)P(B) = 0.5 * 0.5 = 0.25 → P(AB) = P(A)P(B)
события независимы

P(A|B) = P(AB)/P(B) = P(A)*P(B)/P(B) = P(A)

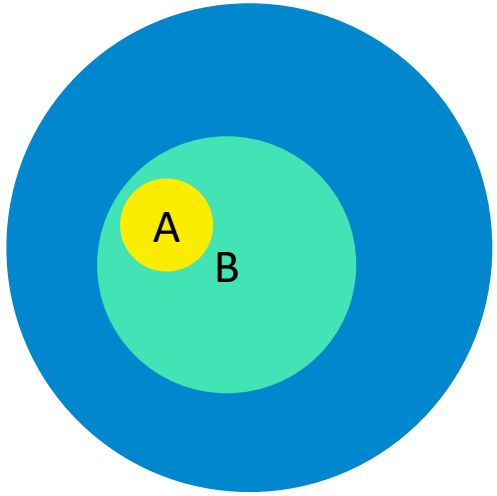
P(A|B) = P(AB)/P(B) = 0.25 / 0.5 = 0.5



# Условная вероятность

Пусть событие A в примере с мишенью — это попадание в «десятку», событие B — попадание в любое место мишени. Если известно, что событие B произошло, то вероятность события A повышается.

$$P(A|B) = \frac{P(AB)}{P(B)} \quad P(B) > 0$$




Пусть: P(A) = 0.05, P(B) = 0.4, P(AB) = P(A) = 0.05

Тогда: $P(A|B) = \frac{P(A)}{P(B)} = \frac{0.05}{0.4} = 0.125^{*}$

*По формуле ПВ: $P(A) = P(A|B)P(B) + P(A|\overline{B})P(\overline{B})$

# Формула полной вероятности

Если события $B_1, B_2, \ldots, B_n$ попарно несовместны ($B_iB_j = \emptyset$ — невозможное событие при любых $i \neq j$ их сумма является достоверным событием ($B_1+B_2+\cdots+B_n = \Omega$), и A есть некое интересующее нас событие, то

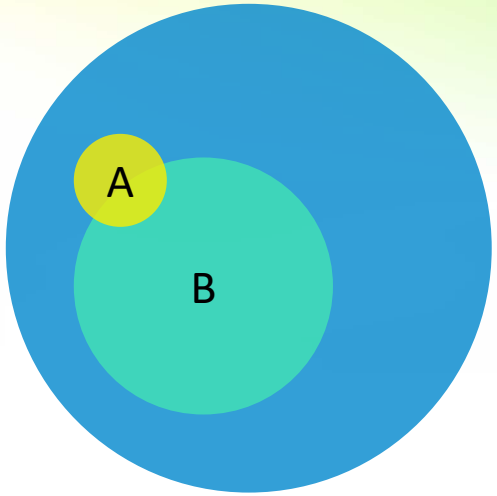
$P(A) = \sum_{k=1}^{n} P(B_k)P(A|B_k)$ - формула полной вероятности

Приведенные условия означают - из событий $B_1, B_2, \ldots, B_n$ может наступить ровно одно и какое-нибудь из них обязательно наступит.

Формула полной вероятности – модель, которая определяет вероятность если в каждом из n случаев (гипотез) известно, как вычислить вероятность любого события *A* (известны условные $P(A|B_k)$ ), то зная вероятности случаев (гипотез) $B_k$, можно вычислить вероятность события *A*. Т.е. условные вероятности при различных гипотезах усредняются с весами, равными вероятностям этих гипотез.

# Формула Байеса

Условные вероятности двух событий A и B связаны между собой формулой Байеса




$$P(A|B) = \frac{P(A)P(B|A)*}{P(B)}$$

Пусть: P(A) = 0.05, P(B) = 0.4, P(B|A) = 0.5

Тогда: $P(A|B) = \frac{0.05*0.025}{0.4} = 0.0625$

Проверка того, может ли, полученный на выборке, быть подтвержден на другой выборке
Можно использовать  результаты пошлых исследований
Если порог ошибок не превышен — результат теста все равно правильный

Часто используется, если есть начальные знания о данных. Но сами знания м.б. не объективны (в этом случае теорема перестает работать)
Трудно сравнивать с другими методами, например частотным анализом

$$^{*}P(A|B) = \frac{P(A)P(B|A)}{P(A)P(B|A)+P(\bar{A})P(B|\bar{A})}$$

# Парадокс формулы Байеса

Имеется заболевание с вероятностью распространения 0,001 и метод диагностического обследования, с P(Б) = 0,9 выявляет больного,  P(«Б»|З) = 0,01 ложноположительный результат. «Б» — событие, что обследование показало, что человек болен.
**Найти вероятность того, что человек здоров, если он был признан больным при обследовании**.
P(«Б»| Б) = 0,9;P(«Б» | З)= 0,01;P(Б) = 0,001, значит P(З) = 0,999.Вероятность того, что человек здоров, если он был признан больным равна условной вероятности:
P(З | «Б»).Чтобы её найти, вычислим сначала полную вероятность признания больным:
P(«Б») = 0,999 × 0,01 + 0,001 × 0,9 = 0,01089.
Вероятность, что человек здоров при результате «болен»:
P(З | «Б») = 0,999 × 0,01 / (0,999 × 0,01 + 0,001 × 0,9) ≈ 0,917.
**Таким образом, 91,7 % людей, у которых обследование показало результат «болен», на самом деле здоровые люди.** Причина этого в том, что по условию задачи вероятность ложноположительного результата хоть и мала, но на порядок больше доли больных в обследуемой группе людей.
Если ошибочные результаты обследования можно считать случайными, то повторное обследование того же человека будет давать независимый от первого результат. В этом случае для уменьшения доли ложноположительных результатов имеет смысл провести повторное обследование людей, получивших результат «болен». Вероятность того, что человек здоров после получения повторного результата «болен», также можно вычислить по формуле Байеса: P(З | «Б», «Б») = 0,999 × 0,01 × 0,01 / (0,999 × 0,01 × 0,01 + 0,001 × 0,9 × 0,9) ≈ 0,1098.

# Парадокс формулы Байеса

А что на практике?

P(Б) = 0,001,
P(З) = 0,999

P(«Б»| Б) = 0,9
P(«Б» | З)= 0,01

P(«Б»| Б) = 0,9
P(«Б» | З)= 0,01

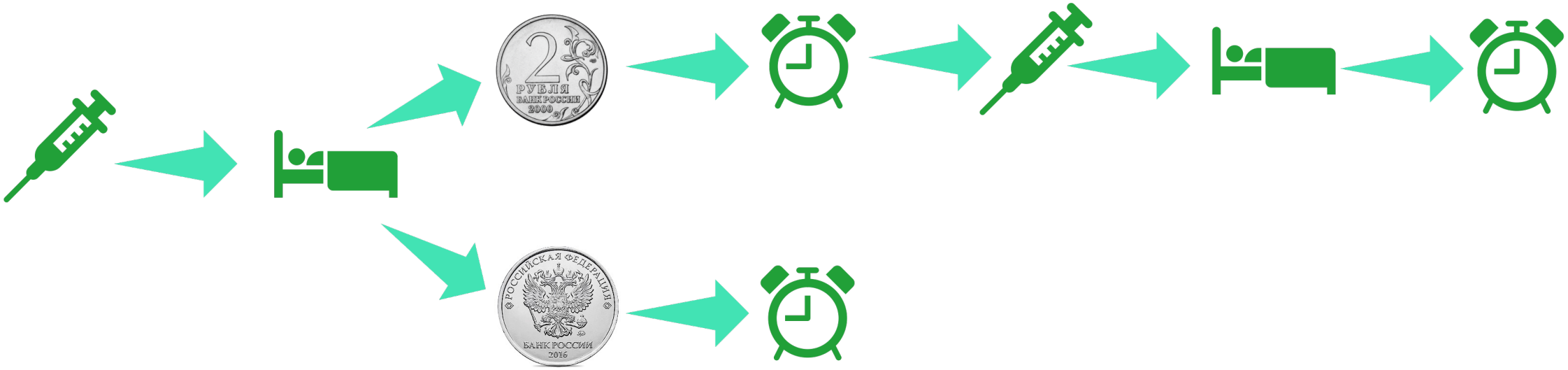
# Объективность наблюдений

**Парадокс спящей красавицы** — Задачу на расчет вероятности, которая имеет два различных решения, противоречащих друг другу.

Испытуемой («Спящей красавице») делается укол снотворного. Бросается симметричная монета. В случае выпадения орла её будят, и эксперимент на этом заканчивается. В случае выпадения решки её будят, делают второй укол (после чего она забывает о побудке) и будят на следующий день, не бросая монеты (в таком случае эксперимент идёт два дня подряд). Вся эта процедура Красавице известна, однако у неё нет информации, в какой день её разбудили.
Представьте себя на месте Спящей красавицы. Вас разбудили. Какова вероятность того, что монета упала орлом?
**Решение 1:** У вас нет никакой информации о результате выпадения монеты и предыдущих побудках. Поскольку известно, что монета честная, можно предположить, что вероятность выпадения орла равна 1/2.
**Решение 2:** Проведём эксперимент 1000 раз. Спящую красавицу будят в среднем 500 раз с орлом и 1000 раз с решкой (т. к. в случае решки спящую красавицу будят 2 раза). Поэтому вероятность выпадения орла равна 1/3.
При этом до начала испытания (до броска монеты) Спящая красавица оценивает эту вероятность как 1/2, но одновременно знает, что после пробуждения она будет оценивать вероятность как 1/3. В этом и состоит парадокс.

# Объективность наблюдений

А что на практике?



# Закон распределения случайной величины

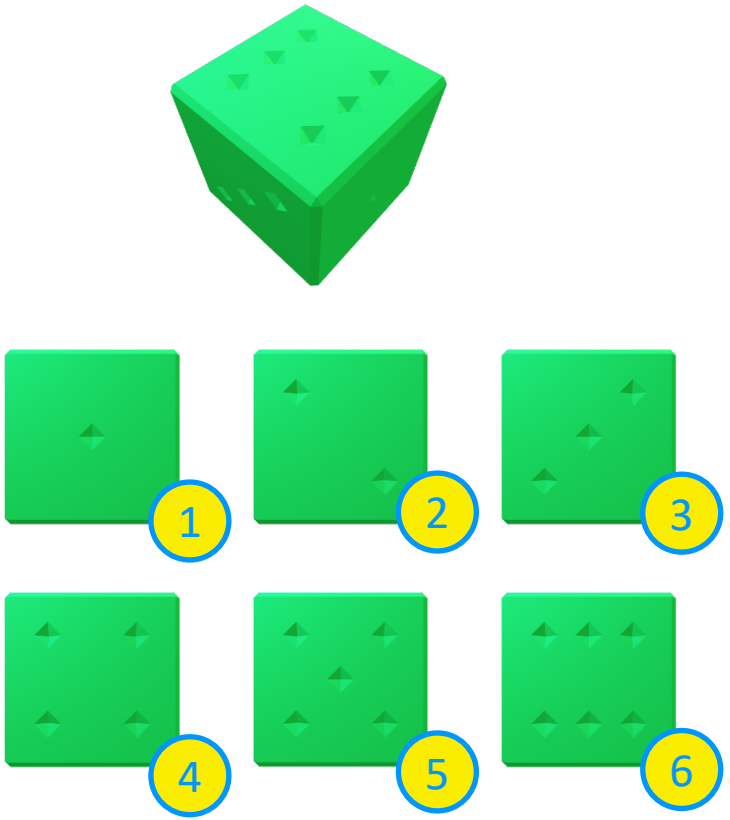
Есть случайная величина $X = X(\omega)$. Среди $S$ чисел $X(\omega)$, $\omega \in \Omega$, могут быть одинаковые. Обозначим $x_1, x_2, \ldots, x_n$ все различные значения функции $X(\omega) \Rightarrow n \leq |\Omega|$.
Можно найти вероятность события $\{X = x_k\}$, $k=1, \ldots, n$. Событие $\{X=x_k\}$ *состоит из всех тех* $\omega$ из $\Omega = \{\omega_1, \omega_2, \ldots \omega_s\}$, для которых $X(\omega) = x_k$ :

$$\{X = x_k\} = \{\omega: X(\omega) = x_k\}$$

*Набор вероятностей это закон распределения случайной величины:*

$$P\{X = x_k\} = \frac{|\{X = x_k\}|}{|\Omega|}, \; k=1, \ldots, n$$

# Дискретные случайные величины




$$\{1, 2, 3, 4, 5, 6\} \Longrightarrow \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{pmatrix}$$

$p_i > 0, \quad i \in \{1,2,3,4,5,6\}$

$$\sum_{i=1}^{6} p_i = 1$$

# Дискретные случайные величины

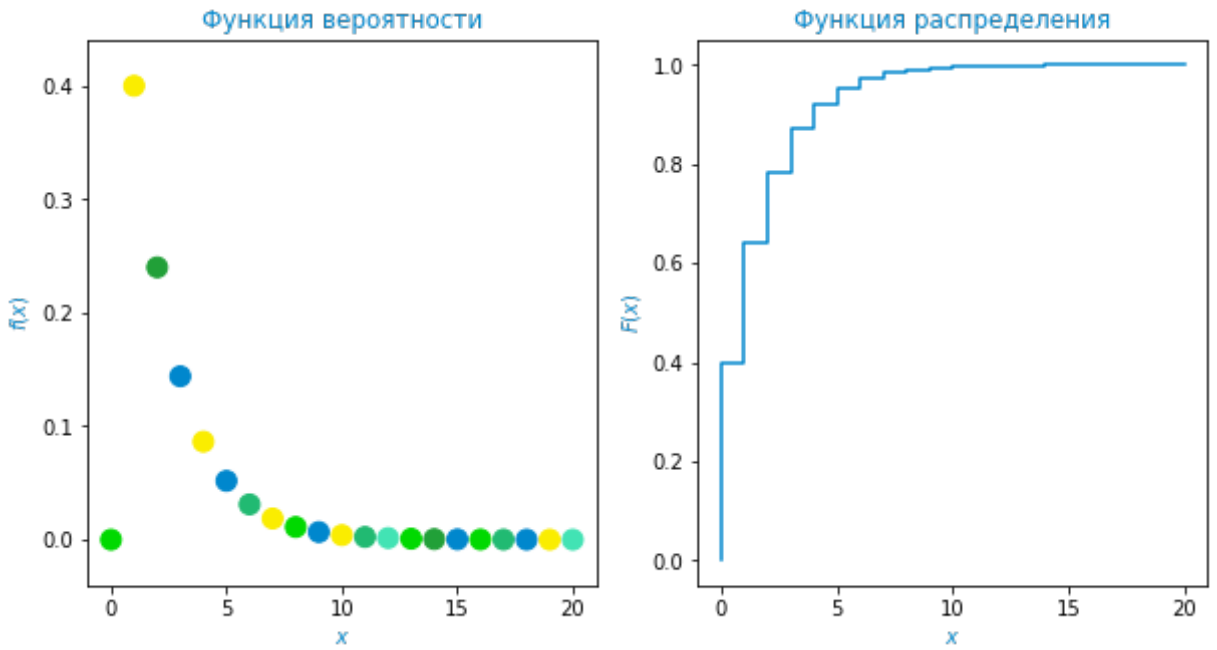
**X** принимает счётное множество* значений $\Omega = \{\omega_1, \omega_2, \omega_3, \ldots\}$ с вероятностями $p_1, p_2, p_3, \ldots$, где

$$p_i \geq 0 \quad \forall i$$

$$\sum_{i=1}^{\infty} p_i = 1$$

**P(X = $\omega_i$) = $p_i$** – функция вероятности

**F(x) = P (X ≤ x)** – функция распределения




* множество, элементы которого можно перенумеровать

# Распределение Бернулли

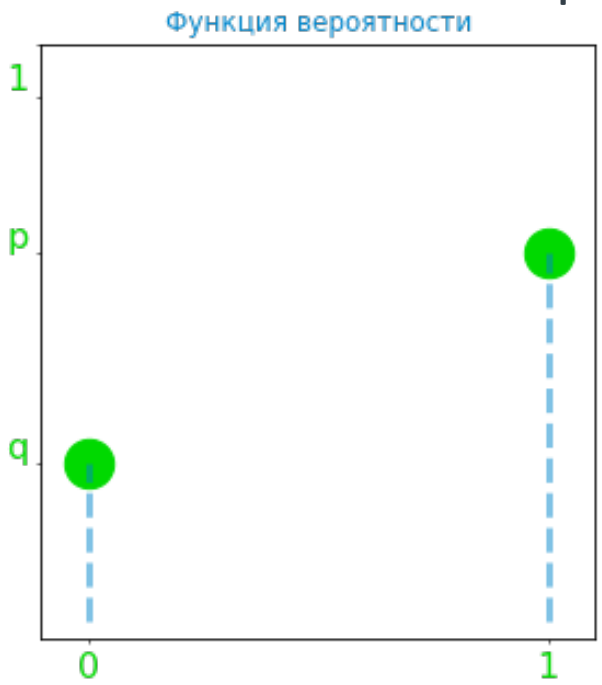
p = 0.7

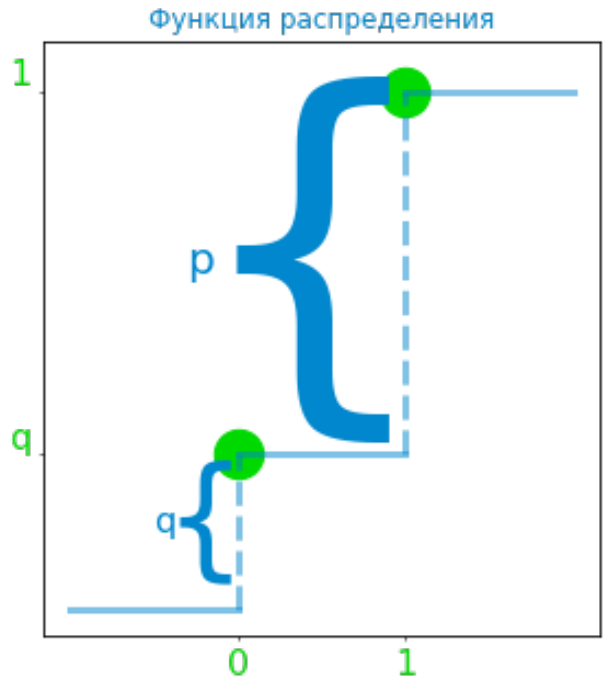
q = 0.3

0 1

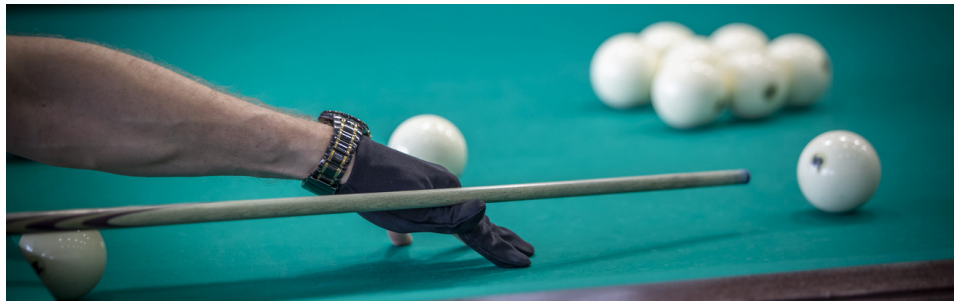
**P(X = 1) = p,**
**P(X = 0) = 1 - p = q**

**X ~ Ber(p)**

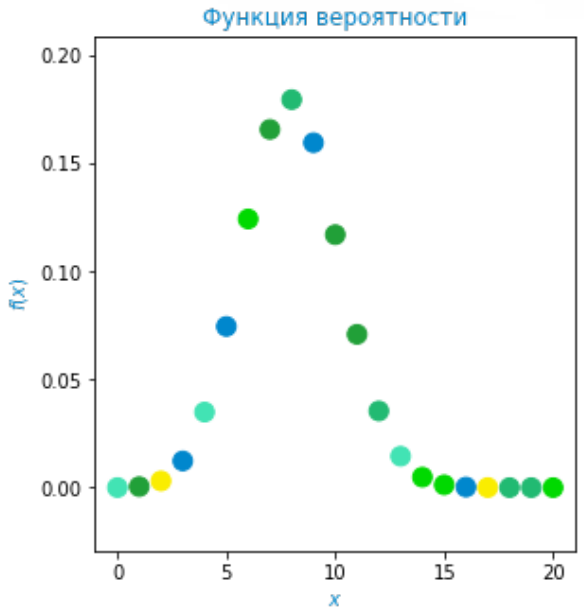


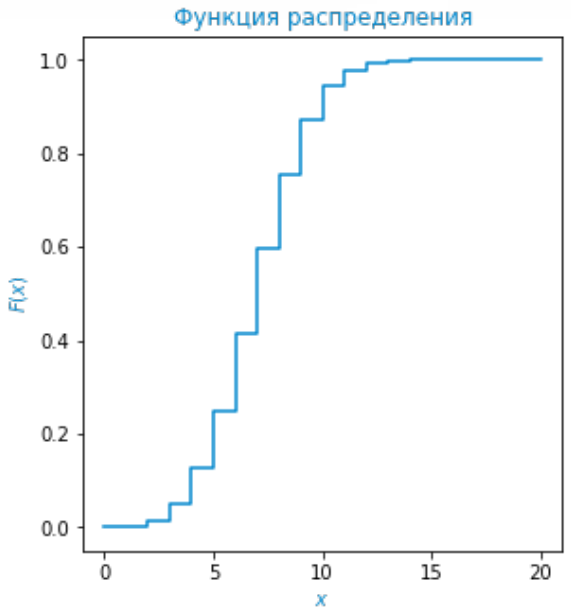



# Биномиальное распределение



p = 0.4




**p** – вероятность попадания

**n** – число попыток

**X** – число попаданий

**$P(X = n) = p^n$** - вероятность попасть n-раз

**$P(X = k) = C_n^k p^k (1 - p)^{n-k}$** * - вероятность попасть k-раз из n

**$X \sim Binom(n, p)$**

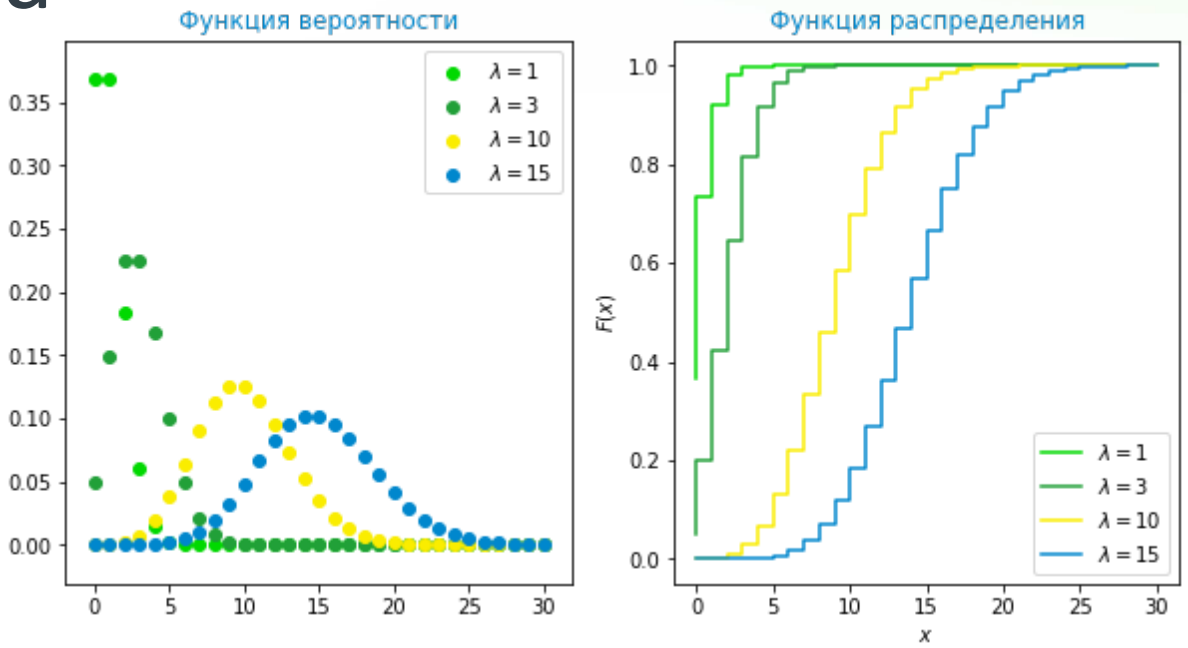
* Биноминальный коэффициент $C_n^k = \frac{n!}{k!(n-k)!}$

# Распределение Пуассона

**X** – число попадании

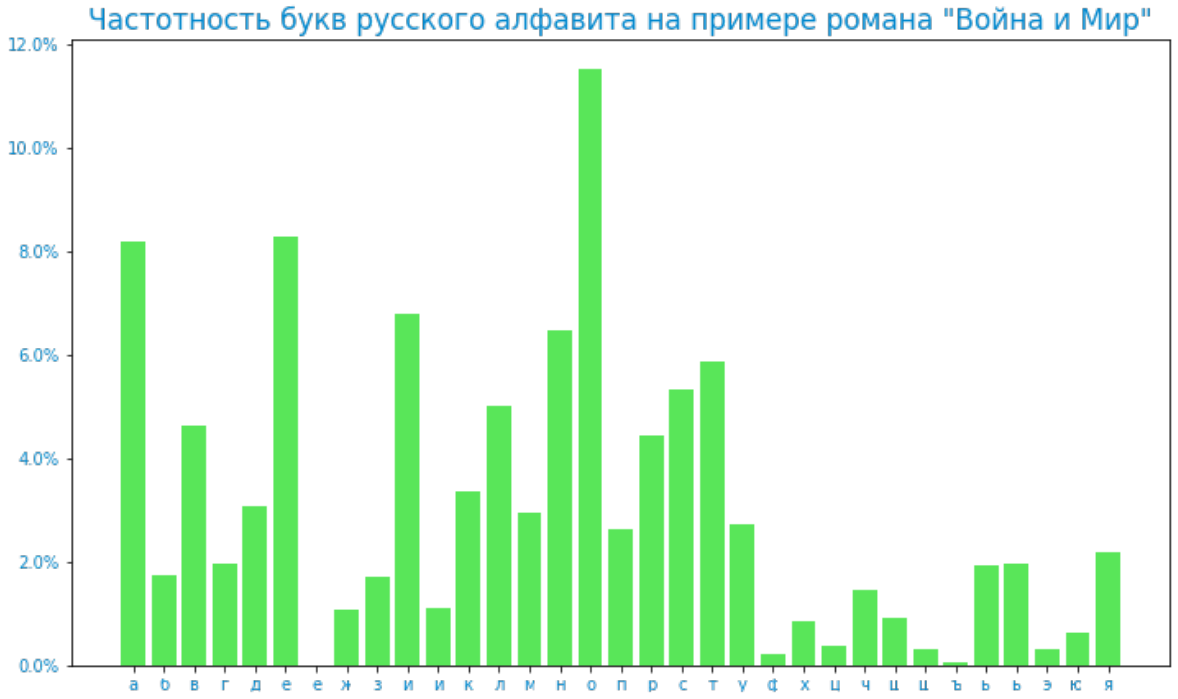
$$\mathbf{P(X = k)} = \frac{\boldsymbol{\lambda^{k} e^{-\lambda}}}{\mathbf{k!}}, \quad \lambda > 0, \quad k = 0, 1, 2, \ldots$$

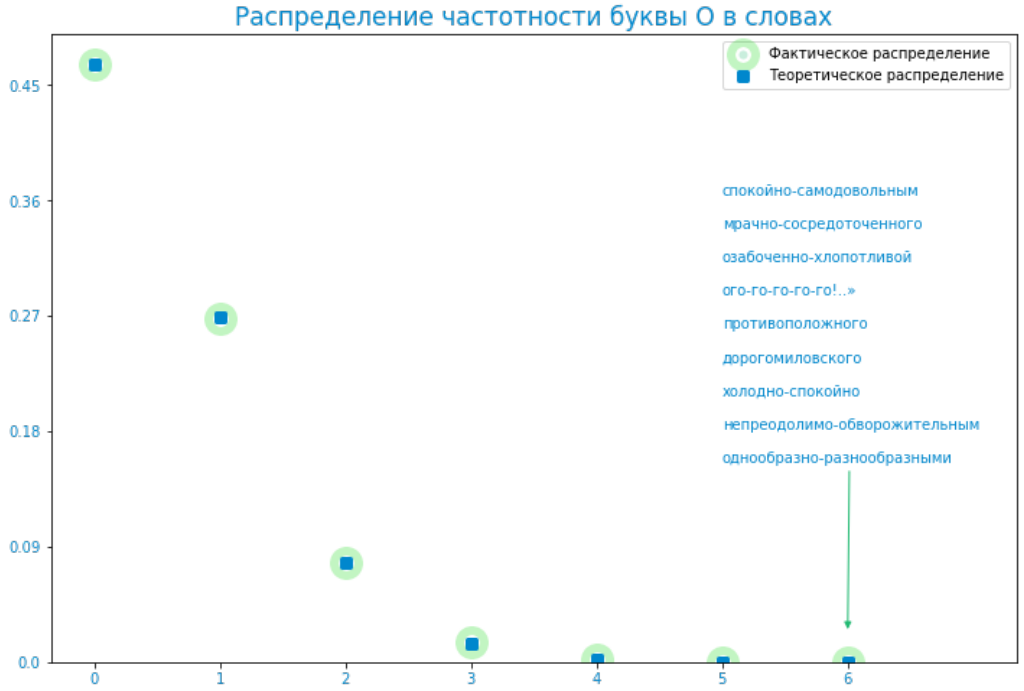
$X \sim \text{Pois}(\lambda)$ *




* $\lambda$ среднее количество событий за фиксированный промежуток времени

# Распределение Пуассона



# Реализация в Python

| Распределение | Scipy | Numpy |
|---|---|---|
| Бернулли | scipy.stats.bernoulli(p) | numpy.random.binomial (*1*, *p*, *size=None*) |
| Биномиальное | scipy.stats.binom(n, p) | numpy. random.binomial (*n*, *p*, *size=None*) |
| Пуассона | scipy.stats.poisson(lam) | numpy. random.poisson(*lam=1.0*, *size=None)* |

СБЕР

# Практика? Практика!

# Непрерывные случайные величины

$$|\Omega| = \infty \Rightarrow \quad P(X = \omega) = 0 \qquad \forall \omega \in \Omega$$

Множество значений $\Omega$ несчётное,
вероятность события $P(X = \omega)$ нулевая

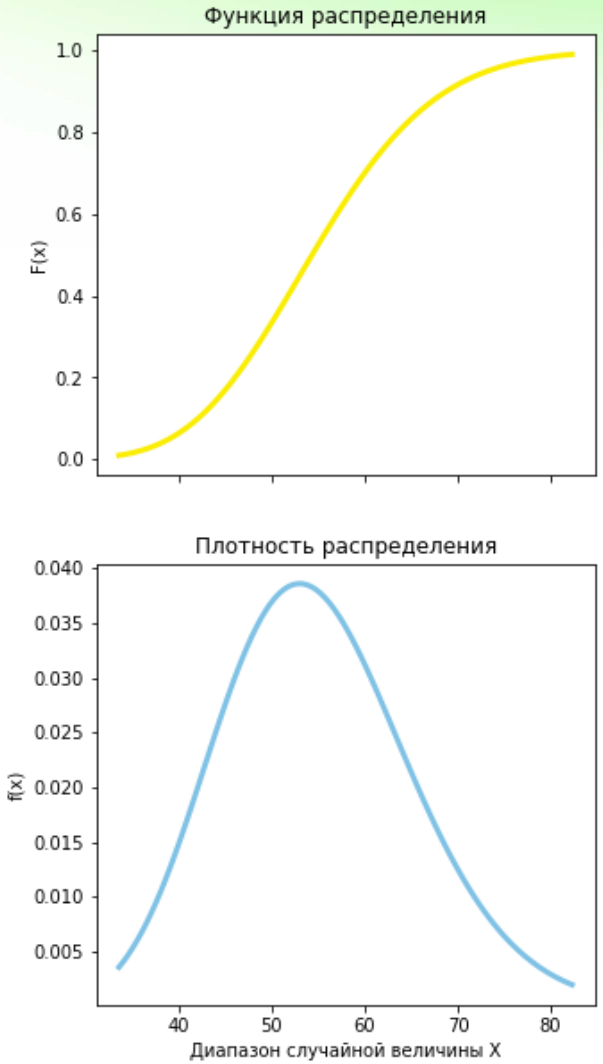
Способы определения:

**F(x) = P (X ≤ x)** – функция распределения

**f(x):** $\int_a^b f(x)dx = P(a \le X \le b)$ - плотность распределения

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

$$\int_{-\infty}^{+\infty} f(t)dt = P(-\infty \le X \le +\infty) = 1$$

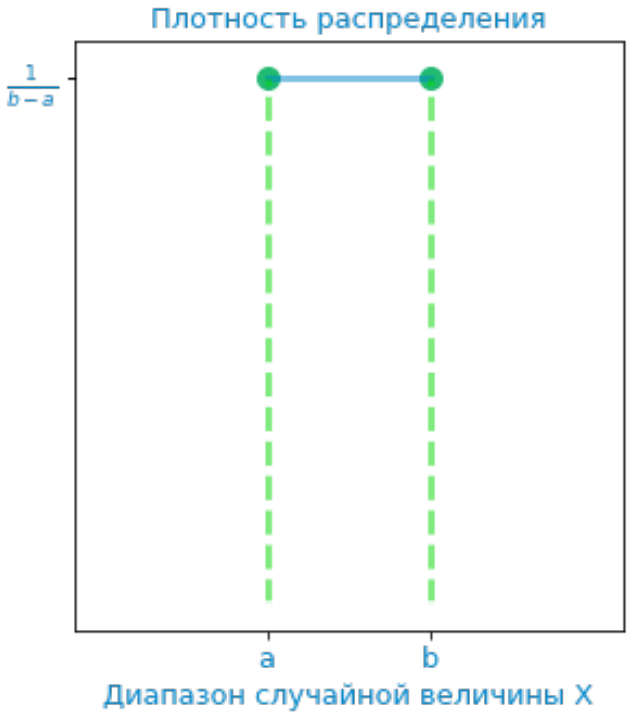


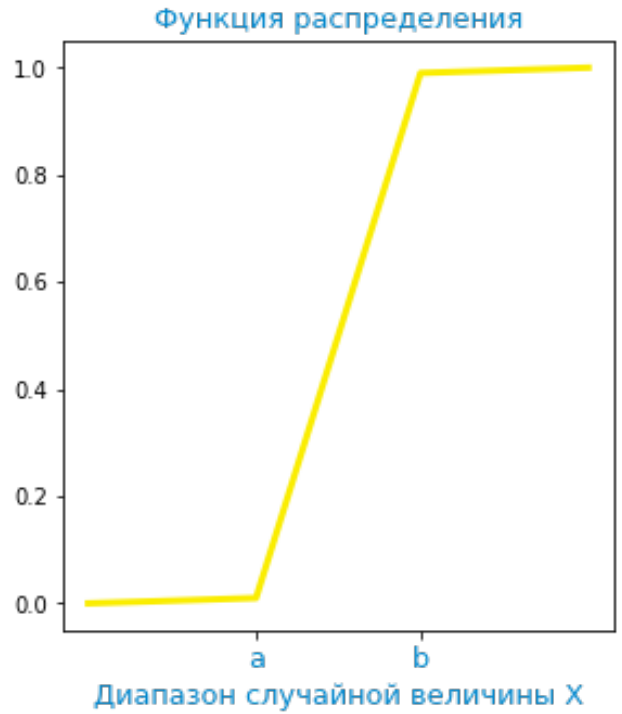

* множество, элементы которого не могут быть перенумерованы

# Равномерное распределение

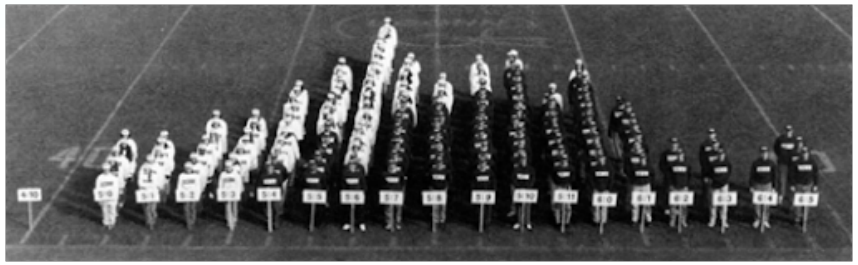

$$X \sim U\,(a, b)$$

$$f(x) = \begin{cases} \dfrac{1}{b\ -a} & x\ \in [a, b] \\ \\ 0 & x \notin [a, b] \end{cases}$$



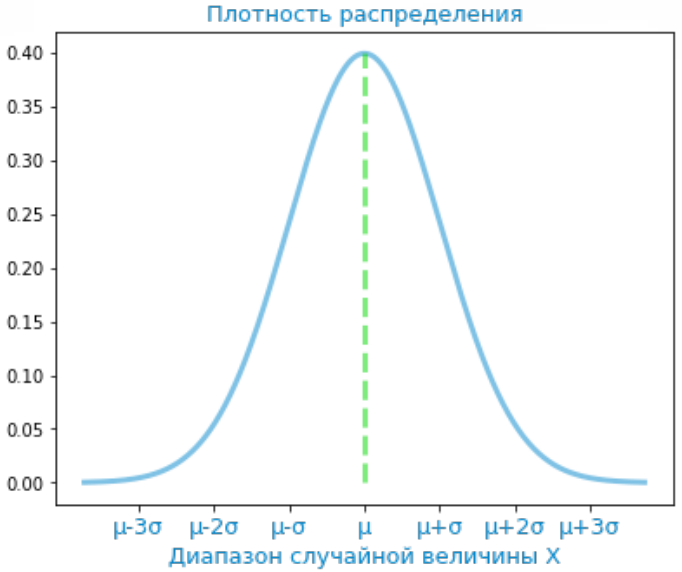



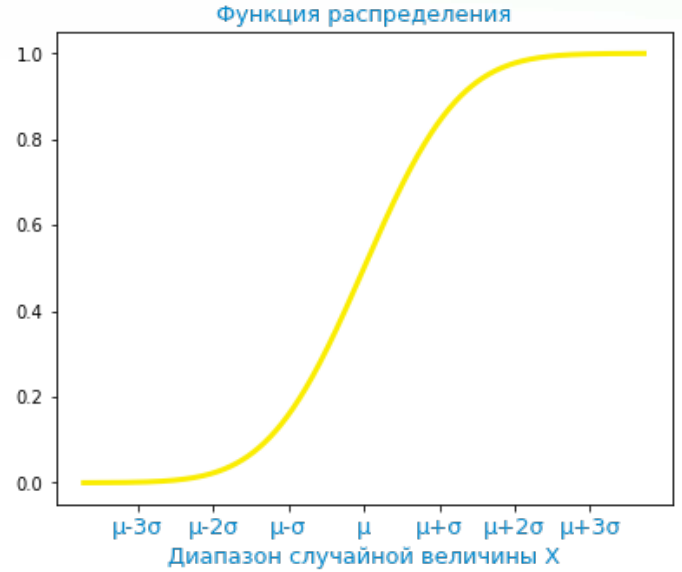
# Нормальное распределение

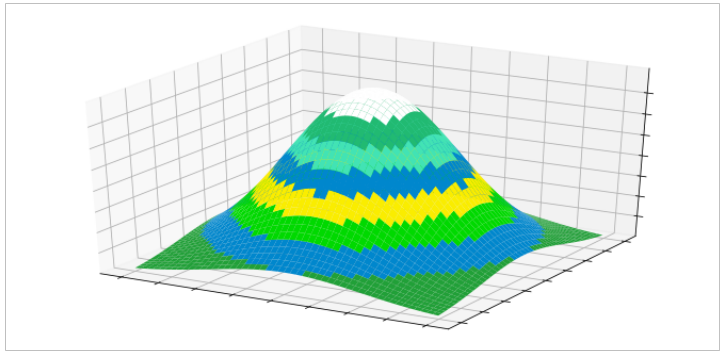


$$X \sim N\left(\mu, \sigma^2\right)$$

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



# Реализация в Python

| Распределение | Random | Scipy | Numpy |
|---|---|---|---|
| **Равномерное** | **random.uniform**(A, B) | **scipy.stats.uniform (loc, scale-loc)** | **numpy.random.uniform (*low=0.0, high=1.0, size=None*)** |
| **Нормальное** | **random.gauss($\mu, \sigma$)** **random.normalvariate($\mu, \sigma$)** | **scipy.stats.norm(loc, scale)** | **numpy. random.normal (*loc=0.0, scale=1.0, size=None*)** |

СБЕР

# Практика? Практика!

# Резюме

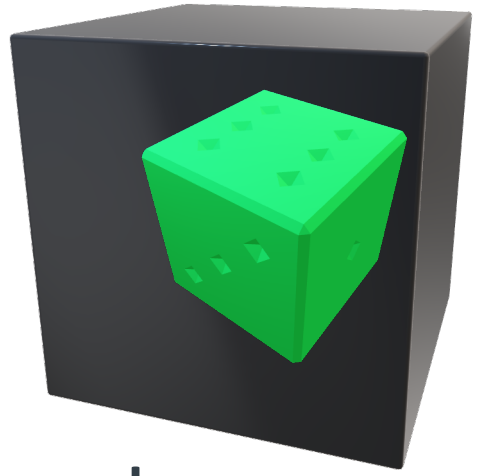
- Познакомились в библиотекой Pandas
- Поговорили про теорию вероятностей и статистику
  - Рассмотрели основные понятия теории вероятностей
  - Порешали задачи
- Посмотрели что может предложить Python для работы со случайными величинами

# Пояснения к заданию

- Решения задач «классическим» способом можно найти в интернете
- Ознакомление с этими решениями приветствуется

*А что на практике?*

*Результаты исследований и испытаний – это не события*
*В результатах испытаний бывают ошибки*
*С помощью испытаний мы получаем вероятности определенного исхода*



- Вспоминаем принцип «Черного ящика»
- Моделируем условие задач на **Python**:
  Используем возможности библиотек **random, SciPy, NumPy, pandas** ….
  Производим расчеты
- Сверяем полученные результаты с решениями из интернета ☺

# Полезные ссылки

Функции для работы с распределениями в SciPy:
http://scipy.github.io/devdocs/reference/stats.html#module-scipy.stats
Функции для работы со случайными величинами в random:
https://docs.python.org/3/library/random.html
Функции для работы со случайными величинами в NumPy:
https://numpy.org/doc/stable/reference/random/index.html
Функции для работы с распределениями в NumPy:
https://numpy.org/doc/stable/reference/routines.statistics.html
Функции для работы с распределениями в Pandas:
https://pandas.pydata.org/docs/user_guide/basics.html

# Обратная связь

?

СБЕР

# Спасибо за внимание!