https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/nbextensions/latex_envs/README.html (https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/nbextensions/latex_envs/README.html)

Ввод [1]:
```
#%pip install jupyter_contrib_nbextensions
```

Ввод [2]:
```
#%pip install jupyter_latex_envs
```

Ввод [3]:
```
#%pip install jupyter_latex_envs --upgrade
```

Ввод [10]:
```
from PIL import ImageGrab
from IPython.display import display, Image
def ins(ratio=1.0):
    im_data = ImageGrab.grabclipboard()
    new_size = tuple([int(i*ratio) for i in im_data.size])
    thumb = im_data.resize(new_size)
    fn = "temp.PNG"
    thumb.save(fn)
    img = Image(filename=fn)
    display(img)
```

Ввод [11]:
```
import numpy as np
from itertools import *
from more_itertools import *
from scipy.stats import *
from sympy import *
from scipy.special import *
import math

```

Ввод [12]:
```
import locale as loc
loc.setlocale(loc.LC_ALL, 'ru')
init_printing(use_unicode=True,use_latex=True)
```

Ввод [13]:
```
from IPython.display import display, Math, Latex
```

Ввод [14]:
```
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
from matplotlib import rcParams
#########################
import locale
locale.setlocale(locale.LC_NUMERIC, 'russian')
plt.rcParams['axes.formatter.use_locale'] = True
#########
#########
plt.rcParams['font.size'] = 36
plt.rcParams["font.family"] = "Times New Roman"
plt.rcParams['mathtext.fontset'] = 'cm'
###################
```

Лекция № 1. Классическое определение вероятности

Классическое определение вероятности

Рассмотрим стохастический эксперимент, который состоит из n ***одинаково возможных исходов*** ω_i, т.е. $|\Omega| = n$, $\mathbb{P}(\omega_i) = \frac{1}{n}$. Предположим, что событию A благоприятствует m из этих исходов т.е. $|A| = m$. Тогда

Пример .1.

Корзина содержит 23 шара: 8 белых, 6 синих и 9 красных. На каждом шаге из корзины "наудачу" извлекается шар и назад в корзину не возвращается. Исход n последовательных извлечений называется ***выборкой объема n без возвращения или бесповторной выборкой***. Из корзины наудачу (без возвращения) извлечены 6 шаров. Найти вероятности следующих событий и показать их статистическую устойчивость:

- а) все извлеченные шары красного цвета (событие A);
- б) в выборке окажется 3 синих, 2 белых и 1 красный;
- в) в точности 4 белых шара.

Решение.

1. Пространство элементарных исходов
$$\Omega = \{\omega = (6 \text{ вынутых шаров, порядок не учитываем}\}, |\Omega| = C_{23}^6 = \frac{23!}{6!17!} = 100\,947.$$
(комментарий, поскольку шары обратно не возвращаются и порядок вынутых шаров не важен)
2. A={все шары красного цвета}, $|A| = C_9^6 \cdot C_8^0 \cdot C_6^0 = 84$
3. $$\mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{C_9^6 \cdot C_8^0 \cdot C_6^0}{C_{23}^6} = \frac{84}{100947} = \frac{4}{4807} \approx 0,000832.$$

Ответ:

а) $\mathbb{P}(A) \approx 0{,}000832$; б) $\mathbb{P}(B) \approx 0{,}049927$; в) $\mathbb{P}(\mathrm{C}) \approx 0{,}07281$.

Создание корзины (box)

Ввод [7]:
```
def union(A, B):
    "Соединямем элемент A с каждым элементом из B."
    return {a + b
            for a in A for b in B}

box = union('W', '12345678') | union('B', '123456') | union('R', '123456789')
```

Ввод [8]:
```
print(box)
```

```
{'W6', 'W2', 'W8', 'W5', 'R8', 'R3', 'R1', 'B1', 'B6', 'R2', 'W1', 'W4', 'W3', 'B4', 'R5',
'R9', 'W7', 'B2', 'B3', 'R4', 'R6', 'B5', 'R7'}
```

Ввод [9]:
```
len(box)
```

Out[9]: 23

Ввод [10]:
```
def Omega(box, k):
    "Все неупорядоченные комбинации из k шаров"
    return {' '.join(comb)
            for comb in combinations(box, k)} # Функция combinations() модуля itertools

"Элементарные исходы"
omega = Omega(box, 6)

len(omega)
```

Out[10]: 100947

Ввод [11]:
```
1 omega
```

Out[11]:
```
{'W2 W8 W1 W4 R4 R6',
 'W6 W8 W5 W1 W3 R7',
 'W6 W2 W8 W5 W1 R5',
 'W2 W8 W1 B4 B3 R4',
 'W5 R1 B1 W7 B2 R7',
 'W6 R1 B6 W1 R9 R4',
 'B1 B6 R9 B3 R6 B5',
 'R1 B6 R9 B2 B3 B5',
 'B6 R2 W1 R5 R9 R6',
 'W2 R3 B3 R4 R6 B5',
 'R1 B6 W3 W7 R4 B5',
 'W6 W8 W1 B4 R9 R4',
 'R3 R1 B1 W3 B5 R7',
 'W2 W8 W5 B1 B6 B5',
 'W2 R3 B1 B6 R2 R7',
 'W6 W7 B2 R4 R6 R7',
 'W2 B1 B6 W3 R5 R6',
 'W8 R8 R2 B2 R6 R7',
 'W2 B1 B6 W3 B4 R7',
```

Ввод [12]:
```
1 comb(23, 6, exact=True, repetition=False)
```

Out[12]: 100947

Ввод [13]:
```
1 math.comb(23,6)
```

Out[13]: 100947

Стохастический эксперимент

Ввод [14]:
```
1 np.random.choice(list(omega),size=10,replace=False)
```

Out[14]:
```
array(['W8 R3 B6 R5 R9 B5', 'W5 R8 B1 B6 W4 R9', 'W8 W5 B6 R4 B5 R7',
       'W6 W5 B6 R5 B3 R7', 'R2 W3 B4 W7 R4 B5', 'W8 R8 B1 R9 R6 R7',
       'W6 B1 W1 W4 R4 B5', 'W8 R3 W1 W3 B4 R9', 'W8 R8 R3 W7 R4 R6',
       'W5 W4 B4 R5 R4 R7'], dtype='<U17')
```

Ввод [15]:
```
1 from random import *
2 "Случайный выбор 10 исходов, каждый исход это вектор из 6 шаров"
3 sample(list(omega), 10)
```

Out[15]:
```
['R3 R1 R2 W1 W3 B4',
 'W8 W4 R9 B2 B3 R7',
 'W5 R8 R2 W3 R9 W7',
 'W6 W5 B6 W4 B5 R7',
 'W8 W5 B6 W3 W7 B2',
 'W5 R8 R2 R5 R9 B3',
 'W6 W2 R3 B6 R9 B5',
 'W2 W8 R3 B3 B5 R7',
 'W5 R8 B6 R2 W1 R5',
 'W8 W1 B4 R5 B2 R4']
```

Ввод [16]:
```
1 from fractions import Fraction
2 def P(event, space):
3     "Классическое определение вероятности  с одинаково возможными исходами."
4     return Fraction(len(event & space), len(space))
```

Ввод [17]:

```
Cub    = {1, 2, 3, 4, 5, 6}
A1 = {   2,    4,    6}

print(P(A1, Cub))
```

1/2

Ввод [18]:

```
" Событие A={все шары красного цвета}"
A = {r for r in omega if r.count('R') == 6}
A
```

Out[18]: {'R1 R2 R5 R4 R6 R7',
'R1 R2 R5 R9 R4 R6',
'R1 R2 R5 R9 R4 R7',
'R1 R2 R5 R9 R6 R7',
'R1 R2 R9 R4 R6 R7',
'R1 R5 R9 R4 R6 R7',
'R2 R5 R9 R4 R6 R7',
'R3 R1 R2 R4 R6 R7',
'R3 R1 R2 R5 R4 R6',
'R3 R1 R2 R5 R4 R7',
'R3 R1 R2 R5 R6 R7',
'R3 R1 R2 R5 R9 R4',
'R3 R1 R2 R5 R9 R6',
'R3 R1 R2 R5 R9 R7',
'R3 R1 R2 R9 R4 R6',
'R3 R1 R2 R9 R4 R7',
'R3 R1 R2 R9 R6 R7',
'R3 R1 R5 R4 R6 R7',
'R3 R1 R5 R9 R4 R6',
'R3 R1 R5 R9 R4 R7',
'R3 R1 R5 R9 R6 R7',
'R3 R1 R9 R4 R6 R7',
'R3 R2 R5 R4 R6 R7',
'R3 R2 R5 R9 R4 R6',
'R3 R2 R5 R9 R4 R7',
'R3 R2 R5 R9 R6 R7',
'R3 R2 R9 R4 R6 R7',
'R3 R5 R9 R4 R6 R7',
'R8 R1 R2 R4 R6 R7',
'R8 R1 R2 R5 R4 R6',
'R8 R1 R2 R5 R4 R7',
'R8 R1 R2 R5 R6 R7',
'R8 R1 R2 R5 R9 R4',
'R8 R1 R2 R5 R9 R6',
'R8 R1 R2 R5 R9 R7',
'R8 R1 R2 R9 R4 R6',
'R8 R1 R2 R9 R4 R7',
'R8 R1 R2 R9 R6 R7',
'R8 R1 R5 R4 R6 R7',
'R8 R1 R5 R9 R4 R6',
'R8 R1 R5 R9 R4 R7',
'R8 R1 R5 R9 R6 R7',
'R8 R1 R9 R4 R6 R7',
'R8 R2 R5 R4 R6 R7',
'R8 R2 R5 R9 R4 R6',
'R8 R2 R5 R9 R4 R7',
'R8 R2 R5 R9 R6 R7',
'R8 R2 R9 R4 R6 R7',
'R8 R3 R1 R2 R4 R6',
'R8 R3 R1 R2 R4 R7',
'R8 R3 R1 R2 R5 R4',
'R8 R3 R1 R2 R5 R6',
'R8 R3 R1 R2 R5 R7',
'R8 R3 R1 R2 R5 R9',
'R8 R3 R1 R2 R6 R7',
'R8 R3 R1 R2 R9 R4',
'R8 R3 R1 R2 R9 R6',
'R8 R3 R1 R2 R9 R7',
'R8 R3 R1 R4 R6 R7',
'R8 R3 R1 R5 R4 R6',
'R8 R3 R1 R5 R4 R7',
'R8 R3 R1 R5 R6 R7',
'R8 R3 R1 R5 R9 R4',
'R8 R3 R1 R5 R9 R6',
'R8 R3 R1 R5 R9 R7',
'R8 R3 R1 R9 R4 R6',
'R8 R3 R1 R9 R4 R7',
'R8 R3 R1 R9 R6 R7',
'R8 R3 R2 R4 R6 R7',
'R8 R3 R2 R5 R4 R6',
'R8 R3 R2 R5 R4 R7',
'R8 R3 R2 R5 R6 R7',
'R8 R3 R2 R5 R9 R4',

```
 'R8 R3 R2 R5 R9 R6',
 'R8 R3 R2 R5 R9 R7',
 'R8 R3 R2 R9 R4 R6',
 'R8 R3 R2 R9 R4 R7',
 'R8 R3 R2 R9 R6 R7',
 'R8 R3 R5 R4 R6 R7',
 'R8 R3 R5 R9 R4 R6',
 'R8 R3 R5 R9 R4 R7',
 'R8 R3 R5 R9 R6 R7',
 'R8 R3 R9 R4 R6 R7',
 'R8 R5 R9 R4 R6 R7'}
```

Ввод [19]:
```
len(A)
```

Out[19]: 84

Ввод [20]:
```
B = {s for s in omega if
          s.count('B') == 3 and s.count('W') == 2 and s.count('R') == 1}
len(B)
```

Out[20]: 5040

Ввод [21]:
```
C = {s for s in omega if
      s.count('W') == 4}
```

Ввод [22]:
```
len(C)
```

Out[22]: 7350

Ввод [23]:
```
print('P(A)=',P(A, omega),"=", float(P(A, omega)))
```

```
P(A)= 4/4807 = 0.0008321198252548367
```

Ввод [24]:
```
print('P(B)=',P(B, omega),"=", float(P(B, omega)))
```

```
P(B)= 240/4807 = 0.0499271895152902
```

Ввод [25]:
```
print('P(C)=',P(C, omega),"=", float(P(C, omega)))
```

```
P(C)= 350/4807 = 0.07281048470979821
```

Статистическое определение вероятности

Пусть некоторый опыт повторяется (реализуется) N раз. Для события A, связанного с данным опытом, обозначим через $N(A)$ его ***частоту*** -- количество реализаций, в которых "наблюдалось" (наступило) событие A. Определим ***относительную частоту*** $\hat{p}(A)$ события A как отношение частоты события A к N, т.е. $\hat{p}(A) \stackrel{\text{def}}{=} \frac{N(A)}{N}$.

Статистическое определение вероятности утверждает, что при большом числе реализаций опыта N выполняется приближенное равенство

$$\mathbb{P}(A) \approx \hat{p}(A) \stackrel{\text{def}}{=} \frac{N(A)}{N}.$$

Формально статистическое определение вероятности очень похоже на классическое определение вероятности: и в том и другом определении вероятность равна отношению. Различие, конечно, кроется в том смысле, который придается букве "эн":

- $n = |\Omega|$ -- число элементарных исходов;
- N -- число реализаций опыта.

Ввод [26]:
```
Omega=[0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,2,2]
omega=list(np.random.choice(Omega,6,replace=False)) #(без возвращения)
omega
```

Out[26]: [2, 0, 2, 0, 1, 1]

Ввод [27]:
```
import random
Omega=[0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,2,2]
count=0
N=1550000
x=[]
y=[]
for i in range(1,N+1):
    x.append(i)
    #omega=random.sample(Omega, k=6)
    omega=list(np.random.choice(Omega,6,replace=False)) #(без возвращения)
    #omega=list(np.random.choice(Omega,6,replace=True))  #(с возвращением)
    if omega.count(0) == 6:
        count+=1
    y.append(count/(i))
pstatA=count/N
print(pstatA)
```

```
0.0008870967741935484
```

Ввод [28]:
```
y.count(0)
```

Out[28]: 4066

Ввод [29]:
```
random.sample(list(omega), k=6)
```

Out[29]: [2, 0, 1, 2, 0, 0]

https://docs.python.org/3/library/random.html (https://docs.python.org/3/library/random.html)

"random.sample(population, k)

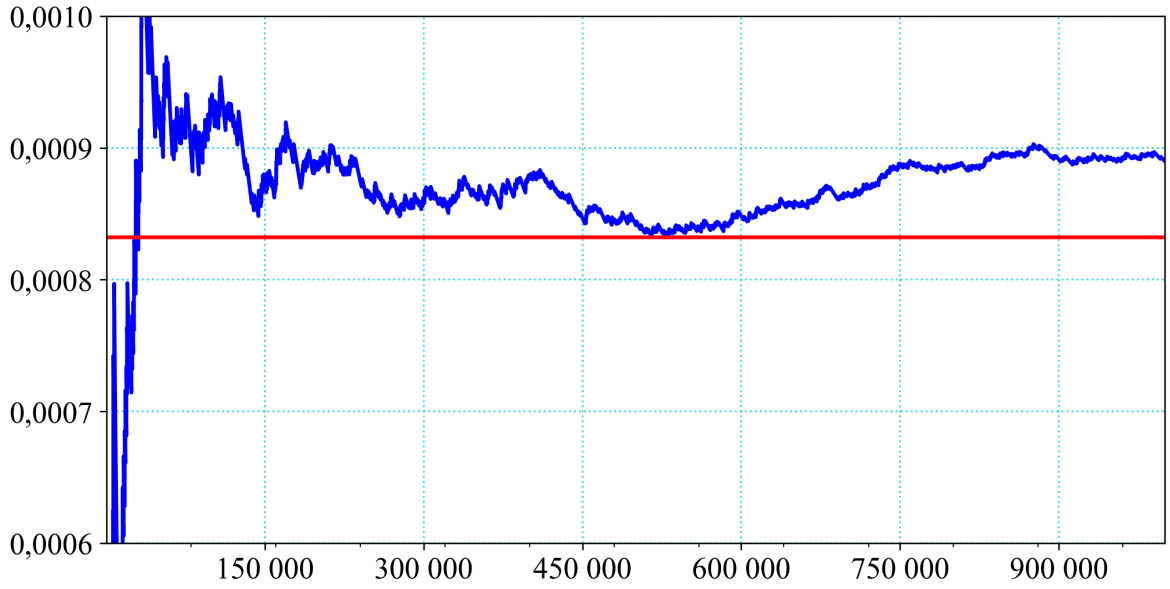
Return a k length list of unique elements chosen from the population sequence or set. Used for random sampling \textbf{without replacement}."

random.choices(population, weights=None, *, cum_weights=None, k=1)

Return a k sized list of elements chosen from the population \textbf{ with replacement.}

Ввод [30]:

```
rcParams['figure.figsize'] = (30, 20)
rcParams['figure.dpi'] = 300
import matplotlib.pyplot as plt
fig,ax =plt.subplots(figsize=(10, 5))
plt.tick_params(labelsize = 16)
plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=1)
ax.xaxis.set_major_locator(ticker.MaxNLocator(8))
ax.xaxis.set_minor_locator(ticker.MaxNLocator(15))
plt.ylim(0.0006,0.001)
plt.xlim(250,1000000)
#####################################################################
plt.plot(x,y,color='b',lw=2)
plt.plot(x,[0.0008321198252548367]*len(x),color='r',lw=2)
plt.show()
fig.savefig("Sol_2_2a.pdf", bbox_inches='tight')
```



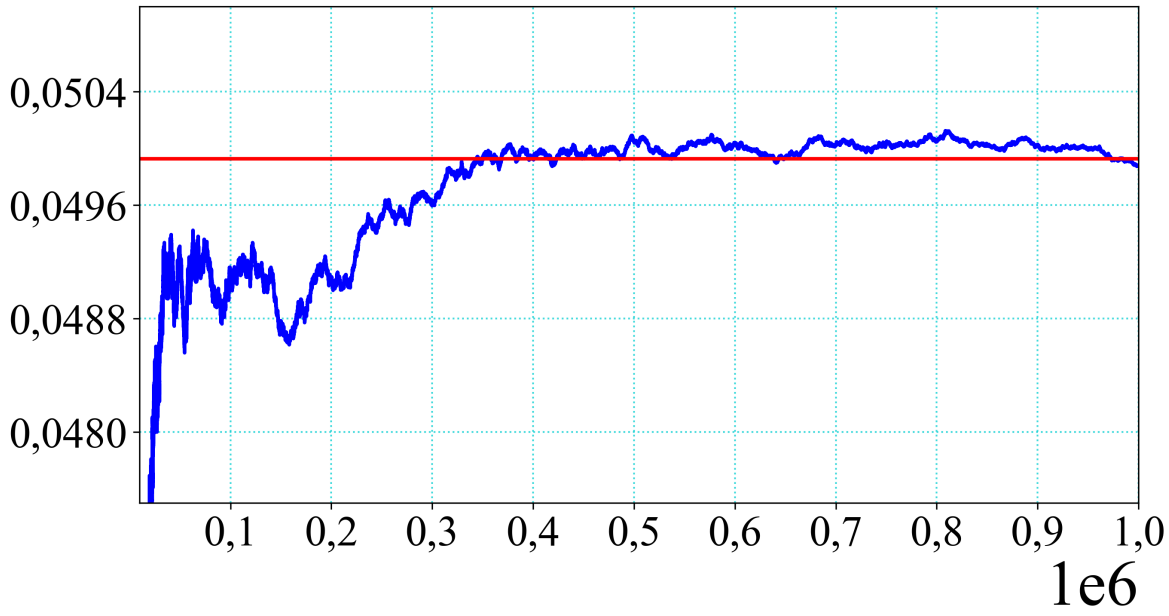
Ввод [31]:

```
import random
Omega=[0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,2,2] #0 -- красный шар, 1 -- синий шар,
count=0
N=1550000
x=[]
y=[]
for i in range(N):
    x.append(i+1)
    #omega=random.sample(Omega, k=6)
    omega=list(np.random.choice(Omega,6,replace=False)) #(без возвращения)
    #omega=list(np.random.choice(Omega,6,replace=True))  #(с возвращением)
    if omega.count(0) == 1 and omega.count(1) == 3 and omega.count(2) == 2:
        count+=1
    y.append(count/(i+1))
pstatB=count/N
print(pstatB)
```

```
0.049820645161290325
```

Ввод [32]:

```
fig,ax =plt.subplots(figsize=(10, 5))
plt.tick_params(labelsize = 24)
plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=1)
ax.xaxis.set_major_locator(ticker.MaxNLocator(10))
ax.xaxis.set_minor_locator(ticker.MaxNLocator(10))
ax.yaxis.set_major_locator(ticker.MaxNLocator(5))
plt.ylim(0.0475,0.051)
plt.xlim(10000,1000000)
#################################################################
plt.plot(x,y,color='b',lw=2)
plt.plot(x,[0.0499271895152902]*len(x),color='r',lw=2)
plt.show()
fig.savefig("pictureB.pdf", bbox_inches='tight')
```



Ввод [33]:

```
?random.sample
```

Задача. Найдите вероятность события A, если "статистическая вероятность" $\hat{p}(A) \approx 0.0035345098039215686$.

Ввод [34]:

```
import random
Omega=[0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,2,2]
count=0
N=2550000
x=[]
y=[]
for i in range(N):
    x.append(i+1)
    #omega=random.choices(Omega,weights=None, cum_weights=None, k=6)
    omega=np.random.choice(Omega, 6,replace=True)
    if np.count_nonzero(omega == 0) == 6:
        count+=1
    y.append(count/(i+1))
pstat=count/N
print(pstat)
```

```
0.003604313725490196
```

Ввод [35]:

```
(9/23)**6
```

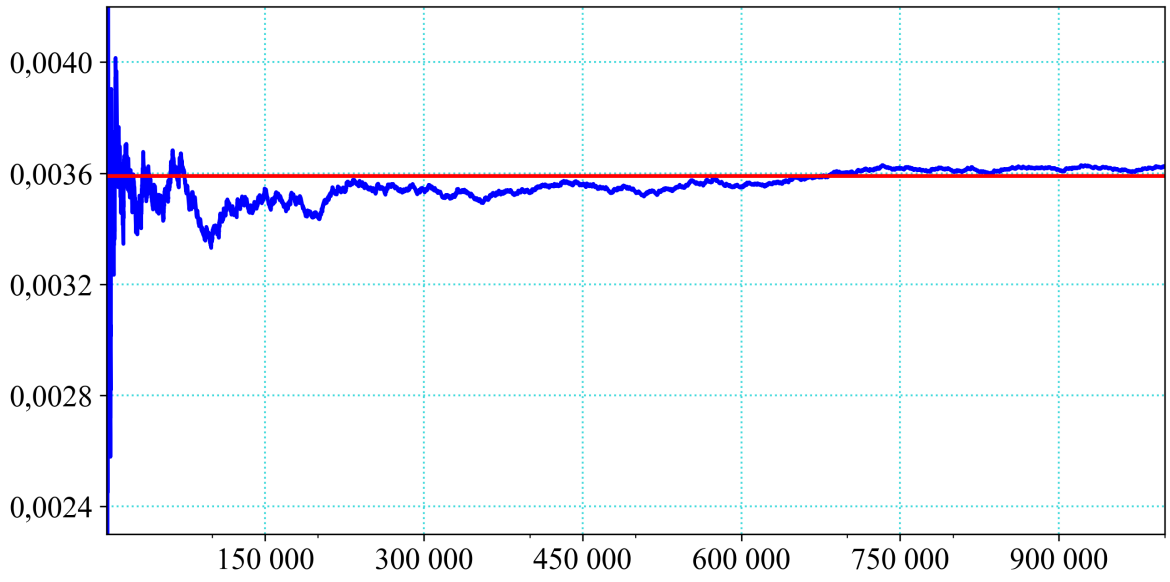
Out[35]: 0.00358994702966927

Ввод [36]:

```
rcParams['figure.figsize'] = (20, 10)
rcParams['figure.dpi'] = 300

import matplotlib.pyplot as plt
#fig=plt.figure()
fig,ax =plt.subplots(figsize=(10, 5))
ax.xaxis.set_major_locator(ticker.MaxNLocator(9))
ax.xaxis.set_minor_locator(ticker.MaxNLocator(10))
ax.yaxis.set_major_locator(ticker.MaxNLocator(5))

plt.tick_params(labelsize = 16)
plt.ylim(0.0023,0.0042)
plt.xlim(100,1000000)
plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=1)
plt.plot(x,y,color='b',lw=2)
plt.plot(x,[0.00358994702966927]*len(x),color='r',lw=2)
plt.show()
fig.savefig("Sol_2_2a_new.pdf", bbox_inches='tight')
```



Ввод [ ]:

```
```