

```
Ввод [1]: 1 from PIL import ImageGrab
2 from IPython.display import display, Image
3
4 def ins(ratio=1.0):
5     im_data = ImageGrab.grabclipboard()
6     new_size = tuple([int(i*ratio) for i in im_data.size])
7     thumb = im_data.resize(new_size)
8     fn = "temp.PNG"
9     thumb.save(fn)
10    img = Image(filename=fn)
11    display(img)
```

```
Ввод [2]: 1 import numpy as np
2 #####
3 from scipy.stats import * # Основная библиотека в этом семестре
4 #####
5 from sympy import *
6 from scipy.special import *
7 import math
```

```
Ввод [4]: 1 # Настройка графика
2 import matplotlib.pyplot as plt # Основная библиотека
3 import matplotlib.ticker as ticker
4 from matplotlib import rcParams
5 #####
6 import locale
7 locale.setlocale(locale.LC_NUMERIC, 'russian')
8 plt.rcParams['axes.formatter.use_locale'] = True
9 #####
10 #####
11 plt.rcParams['font.size'] = 36
12 plt.rcParams["font.family"] = "Times New Roman"
13 plt.rcParams['mathtext.fontset'] = 'cm'
14 #####
```

```
Ввод [10]: 1 ins(1)
```

3.1. Два лица A и B условились встретиться в определенном месте между двумя и тремя часами дня. Лицо A ждет другого в течение 10 минут, после чего уходит; лицо B ждет другого в течение 12 минут. Считая, что момент прихода на встречу выбирается каждым «наудачу» в пределах указанного часа, найдите вероятность, что встреча состоится. Найти соответствующую статистическую вероятность и показать статистическую устойчивость вероятности в зависимости от числа экспериментов N .

Ответ: 0,332778 Статистическая устойчивость события A для числа экспериментов $N = 150\,000$ с детализацией изображена на рисунке (статистическая вероятность $p_{stat} = 0.33268$):



Рис. 1. Ответ к задаче 3.1.

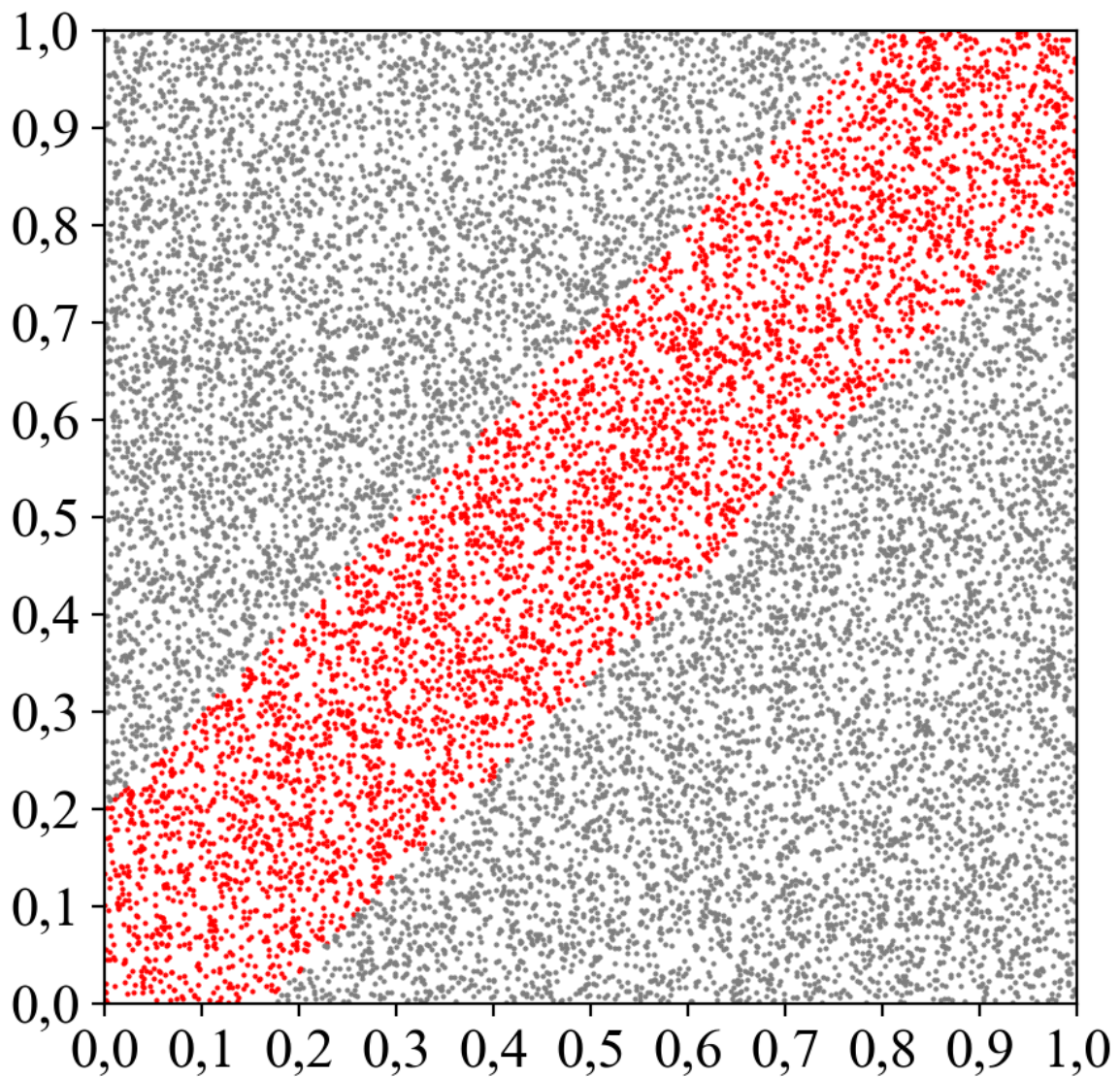
```
Ввод [5]: 1 U1=uniform() # создаем равномерное (непрерывное) распределение на отрезке [0;1]
2 U2=uniform()
3
```

```
Ввод [7]: 1 U1.rvs(size=1)[0] #случайно взятое наугад число из отрезка [0;1] (метод rvs для большинства других распределений!!!)
```

Out[7]: 0.050102305577813

Ввод [8]:

```
1 rcParams['figure.figsize'] = (5, 5)
2 rcParams['figure.dpi'] = 200
3 fig,ax =plt.subplots(figsize=(5, 5))
4 plt.tick_params(labelsize = 16)
5 #plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=1)
6 ax.xaxis.set_major_locator(ticker.MultipleLocator(0.1))
7 ax.yaxis.set_major_locator(ticker.MultipleLocator(0.1))
8 plt.xlim(0,1)
9 plt.ylim(0,1)
10 ax.set_aspect('equal', adjustable='box')
11 #ax.xaxis.set_major_locator(ticker.MaxNLocator(10))
12 #ax.xaxis.set_minor_locator(ticker.MaxNLocator(8))
13 #plt.axis('equal')
14 #####
15 import random
16 N=15000
17 count=0
18 for i in range(N):
19     x = U1.rvs(size=1)[0]
20     y = U2.rvs(size=1)[0]
21     #x = random.uniform(0,1)
22     #y = random.uniform(0,1)
23     if ????: # должно быть условие для события A, что встреча состоится
24         plt.plot(x,y,marker='.', markersize=1,color = 'r')
25         count+=1
26     else:
27         plt.plot(x,y,marker='.', markersize=1,color = 'gray')
28 plt.show()
29 #fig.savefig("Sol_4_1.pdf", bbox_inches='tight')
30 #fig.savefig("Sol_3_1.png", bbox_inches='tight')
31 pstat=count/N
32 pstat
```



Out[8]: 0.3261333333333333

Ввод [11]:

```
1 prob=0.332778
```

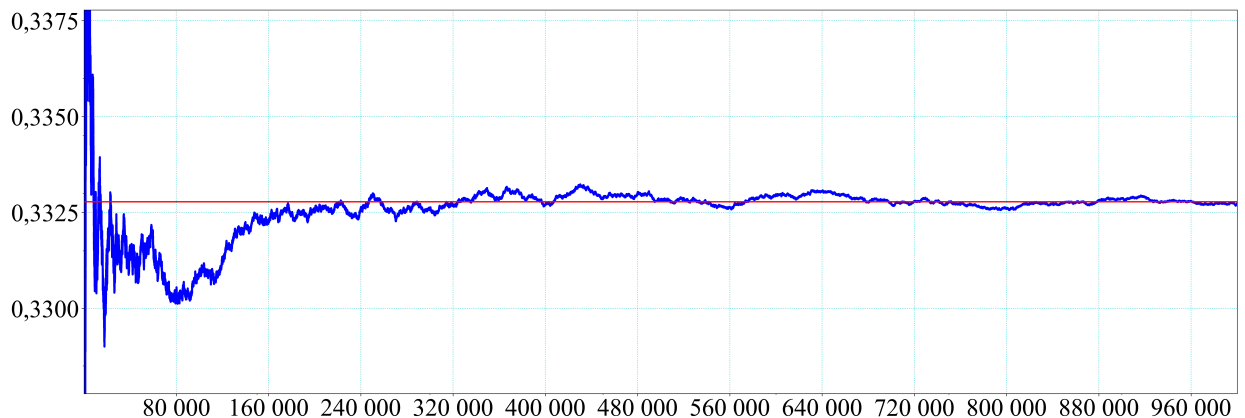
Ввод [9]:

```
1 N=1000000
2 count=0
3 x=[]
4 y=[]
5 for i in range(N+1):
6     x.append(i+1)
7     x1 = random.uniform(0,1) # Можно и так, но лучше через метод rvs для равномерного распределения из U[0;1]
8     y1 = random.uniform(0,1)
9     if ????: # должно быть условие для события A, что встреча состоится
10         plt.plot(x,y,'ro',color="red")
11         plt.plot(x,y,marker='.', markersize=1,color='red')
12         count+=1
13     y.append(count/(i+1))
14 plt.show()
15 #fig.savefig("Sol_3_1.png", bbox_inches='tight')
16 pstat=count/N
17 pstat
```

Out[9]: 0.332725

Ввод [12]:

```
1 rcParams['figure.figsize'] = (15, 10)
2 rcParams['figure.dpi'] = 300
3
4 fig,ax =plt.subplots(figsize=(30, 10))
5 plt.tick_params(labelsize = 36)
6 #ax.xaxis.set_major_locator(ticker.MultipleLocator(200000))
7 ax.yaxis.set_major_locator(ticker.MultipleLocator(0.005))
8 ax.xaxis.set_major_locator(ticker.MaxNLocator(15))
9 ax.xaxis.set_minor_locator(ticker.MaxNLocator(8))
10 ax.yaxis.set_major_locator(ticker.MaxNLocator(5))
11 ax.yaxis.set_minor_locator(ticker.MaxNLocator(8))
12 plt.ylim(prob-0.005,prob+0.005)
13 plt.xlim(0.1,N)
14 #fig=plt.figure()
15 #plt.tick_params(labelsize = 20)
16 plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=1)
17 plt.plot(x,y,color='b',lw=3)
18 plt.plot(x,[prob]*len(x),color='r',lw=2)
19 plt.show()
20 fig.savefig("Sol_3_1stat.png", bbox_inches='tight')
```



Ввод []:

1