

Практическая работа № 1¹

Основные понятия статистического эксперимента

Валерий Сергеевич Верхотуров

БСБО-05-20

¹ <https://github.com/ValeryVerkhoturov/systems-modeling/blob/master/практика%201/practice%201.ipynb>

Задание 1.

Средствами системы MathCAD реализовать подпрограмму CALC_PI, реализующую вычисление значения числа π в соответствии с описанной моделью. Входными параметрами подпрограммы должны являться координаты центра окружности - x_0 и y_0 и значение ее радиуса - r_0 , а также число экспериментов для генератора случайных чисел - $ExpNmb$. Выходным значением должно являться вычисленное значение числа π . Для генерации случайных значений на интервале от 0 до 1 использовать встроенную функцию MathCAD – *rnd*.

```
def calc_pi(x_0, y_0, r, exp_nmb):  
    m = 0  
    x_min = x_0 - r  
    x_max = x_0 + r  
    y_min = y_0 - r  
    y_max = y_0 + r  
    for i in range(exp_nmb):  
        xp = random.uniform(x_min, x_max)  
        yp = random.uniform(y_min, y_max)  
        if (xp - x_0) ** 2 + (yp - y_0) ** 2 <= r ** 2:  
            m += 1  
    pi = 4 * m / exp_nmb  
    return pi
```

Задание 2.

С помощью подпрограммы CALC_PI проводится расчет значения числа для заданной окружности и числа экспериментов $ExpNmb = 10^4$. Затем последовательно проводятся расчеты значения числа для числа экспериментов $ExpNmb = 10^5, 10^6, 10^7, 10^8$ и занести полученные результаты в вектор SERIA_1. Провести еще четыре серии расчетов для числа экспериментов $ExpNmb = 10^4, 10^5, 10^6, 10^7, 10^8$ и занести результаты в вектора SERIA_2, SERIA_3, SERIA_4 и SERIA_5.

```
EXP_NUMBS = [10 ** 4, 10 ** 5, 10 ** 6, 10 ** 7, 10 ** 8]
SER_NAMES = ["SERIA_1", "SERIA_2", "SERIA_3", "SERIA_4",
«SERIA_5»]
```

```
def calculation():
    ser = pd.Series(dtype=float64, index=EXP_NUMBS)
    for exp_nmb in EXP_NUMBS:
        pi = calc_pi(0, 0, 1, exp_nmb)
        ser.at[exp_nmb] = pi
    # display(ser)
    return ser
```

```
def series_of_calculation():
    df = pd.DataFrame(index=EXP_NUMBS)
    for ser_name in SER_NAMES:
        ser = calculation()
        df[ser_name] = ser
    return df
```

```
def save_result():
    df = series_of_calculation()
    display(df)
    df.to_csv(RESULT_PATH, mode="w+") # overwrite file mode
```

```
save_result()
```

Задание 3.

Рассчитать погрешность вычислений значений числа для каждой серии экспериментов. Затем получить средний результат по 5-ти сериям для каждого из соответствующего числа экспериментов и рассчитать погрешность вычислений для усредненных значений.

```
def calculate_std(path, ser_names):  
    df = pd.read_csv(path)  
  
    stds = pd.Series(dtype=float64)  
    for seria_name, seria_data in df[ser_names].iteritems():  
        stds.at[seria_name] = seria_data.std()  
    print("Погрешность вычислений значений числа для каждой  
серии экспериментов")  
    display(stds)  
  
    means = pd.Series(dtype=float64)  
    for seria_name, seria_data in df[ser_names].iteritems():  
        means.at[seria_name] = seria_data.mean()  
    print("Средний результат для каждой серии:")  
    display(means)  
    print("Погрешность для усредненных значений:",  
means.std())
```

```
calculate_std(RESULT_PATH, SER_NAMES)
```

Таблица 1. Погрешность вычислений для каждого результата

	$N = 10^4$	$N = 10^5$	$N = 10^6$	$N = 10^7$	$N = 10^8$
SERIA_1	3.128000	3.139960	3.141256	3.142857	3.141825
SERIA_2	3.149200	3.140400	3.141260	3.141615	3.141498
SERIA_3	3.112800	3.144320	3.141328	3.141892	3.141624
SERIA_4	3.164800	3.141880	3.140844	3.141280	3.141678
SERIA_5	3.141200	3.141040	3.140708	3.141479	3.141687

Таблица 2. Средние значения для каждой серии

	SERIA_1	SERIA_2	SERIA_3	SERIA_4	SERIA_5
С р е д н е е значение S_i	0.006116	0.003612	0.013242	0.010463	0.000381

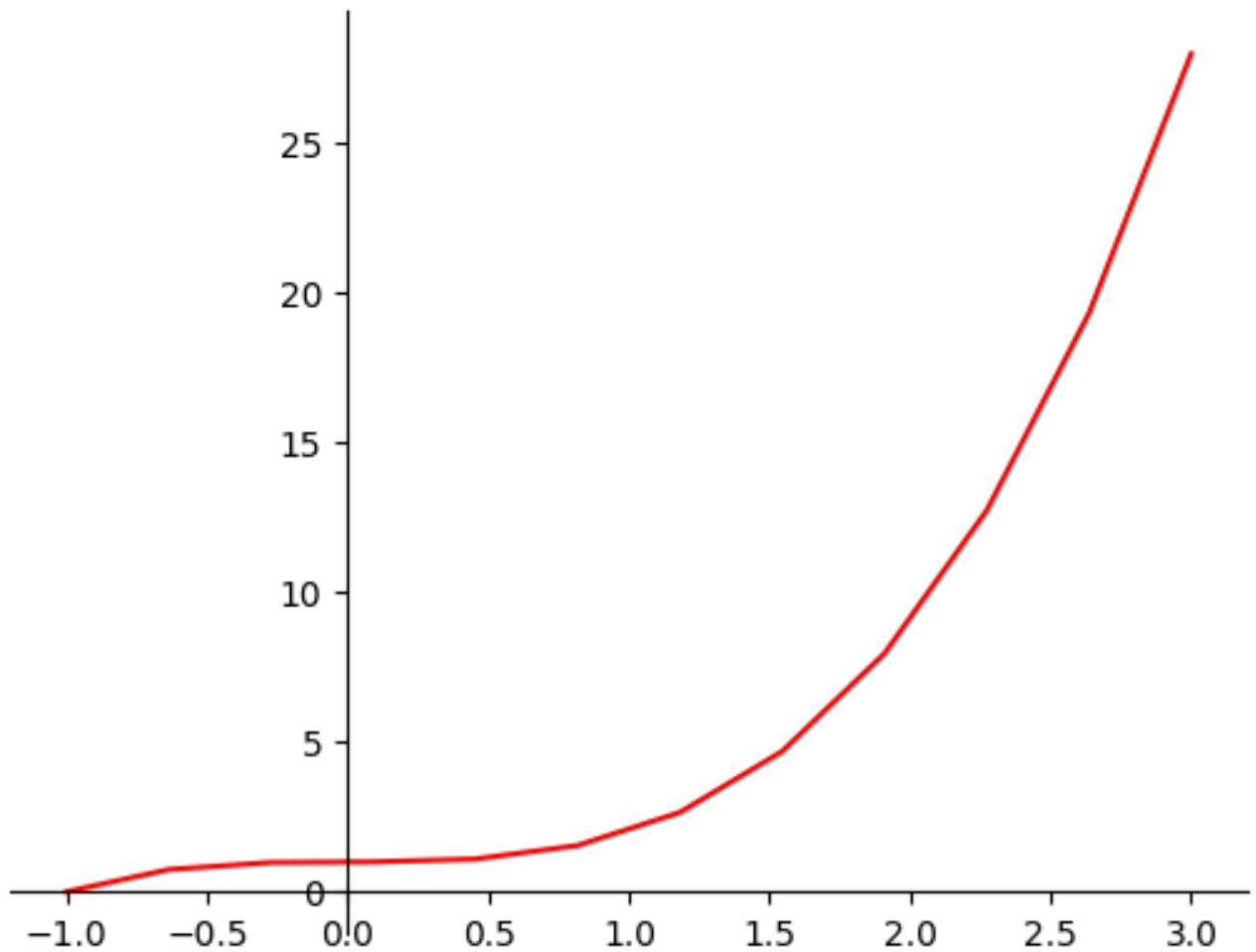
Таблица 3. Погрешность для средних значений

	SERIA_1	SERIA_2	SERIA_3	SERIA_4	SERIA_5
Погрешность Eps_{S_i} $Eps_{S_i} = \left \frac{S_i + \pi}{\pi} \right $	3.13877 9	3.14279 5	3.13639 3	3.14609 7	3.14122 3

Погрешность для усредненных значений: 0.0037210517582362823

Задание 4.

Используя описанную методику расчета решить задачу нахождения значения определенного интеграла функции $f(x) = x^3 + 1$ на интервале от 0 до 2. Провести 3 серии расчетов для числа экспериментов $ExpNmb = 10^4, 10^5, 10^6, 10^7$. Определить погрешность вычислений для усредненных по сериям значений.



```
TASK_4_EXP_NUMBS = [10 ** 4, 10 ** 5, 10 ** 6, 10 ** 7]
TASK_4_SER_NAMES = ["SERIA_1", "SERIA_2", "SERIA_3",
                    "SERIA_4"]
FUNCTION = lambda x: x ** 3 + 1

TASK_4_RESULT_PATH = "task 4 result.csv"
def task_4_draw_function(func):
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(-1,3,12)
```

```
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.spines['left'].set_position('zero')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
```

```
# plot the function
plt.plot(x, func(x), 'r')
```

```
# show the plot
plt.show()
```

```
task_4_draw_function(FUNCTION)
```

```
X_MIN = 0
X_MAX = 2
Y_MIN = 0
Y_MAX = 9
```

```
def task_4_calc_function(func, exp_nmb):
```

```
    m = 0
    s_par = abs((Y_MAX - Y_MIN) * (X_MAX - X_MIN))
    for i in range(exp_nmb):
        xp = random.uniform(X_MIN, X_MAX)
        yp = random.uniform(Y_MIN, Y_MAX)
        if func(xp) > yp:
            m += 1
    s = s_par * m / exp_nmb
    return s
```

```
def task_4_calculation():
    ser = pd.Series(dtype=float64, index=TASK_4_EXP_NUMBS)
```



```

for exp_nmb in TASK_4_EXP_NUMBS:
    s = task_4_calc_function(FUNCTION, exp_nmb)
    ser.at[exp_nmb] = s
display(ser)
return ser

```

```

def task_4_series_of_calculation():
    df = pd.DataFrame(index=TASK_4_EXP_NUMBS)
    for ser_name in TASK_4_SER_NAMES:
        ser = task_4_calculation()
        df[ser_name] = ser
    return df

```

```

def task_4_save_result():
    df = task_4_series_of_calculation()
    df.to_csv(TASK_4_RESULT_PATH, mode="w+")

```

```
task_4_save_result()
```

Таблица 1. Погрешность вычислений для каждого результата

	$N = 10^4$	$N = 10^5$	$N = 10^6$	$N = 10^7$
SERIA_1	6.080400	5.995620	5.991966	5.999263
SERIA_2	6.026400	6.006960	5.991228	5.999742
SERIA_3	6.046200	5.983020	6.016320	6.000268
SERIA_4	6.015600	5.949180	5.997402	5.998792

Таблица 2. Средние значения для каждой серии

	SERIA_1	SERIA_2	SERIA_3	SERIA_4
Среднее значение S_i	0.042496	0.014994	0.026861	0.028598

Таблица 3. Погрешность для средних значений

	SERIA_1	SERIA_2	SERIA_3	SERIA_4
Погрешность Eps_{S_i} $F(X) = \int_0^2 x^3 + 1$ $Eps_{S_i} = \left \frac{S_i + F(x)}{F(x)} \right $	6.016812	6.006083	6.011452	5.990243

Вывод

Существует много способов вычисления числа Π . Самым простым и понятным является численный метод Монте-Карло, суть которого сводится к простейшему перебору точек на площади.