лабораторная 1

1. (2) В городе живут $n+1$ людей. Человек, условный "прародитель" пишет два письма случайно выбранным адресатам, которые образуют "первое" поколение. Те делают то же самое, в результате чего образуется "второе поколение". Вообще каждый из людей, входящих в "поколение" с номером r, посылает два письмо случайно выбранным адресатам. Найти вероятности того, что прародитель не входит ни в одно из поколений с номерами $1, ..., r$.
   пусть человек, получивший письмо, не может отправить письмо самому себе, и обязан отправить его двум разным людям. при этом если человек одновременно получил два письма, то заставим его писать 4, и так далее. но на каждую пару человек ограничение в обязаны быть разными накладывать не будем – получивший два и более письма должен будет с каждым из них разбираться в отдельности. таким образом, в i поколение входит всегда 2^i получателей, так как данная модель позволяет нам считать человека, получившего k писем одновременно, за k получаетелей, и, соответственно, k отправителей следующему поколению. тогда у любого получателя есть $\binom{n}{2}$ способов выбрать, кому написать, и вероятность того, что человек, **не** являющийся сам прародителем, **не** отправит письмо прародителю, равна $\binom{n-1}{2}/\binom{n}{2} = (n-2)/n = 1 - 2/n$ (вероятность того, что прародитель не отправит письмо самому себе, соответственно, $= 1$).
   тогда вероятность того, что прародитель не входит в поколения $1, ..., r$, равна вероятности того, что никто из поколений $0, ..., r-1$ не отправит письмо прародителю, $= 1 \cdot \prod_{i=1}^{r-1}(1-2/n)^{2^i} = (1-2/n)^{2^r-2}$

2. (1) Пусть n писем случайно раскладываются по n конвертам. Найти вероятность, что в точности m писем попадут в свои конверты. Как данную вероятность можно приближенно посчитать при фиксированном m и достаточно большом n?
   зафиксируем, что $0 \leq m \leq n$.
   пусть M – количество писем, попавших в свои конверты, $\mathbb{P}(M = m)$ – вероятность того, что в точности m писем оказались в своих конвертах.
   заметим, что $\mathbb{P}(M = m) =$ количеству способов выбрать m писем из данных n $\cdot$ вероятность того, что выбранные m писем оказались в своих конвертах (обозначим это событие A_m) $\cdot$ вероятность того, что оставшиеся $n-m$ писем оказались не в своих конвертах (обозначим B_{n-m}).
   тогда $\mathbb{P}(M = m) = \binom{n}{m} \cdot \mathbb{P}(A_m) \cdot \mathbb{P}(B_{n-m})$.
   немного переформулируем задачу для $\mathbb{P}(A_m)$: рассмотрим числа от 1 до n, у нас m фиксированных чисел $k_1 \dots k_m$, и нужно найти вероятность того, что в случайной перестановке чисел $(1 \dots n)$ фиксированные числа окажутся на своих местах. всего перестановок, как мы знаем, $n!$. чтобы посчитать перестановки, в которых все k_i стоят на своих местах, нам нужно сначала разложить эти числа по своим местам, а после распределить остальные числа по оставшимся $n-m$ местам, то есть таких перестановок получится $(n-m)!$, а значит, вероятность того, что фиксированные числа окажутся на своих местах в случайной перестановке $\mathbb{P}(A_m) = (n-m)!/n!$.
   теперь рассмотрим событие B_{n-m} – вероятность того, что раскладывая $n-m$ писем по $n-m$ конвертам ни одно из писем не оказалось в своем конверте.
   $\mathbb{P}(B_n) = 1 - \mathbb{P}(\text{хотя бы одно письмо оказалось в своем конверте}) = 1 - \mathbb{P}(\bigcup_{i=1}^n C_i)$, где событие C_i означает, что i письмо оказалось в своем конверте в случайном распределении писем.
   чтобы посчитать $\mathbb{P}(\bigcup_{i=1}^n C_i)$, воспользуемся формулой включений-исключений:

$$\mathbb{P}(\bigcup_{i=1}^n C_i) = \sum_{1 \leq i \leq n} \mathbb{P}(C_i) - \sum_{1 \leq i < j \leq n} \mathbb{P}(C_i \cap C_j) + \sum_{1 \leq i < j < k \leq n} \mathbb{P}(C_i \cap C_j \cap C_k) - \cdots + (-1)^{n-1}\mathbb{P}(C_1 \cap C_2 \cap \cdots \cap C_n)$$

   из предыдущих рассуждений мы так же получаем вероятности C_i и их пересечений:

$$\mathbb{P}(\bigcup_{i=1}^n C_i) = \sum_{1 \leq i \leq n} \frac{(n-1)!}{n!} - \sum_{1 \leq i < j \leq n} \frac{(n-2)!}{n!} + \sum_{1 \leq i < j < k \leq n} \frac{(n-3)!}{n!} - \cdots + (-1)^{n-1}\frac{1}{n!}$$

   упрощая данное выражение, получаем:

$$\begin{aligned}\mathbb{P}(\bigcup_{i=1}^n C_i) &= \binom{n}{1} \cdot \frac{(n-1)!}{n!} - \binom{n}{2} \cdot \frac{(n-2)!}{n!} + \binom{n}{3} \cdot \frac{(n-3)!}{n!} - \cdots + (-1)^{n-1}\frac{1}{n!} \\ &= 1 - \frac{1}{2!} + \frac{1}{3!} - \cdots + (-1)^{n-1}\frac{1}{n!}\end{aligned}$$

   итого,

$$\mathbb{P}(B_n) = 1 - 1 + \frac{1}{2!} - \frac{1}{3!} - \cdots + (-1)^n\frac{1}{n!}$$

$$\mathbb{P}(M=m)=\binom{n}{m}\cdot\frac{(n-m)!}{n!}\cdot(1-1+\frac{1}{2!}-\frac{1}{3!}-\dots+(-1)^{n-m}\frac{1}{(n-m)!})$$
$$=\frac{1}{m!}\cdot(1-1+\frac{1}{2!}-\frac{1}{3!}-\dots+(-1)^{n-m}\frac{1}{(n-m)!})$$

при фиксированном m и достаточно большом n $\mathbb{P}(M=m)$ стремится к $1/(m!\cdot e)$.

3. (1) На части параболы $y=x^2$, $x\in(0,2)$ случайно выбирается точка. Как можно интерпретировать "случайность" в данной задаче? С какой вероятностью угол, образованный радиус-вектором выбранной точки с положительным направлением оси абсцисс, не превосходит $\pi/3$?
вероятность того, что случайно выбранная точка удовлетворяет заданному условию (обозначим это событие A) находится по формуле $\mathbb{P}(A)=m(A)/m(B)$, где $m(B)$ и $m(A)$ – меры пространств всех элементарных исходов и исходов, удовлетворяющих условию.
кривая задана функцией $y=f(x)$. меру пространства всех элементарных исходов вычислим как длину кривой $y=x^2$ на промежутке $x\in(0,2)$. благодаря непрерывности производной $f(x)$, ее можно вычислить по формуле
$$m(B)=\int_0^2\sqrt{1+(f'(x))^2}dx=\int_0^2\sqrt{1+4x^2}dx$$
теперь посмотрим, в каких случаях точка (x_0,y_0) будет удовлетворять заданному условию. чтобы угол между осью Ox и радиус-вектором выбранной точки $(x_0,y_0)\in\{y=x^2,x\in(0,2)\}$ (обозначим его φ) был $\le\pi/3$, нам нужно, чтобы для данной точки отношение y_0/x_0 (равное тангенсу φ) находилось в промежутке $(0,\tan(\pi/3)=\sqrt{3}]$
$$\tan(\varphi)=\frac{y_0}{x_0}=\frac{x_0^2}{x_0}=x_0\in(0,\sqrt{3}]$$
то есть нам подходят все точки прямой на промежутке $x\in(0,\sqrt{3}]$.
таким образом, меру пространства элементарных исходов, удовлетворяющих условию, можно вычислить как
$$m(A)=\int_0^{\sqrt{3}}\sqrt{1+4x^2}dx$$
$$\int\sqrt{1+4x^2}dx=\frac{\ln(|\sqrt{4x^2+1}+2x|)}{4}+\frac{x\sqrt{4x^2+1}}{2}+c$$
$$\mathbb{P}(A)=m(A)/m(G)=\frac{\frac{\ln(\sqrt{13}+2\sqrt{3})}{4}+\frac{\sqrt{3\cdot13}}{2}}{\frac{\ln(\sqrt{17}+4)}{4}+\sqrt{17}}\approx0.777194$$

4. (2) Введем события $A_i=\{X=i\}$, $B_i=\{Y=i\}$, $i\ge0$. Известно, что для любых $i\ge0$ и $j\ge0$ события A_i и B_j – независимы, при этом
$$\mathbb{P}(X=i)=e^{-\lambda}\frac{\lambda^i}{i!},\lambda>0$$
$$\mathbb{P}(Y=j)=e^{-\mu}\frac{\mu^j}{j!},\mu>0$$
Найти $P(X=i|X+Y=j)$.
по формуле условной вероятности получаем,
$$\mathbb{P}(X=i|X+Y=j)=\frac{\mathbb{P}(X=i\cap X+Y=j)}{\mathbb{P}(X+Y=j)}$$
заметим, что
$$\mathbb{P}(X=i\cap X+Y=j)=\mathbb{P}(X=i\cap Y=j-i)=\mathbb{P}(A_i\cap B_{j-i})=\mathbb{P}(A_i)\cdot\mathbb{P}(B_{j-i}),$$
так как события A_i и B_{j-i} независимы
$$=e^{-\lambda-\mu}\frac{\lambda^i\mu^{j-i}}{i!(j-i)!}$$
теперь рассмотрим
$$\mathbb{P}(X+Y=j)=\sum_{i=0}^{j}\mathbb{P}(X=i\cap Y=j-i)=\sum_{i=0}^{j}\mathbb{P}(A_i)\cdot\mathbb{P}(B_{j-i})=\sum_{i=0}^{j}e^{-\lambda-\mu}\frac{\lambda^i\mu^{j-i}}{i!(j-i)!}$$
$$=\frac{e^{-\lambda-\mu}}{j!}\sum_{i=0}^{j}\frac{j!\lambda^i\mu^{j-i}}{i!(j-i)!}=\frac{e^{-\lambda-\mu}}{j!}\sum_{i=0}^{j}\binom{j}{i}\lambda^i\mu^{j-i}=\frac{e^{-\lambda-\mu}}{j!}(\lambda+\mu)^j$$
получаем, что
$$\mathbb{P}(X=i|X+Y=j)=\frac{e^{-\lambda-\mu}\cdot\lambda^i\mu^{j-i}}{i!(j-i)!}\cdot\frac{j!}{e^{-\lambda-\mu}\cdot(\lambda+\mu)^j}=\binom{j}{i}\frac{\lambda^i\mu^{j-i}}{(\lambda+\mu)^j}$$

5. Рассмотрите схемы Бернулли при $n \in \{100, 1000, 10000\}$ и $p \in \{0.001, 0.01, 0.1, 0.25, 0.5\}$ и рассчитайте точные вероятности (где это возможно) $\mathbb{P}(S_n \in [n/2 - \sqrt{npq}, n/2 + \sqrt{npq}]$, $\mathbb{P}(S_n \leq 5)$ и максимальную вероятность вида $\mathbb{P}(S_n = k)$, S_n – количество успехов в n испытаниях, и приближенные с помощью одной из предельных теорем. Сравните точные и приближенные вероятности. Объясните результаты.
для нахождения приближенных вероятностей будем пользоваться следующими формулами (и вольфрамом):

(a) локальная формула Муавра-Лапласа:

$$\mathbb{P}(S_n = k) \approx \frac{1}{\sqrt{2\pi npq}} e^{-(k-np)^2/(2npq)}$$

(b) интегральная формула Лапласа:

$$\mathbb{P}(k_1 \leq S_n \leq k_2) \approx \frac{1}{\sqrt{2\pi}} \int_{x_1}^{x_2} e^{-t^2/2} dt, \; x_i = \frac{k_i - np}{\sqrt{npq}}$$

а для вычислять точные значения будем так:

```
import math
from prettytable import PrettyTable
from decimal import Decimal
from tqdm import tqdm


bernoulli_formula = lambda n, k, p, q: Decimal(math.comb(n, k)) * Decimal((p ** k) * (q ** (n - k)))

def sn_in_range(n, p):
    return sum([bernoulli_formula(n, _, p, (1 - p))
                for _ in range(math.ceil(n/2 - math.sqrt(n*p*(1-p))),
                               math.floor(n/2 + math.sqrt(n*p*(1-p))) + 1)])

def sn_leq_5(n, p):
    return sum([bernoulli_formula(n, _, p, (1 - p)) for _ in range(6)])

def sn_eq_k(n, p):
    return max([bernoulli_formula(n, _, p, (1 - p)) for _ in range(n + 1)])

def test():
    ps = [0.001, 0.01, 0.1, 0.25, 0.5]
    ns = [100, 1000, 10000]
    formulas = [sn_in_range, sn_leq_5, sn_eq_k]
    formulas_names = ['sn in range', 'sn <= 5', 'sn = k']
    table = PrettyTable()
    pss = [str(p) for p in ps]
    pss.insert(0, 'formula')
    pss.insert(0, 'n\p')
    table.field_names = pss
    for n in ns:
        for i in tqdm(range(len(formulas))):
            row = [str(n), formulas_names[i]]
            for p in ps:
                row.append(round(formulas[i](n, p), 5))
            table.add_row(row)
    print(table)
```

вот что получилось на (1) формулах и вольфраме и (2) питоне:

n\p	formula	0.001	0.01	0.1	0.25	0.5
100	sn in range	0.0	0.0	0.0	0.0	0.7263
100	sn <= 5	0.6241	0.8425	0.0474	0.0	0.0
100	sn = k	0.6219	0.3426	0.1306	0.0913	0.0793
1000	sn in range	0.0	0.0	0.0	0.0	0.6728
1000	sn <= 5	0.8414	0.0553	0.0	0.0	0.0
1000	sn = k	0.3415	0.1247	0.042	0.0291	0.0252
10000	sn in range	0.0	0.0	0.0	0.0	0.6875
10000	sn <= 5	0.0561	0.0	0.0	0.0	0.0
10000	sn = k	0.1241	0.04	0.0133	0.0092	0.008

n\p	formula	0.001	0.01	0.1	0.25	0.5
100	sn in range	0.00000	0.00000	0.00000	0.00000	0.72875
100	sn <= 5	1.00000	0.99947	0.05758	0.00000	0.00000
100	sn = k	0.90479	0.36973	0.13187	0.09180	0.07959
1000	sn in range	0.00000	0.00000	0.00000	0.00000	0.67306
1000	sn <= 5	0.99941	0.06614	0.00000	0.00000	0.00000
1000	sn = k	0.36806	0.12574	0.04202	0.02912	0.02523
10000	sn in range	0.00000	0.00000	0.00000	0.00000	0.00000
10000	sn <= 5	0.06699	0.00000	0.00000	0.00000	0.00000
10000	sn = k	0.12517	0.04006	0.00000	0.00000	0.00000

вывод: с увеличением n, точность вычислений на формулах, как и время работы программы, заметно повышается, так что при достаточно больших значениях посчитать по формулке кажется хорошей идеей в связи с относительно медлительным точным вычислением.

лабораторная 2

1. (3) Случайная величина X принимает значения -1, 0, 1 с вероятностями p_1, p_2 и p_3 соответственно. Какие условия нужно наложить на p_1, p_2, p_3, чтобы случайная величина X была представима в виде суммы двух независимых одинаково распределенных случайных величин?
пусть Y и Z - две независимые одинаково распределенные случайные величины, которые в сумме дают X. чтобы X принимала только данных 3 значения, нужно, чтобы Y и Z принимали по два: $\frac{1}{2}$ и $-\frac{1}{2}$ с вероятностями p и $1-p$ соответственно. почему: очевидно, что Y и Z должны принимать 2 значения, так как иначе у их суммы будет меньше или больше. предположим, что они принимают значения a и b (положим $a \leq b$, тогда система ниже задается однозначно):
$$\begin{cases} 2a = -1 \\ a + b = 0 \\ 2b = 1 \end{cases}$$
решая систему получаем $a = -\frac{1}{2}$ и $b = \frac{1}{2}$. можем задать следующие соотношения для вероятностей X:
$$\begin{aligned} &\mathbb{P}(Y+Z=-1) = \mathbb{P}(Y=-\frac{1}{2} \cap Z=-\frac{1}{2}) = (1-p)^2 = p_1 \\ &\mathbb{P}(Y+Z=0) = \mathbb{P}(Y=-\frac{1}{2} \cap Z=\frac{1}{2}) + \mathbb{P}(Y=\frac{1}{2} \cap Z=-\frac{1}{2}) = 2p(1-p) = p_2 \\ &\mathbb{P}(Y+Z=1) = \mathbb{P}(Y=\frac{1}{2} \cap Z=\frac{1}{2}) = p^2 = p_3 \end{aligned}$$
тогда для того, чтобы случайная величина X была представима в виде суммы двух независимых одинаково распределенных случайных величин, необходимо и достаточно, чтобы параметры удовлетворяли следующей системе уравнений:
$$\begin{cases} (1-p)^2 = p_1 \\ 2p(1-p) = p_2 \\ p^2 = p_3 \\ p_1 + p_2 + p_3 = 1 \\ p_i \geq 0 \\ p \in [0,1] \end{cases} \Rightarrow \begin{cases} (1-p)^2 = p_1 \\ 2p(1-p) = p_2 \\ p^2 = p_3 \\ (1-p)^2 + 2p(1-p) + p^2 = 1 \\ 2p(1-p) \geq 0 \\ p \in [0,1] \end{cases} \Rightarrow \begin{cases} (1-p)^2 = p_1 \\ 2p(1-p) = p_2 \\ p^2 = p_3 \\ p \in [0,1] \end{cases}$$

2. (3) Пусть U, $V \sim U[0,1]$ и они независимы. Определим
$$X = \sqrt{-\ln V}\cos(2\pi U), \quad Y = \sqrt{-\ln V}\sin(2\pi U)$$
Показать, что $Z = (X, Y)$ – стандартный гауссовский вектор.
воспользуемся определением: для распределений с невырожденной ковариационной матрицей верно, что случайный вектор Z является гауссовским, если плотность задается следующей формулой:
$$f_Z(z) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp -\frac{1}{2}(z-\mu)^\top \Sigma^{-1}(z-\mu)$$
где Σ – ковариационная матрица, а μ – вектор средних значений. ну погнали..
$$f_U(u) = f_V(v) = \begin{cases} 1, \in [0,1] \\ 0, \text{ иначе} \end{cases}$$
$$f_{U,V}(u,v) = f_U(u)f_V(v), \text{ так как } U \text{ и } V \text{ независимы} = \begin{cases} 1, (u,v) \in [0,1]\times[0,1] \\ 0, \text{ иначе} \end{cases}$$
$$\begin{aligned} \mathbb{E}[X] &= \mathbb{E}[\sqrt{-\ln V}\cos(2\pi U)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{-\ln v}\cos(2\pi u) f_{U,V}(u,v)\,\mathrm{d}u\,\mathrm{d}v = \int_0^1\int_0^1 \sqrt{-\ln v}\cos(2\pi u)\,\mathrm{d}u\,\mathrm{d}v \\ &= \int_0^1 \frac{1}{2\pi}\sqrt{-\ln v}\sin(2\pi u)\Big|_0^1 \mathrm{d}v = \int_0^1 0\,\mathrm{d}v = 0 \\ \mathbb{D}[X] &= \mathbb{E}[X^2] - 0 = \mathbb{E}[-\ln V\cos^2(2\pi U)] = \int_0^1\int_0^1 -\ln v\cos^2(2\pi u)\,\mathrm{d}u\,\mathrm{d}v = \int_0^1 -\frac{\ln v}{2}\,\mathrm{d}v = \frac{1}{2} \\ \mathbb{E}[Y] &= \mathbb{E}[\sqrt{-\ln V}\sin(2\pi U)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{-\ln v}\sin(2\pi u) f_{U,V}(u,v)\,\mathrm{d}u\,\mathrm{d}v = \int_0^1\int_0^1 \sqrt{-\ln v}\sin(2\pi u)\,\mathrm{d}u\,\mathrm{d}v \\ &= \int_0^1 \frac{-1}{2\pi}\sqrt{-\ln v}\cos(2\pi u)\Big|_0^1 \mathrm{d}v = \int_0^1 0\,\mathrm{d}v = 0 \\ \mathbb{D}[Y] &= \mathbb{E}[Y^2] - 0 = \mathbb{E}[-\ln V\sin^2(2\pi U)] = \int_0^1\int_0^1 -\ln v\sin^2(2\pi u)\,\mathrm{d}u\,\mathrm{d}v = \int_0^1 -\frac{\ln v}{2}\,\mathrm{d}v = \frac{1}{2} \end{aligned}$$

$$\operatorname{cov}(X,Y) = \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])] = \mathbb{E}[-\ln V\cos(2\pi U)\sin(2\pi U)] = \int_0^1\int_0^1 -\ln v\cos(2\pi U)\sin(2\pi u)\,\mathrm{d}u\,\mathrm{d}v$$
$$= \int_0^1 -\frac{1}{4\pi}\ln v\sin^2(2\pi u)\Big|_0^1\mathrm{d}v = \int_0^1 0\,\mathrm{d}v = 0$$

итого получаем ковариационную матрицу Σ и вектор средних значений μ:

$$\Sigma = \begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix} \quad \mu = \begin{bmatrix}0\\0\end{bmatrix}$$

$$|\Sigma| = \frac{1}{4} \quad \Sigma^{-1} = \begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$$

$$f_Z(z) = \frac{1}{2\pi|\Sigma|^{1/2}}\exp -\frac{1}{2}(z-\mu)^\top\Sigma^{-1}(z-\mu) = \frac{1}{\pi}\exp -\frac{2x^2+2y^2}{2} = \frac{1}{\pi}\exp -(x^2+y^2)$$

проверим, что плотность действительно такая, по формуле для вычисления плотности преобразования случайных величин для двумерного случая (из 6 дз M32391):

$$f_{X,Y}(x,y) = f_{U,V}(u(x,y),v(x,y))|\det D(u(x,y),v(x,y))|,\ \ D(.) - \text{матрица Якоби}$$

$$\sqrt{-\ln v} = \frac{x}{\cos(2\pi u)} = \frac{y}{\sin(2\pi u)} \ \Rightarrow\ \tan(2\pi u) = \frac{y}{x} \ \Rightarrow\ u = \frac{1}{2\pi}\arctan\frac{y}{x}$$
$$\ln\frac{1}{v} = \frac{x^2}{\cos^2(2\pi u)} \ \Rightarrow\ \frac{1}{v} = \exp\frac{x^2}{\cos^2(2\pi u)} \ \Rightarrow\ v = \exp\frac{-x^2}{\cos^2(2\pi u)} = \exp\frac{-x^2}{\cos^2\arctan(y/x)} =$$
$$\left[w := \arctan\frac{y}{x} \ \Rightarrow\ \frac{y}{x} = \tan w = \frac{\sin w}{\cos w} \ \Rightarrow\ \frac{y^2}{x^2}+1 = \frac{\sin^2 w}{\cos^2 w}+1 = \frac{1}{\cos^2(w)} = \frac{1}{\cos^2\arctan(y/x))}\right]$$
$$= \exp -(x^2+y^2)$$
$$\det D = \begin{vmatrix}\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y}\\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{vmatrix} = \begin{vmatrix}-y/(2\pi(x^2+y^2)) & x/(2\pi(x^2+y^2))\\ -2x\exp-(x^2+y^2) & -2y\exp-(x^2+y^2)\end{vmatrix} = \frac{1}{\pi}\exp -(x^2+y^2)$$

получаем

$$f_{X,Y}(x,y) = \frac{1}{\pi}\exp -(x^2+y^2)$$

и правда совпадает, значит вектор является гауссовским. стандратным, к сожалению, не является (для этого Σ должна была быть единичной матрицей)

3. (3) Для заданной плотности p наследуйтесь от класса `rv_continuous` и сгенерируйте с помощью реализованного подкласса для вашего распределения n случайных чисел из данного распределения. Посмотрите, как будет работать алгоритм при росте n (сразу большим n не делайте). Реализуйте генерацию для вашего распределения с помощью обратной к функции распределения (при написании функции для F^{-1} вызов специальных функций, например, квантили стандартного нормального закона разрешается). Проведите тот же эксперимент. Выберите еще один метод для генерации случайных чисел (можно, например, rejecting sampling, ratio of uniforms или другой метод). Опишите его математическое обоснование. Воспользуйтесь реализацией или сами реализуйте. Проведите тот же эксперимент.

$$p(x) = \frac{4\ln^3 x}{x}\mathbb{1}(x\in[1,e])$$

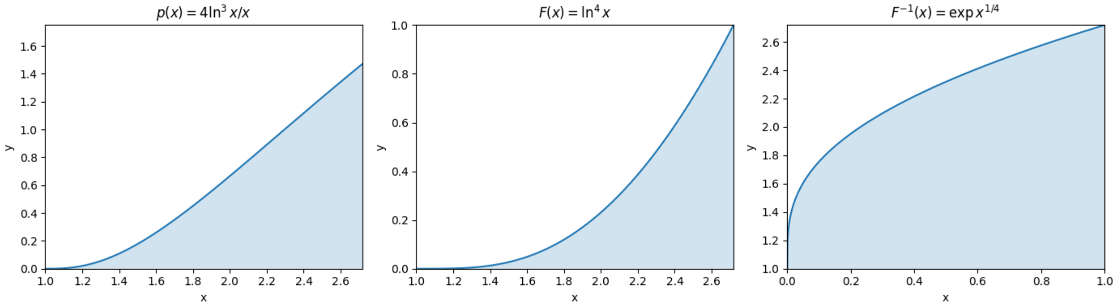
найдем функцию распределения:

$$F(x) = \int_1^x 4\ln^3 t\frac{\mathrm{d}t}{t} = \ln^4 t\Big|_1^x = \ln^4 x,\ \ x\in[1,e]$$

и обратную к ней:

$$F^{-1}(x) = \exp(x^{\frac{1}{4}}),\ \ x\in[0,1]$$

вот такие красивые:



метод обратного преобразования описан здесь, про rejected sampling:
если данная нам функция плотности некой случайной величины ξ слишком страшная и сложная для генерации, то найдем другую функцию плотности $g(x)$, попроще, такую, что $\forall x\ p(x) \le c \cdot g(x)$. и план такой:

(a) генерируем y по g

(b) принимаем значение с вероятностью $f(y)/(c \cdot g(y))$, иначе возвращаемся к (a)

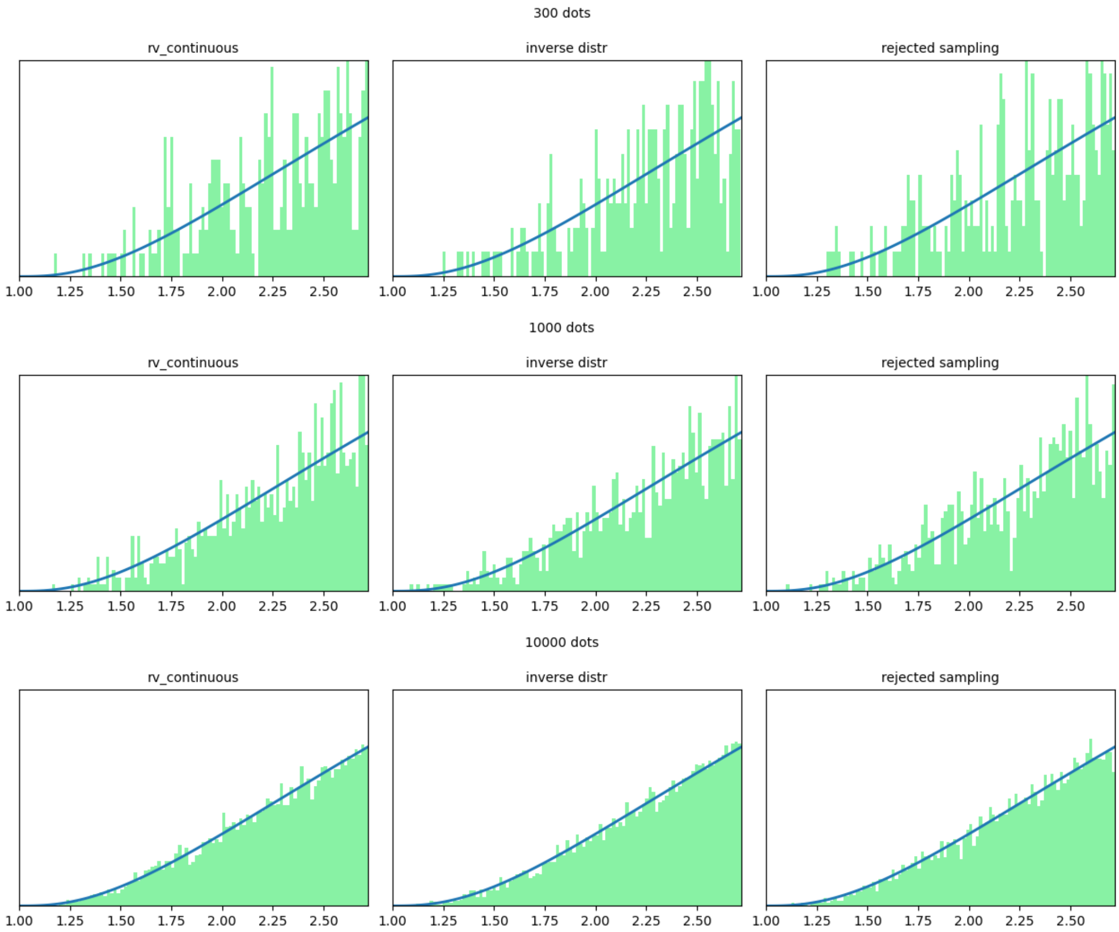
принять значение с вероятностью $f(y)/(c \cdot g(y)) \in [0, 1]$ эквивалентно тому, что $u \le f(y)/(c \cdot g(y))$, где значение u сгенерировано по стандартному равномерному распределению.
докажем, что $Y|U \le f(Y)/(c \cdot g(Y)) \sim \xi$, то есть они имеют одинаковое распределение
$\exists\ B \subseteq [1, e] \subseteq \mathbb{R}$. тогда

$$\begin{aligned}
\mathbb{P}\Big(Y \in B \Big| U \le \frac{f(Y)}{c \cdot g(Y)}\Big) &= \mathbb{P}\Big(Y \in B, U \le \frac{f(Y)}{c \cdot g(Y)}\Big) \Big/ \mathbb{P}\Big(U \le \frac{f(Y)}{c \cdot g(Y)}, Y \in \mathbb{R}\Big) \\
&\text{(формула байеса, добавляем } Y \in \mathbb{R} \text{ за бесплатно)} \\
&= \int_B \mathbb{P}\Big(U \le \frac{f(Y)}{c \cdot g(Y)}, Y = y\Big) g(y)\,\mathrm{d}y \Big/ \int_{\mathbb{R}} \mathbb{P}\Big(U \le \frac{f(Y)}{c \cdot g(Y)}, Y = y\Big) g(y)\,\mathrm{d}y \\
&\text{(формула полной вероятности)} \\
&= \int_B \mathbb{P}\Big(U \le \frac{f(y)}{c \cdot g(y)}\Big) g(y)\,\mathrm{d}y \Big/ \int_{\mathbb{R}} \mathbb{P}\Big(U \le \frac{f(y)}{c \cdot g(y)}\Big) g(y)\,\mathrm{d}y \\
&\text{(так как } U \sim U[0,1] \text{ и } \frac{f(y)}{c \cdot g(y)} \in [0,1]) \\
&= \int_B \frac{f(y)}{c \cdot g(y)} g(y)\,\mathrm{d}y \Big/ \int_{\mathbb{R}} \frac{f(y)}{c \cdot g(y)} g(y)\,\mathrm{d}y \\
&= \int_B f(y)\,\mathrm{d}y \Big/ \int_{\mathbb{R}} f(y)\,\mathrm{d}y = \frac{\mathbb{P}(\xi \in B)}{\mathbb{P}(\xi \in \mathbb{R})} = \mathbb{P}(\xi \in B) \\
&\Rightarrow Y|U \le f(Y)/(c \cdot g(Y)) \sim \xi, \text{ чтд}
\end{aligned}$$

реализация:

```
p = lambda x: 4.0*np.log(x)**3/x  # плотность
a, b = 1.0, math.e   # границы
F = lambda x: np.log(x)**4  # функция распределения
F_inverse = lambda x: np.exp(x**0.25)  # обратная

class distribution(rv_continuous):
    def _pdf(self, x):
        return p(x)

rv_dis = distribution(a=a, b=b)

def get_rv_dots(n):
    return rv_dis.rvs(size=n)

def get_inverse_dots(n):
    return F_inverse(np.random.random(n))

g = lambda x: 1.0/(b-a)  # плотность попроще (равномерное распределение)

def get_rejected_sampling_dots(n):
    x, c = np.array([]), 4.0*(math.e-1)/math.e
    while x.size < n:
        y, u = np.random.uniform(a, b), np.random.uniform(0, 1)
        if u <= p(y)/(c*g(y)):
            x = np.append(x, y)
    return x
```

чтож, оно действительно работает..:



ниже представлена табличка времени работы, в зависимости от метода и количества запрашиваемых точек.

dots	rv_continuous	inverse distr	rejected sampling
100	0.0406930447	2.47955e-05	0.0009970665
1000	0.3891251087	2.71797e-05	0.0086219311
10000	3.8463330269	0.0001690388	0.0992381573
50000	19.2876479626	0.0007810593	0.6470673084

как мы и ожидали, прочитав условие задания, отнаследоваться от `rv_continuous` не всегда оказывается лучшим вариантом: генерация работает намного дольше, чем самодельные аналоги. метод обратного преобразования оказывается самым быстрым, но надо вспомнить, что тут потребовалась и функцию распределения найти, и обратную к ней, и это тоже может быть не сильно просто (и не всегда возможно). rejected sampling в целом тоже показывает себя довольно неплохо, но так же вспоминаем про цикл аля `while True` в реализации и поиск подходящей плотности/константы, который мы делали руками, и плачем. вывод: выбор метода явно зависит от поставленной задачи. если нужно сгенерировать 10 точек и лень думать, то наследуемся `rv_continuous` и живем счастливую жизнь, иначе иначе

4. (2) Пусть $X_1, ..., X_n$ – i.i.d. величины из некоторого распределения, которое параметризуется параметром $\theta \in E$. Пусть $\overline{X}_n = (X_1 + ... + X_n)/n$, μ_θ математическое ожидания случайной величины X_i. Найти с помощью неравенства Чебышёва и центральной предельной теоремы номер n, при котором равномерно для $\theta \in E$ и заданных $\delta, \varepsilon \in (0, 1)$ выполняется соотношение

$$\mathbb{P}(|\overline{X}_n - \mu_n| \le \varepsilon) \ge 1 - \delta.$$

Для найденного по неравенству Чебышёва n сгенерировать 100 выборок найденного объема n и посчитать количество и долю выборок, для которых $|\overline{X}_n - \mu_n| \le \varepsilon$. Параметры эксперимента: $\varepsilon = 0.01$, $\delta = 0.05$ и θ, при котором найдена равномерная оценка. То же самое сделать и для n, найденного с помощью ЦПТ. В вариантах указывается класс распределений, затем множество возможных значений параметра E.

$$\mathrm{Pois}(\lambda), \lambda \in (0, 10)$$

для $X_i \sim \text{Pois}(\lambda)$ известны их математические ожидания и дисперсии: все $= \lambda$. по неравенству чебышева имеем

$$\mathbb{P}(|\overline{X}_n - \mu_n| \le \varepsilon) = \mathbb{P}(\left|\frac{X_1 + ... + X_n}{n} - \lambda\right| \le \varepsilon) = \mathbb{P}(|X_1 + ... + X_n - n\lambda| \le n\varepsilon)$$
$$\ge 1 - \frac{\mathbb{D}[X_1 + ... + X_n]}{n^2\varepsilon^2} = 1 - \frac{n\lambda}{n^2\varepsilon^2} = 1 - \frac{\lambda}{n\varepsilon^2} \ge 1 - \delta$$

получаем $n \ge \lambda/(\varepsilon^2\delta)\big[\varepsilon = 0.01, \delta = 0.05\big] = 200000\lambda$.
теперь цпт:

$$\frac{\overline{X}_n - \mu_\theta}{\sqrt{\sigma_\theta^2/n}} \to \mathcal{N}(0,1) \;\Rightarrow\; \frac{|\overline{X}_n - \lambda|}{\sqrt{\lambda/n}} \to |\mathcal{N}(0,1)|$$

$$\mathbb{P}(|\overline{X}_n - \mu_\theta| \le \varepsilon) = \mathbb{P}\Big(\frac{|\overline{X}_n - \lambda|}{\sqrt{\lambda/n}} \le \frac{\varepsilon}{\sqrt{\lambda/n}}\Big) \approx \Phi(\frac{\varepsilon}{\sqrt{\lambda/n}}) - \Phi(\frac{-\varepsilon}{\sqrt{\lambda/n}}) = \text{erf}(\frac{\varepsilon}{\sqrt{2\lambda/n}}) \ge 1 - \delta$$

подставляем ε и δ и получаем

$$\text{erf}(\frac{0.01}{\sqrt{2\lambda/n}}) \ge 0.95 \;\Rightarrow\; \frac{0.01}{\sqrt{2\lambda/n}} \ge 1.3859 \;\Rightarrow\; n \ge 38415\lambda$$

```python
def test_cheb_cbt():
    table = PrettyTable()
    table.field_names = ['lambda', 'n cheb', 'n clt', '% cheb', '% clt']
    eps = 0.01
    for lamb in range(1, 11):
        n_cheb, n_clt = 200000 * lamb, 38415 * lamb
        cheb = sum([1 if abs(np.mean(poisson.rvs(mu=lamb, size=n_cheb))-lamb) <= eps else 0 for _ in range(100)])
        clt = sum([1 if abs(np.mean(poisson.rvs(mu=lamb, size=n_clt))-lamb) <= eps else 0 for _ in range(100)])
        table.add_row([lamb, n_cheb, n_clt, cheb, clt])
    print(table)
```

lambda	n cheb	n clt	% cheb	% clt
1	200000	38415	100	93
2	400000	76830	100	95
3	600000	115245	100	92
4	800000	153660	100	95
5	1000000	192075	100	96
6	1200000	230490	100	100
7	1400000	268905	100	96
8	1600000	307320	100	97
9	1800000	345735	100	97
10	2000000	384150	100	98

здесь мы посчитали, в скольких из ста выборок распределения размера, полученного выше, выполняется неравенство $\mathbb{P}(|\overline{X}_n - \mu_n| \le \varepsilon)$. по чебышеву числа n получились ну очень большие, поэтому в 100% случаев неравенство выполнилось. а вот с ЦПТ, кажется, получилось найти неплохо приближенное значение n – полученные значения недалеки от искомых 95%. вывод: ЦПТ справляется лучше, верить следует ей.
насчет равномерного выполнения неравенства для $\theta \in E$: не очень понимаю формулировку, но если имеется в виду найти такое n, что при всех λ неравенство будет выполнено, то просто смотрим на строку $\lambda = 10$, так как в обоих способах формула для нахождения n зависит от λ и возрастает с возрастанием параметра.