

Задание 4.

$$P(Sn \in [n/2 - \sqrt{npq}, n/2 + \sqrt{npq}])$$

n: 100, p: 0.001	P: 9.596841476810663e-122	P from CPT: 0.0
n: 100, p: 0.01	P: 6.103987557172999e-72	P from CPT: 0.0
n: 100, p: 0.1	P: 3.6123083280873955e-21	P from CPT: 2.220446049250313e-16
n: 100, p: 0.25	P: 4.2504440254864535e-06	P from CPT: 6.180993840088078e-07
n: 100, p: 0.5	P: 0.7287469759261653	P from CPT: 0.6826894921370862
n: 1000, p: 0.001	P: 0.0	P from CPT: 0.0
n: 1000, p: 0.01	P: 0.0	P from CPT: -1.1102230246251565e-16
n: 1000, p: 0.1	P: 0.0	P from CPT: -1.1102230246251565e-16
n: 1000, p: 0.25	P: 1.4970596601027823e-58	P from CPT: 0.0
n: 1000, p: 0.5	P: 0.6730633175684045	P from CPT: 0.6572182888520888
n: 10000, p: 0.001	P: OVERFLOW	P from CPT: 0.0
n: 10000, p: 0.01	P: OVERFLOW	P from CPT: 0.0
n: 10000, p: 0.1	P: OVERFLOW	P from CPT: -1.1102230246251565e-16
n: 10000, p: 0.25	P: OVERFLOW	P from CPT: 1.1102230246251565e-16
n: 10000, p: 0.5	P: OVERFLOW	P from CPT: 0.6826894921370862

$$P(Sn \leq 5)$$

n: 100, p: 0.001	P: 0.9999999989001893	P from CPT: 0.6241452284609645
n: 100, p: 0.01	P: 0.9994654655360061	P from CPT: 0.8425315989385818
n: 100, p: 0.1	P: 0.057576886487033956	P from CPT: 0.04736129193961791
n: 100, p: 0.25	P: 1.1700149154089138e-07	P from CPT: 1.9259262001858346e-06
n: 100, p: 0.5	P: 6.26162256269268e-23	P from CPT: 1.1102230246251565e-16
n: 1000, p: 0.001	P: 0.9994119298982362	P from CPT: 0.8414343875037021
n: 1000, p: 0.01	P: 0.0661395116072514	P from CPT: 0.05527748145542166
n: 1000, p: 0.1	P: 2.556545693060053e-38	P from CPT: 0.0
n: 1000, p: 0.25	P: 3.9691404522735886e-115	P from CPT: 1.1102230246251565e-16
n: 1000, p: 0.5	P: 7.738505306294352e-289	P from CPT: -1.1102230246251565e-16
n: 10000, p: 0.001	P: 0.06699137339752666	P from CPT: 0.05605431814512302
n: 10000, p: 0.01	P: 2.070048551692478e-36	P from CPT: 0.0
n: 10000, p: 0.1	P: 0.0	P from CPT: -2.220446049250313e-16
n: 10000, p: 0.25	P: 0.0	P from CPT: 1.1102230246251565e-16
n: 10000, p: 0.5	P: 0.0	P from CPT: 1.1102230246251565e-16

$$P(Sn = k)$$

n: 100, p: 0.001	max P: 0.9047921471137089	max P from CPT: 1.2005793277351
n: 100, p: 0.01	max P: 0.36972963764972644	max P from CPT: 0.4009520779091839
n: 100, p: 0.1	max P: 0.13186534682448858	max P from CPT: 0.1329807601338109
n: 100, p: 0.25	max P: 0.09179969176683679	max P from CPT: 0.09213177319235613
n: 100, p: 0.5	max P: 0.07958923738717877	max P from CPT: 0.07978845608028654
n: 1000, p: 0.001	max P: 0.3680634882592229	max P from CPT: 0.3991419012697672
n: 1000, p: 0.01	max P: 0.12574021112620629	max P from CPT: 0.12679217987703037
n: 1000, p: 0.1	max P: 0.0420167908610866	max P from CPT: 0.042052208700336005
n: 1000, p: 0.25	max P: 0.029124105883705082	max P from CPT: 0.029134624815788775
n: 1000, p: 0.5	max P: 0.0252250181783608	max P from CPT: 0.025231325220201602
n: 10000, p: 0.001	P: OVERFLOW	max P from CPT: 0.12621975176225178
n: 10000, p: 0.01	P: OVERFLOW	max P from CPT: 0.040095207790918394
n: 10000, p: 0.1	P: OVERFLOW	max P from CPT: 0.01329807601338109
n: 10000, p: 0.25	P: OVERFLOW	max P from CPT: 0.009213177319235613
n: 10000, p: 0.5	P: OVERFLOW	max P from CPT: 0.007978845608028654

Вывод: [код по ссылке](#)

Приближения вычислялись с помощью локальной центральной предельной теоремы (ее частного случая), точные – по формуле Бернулли для серии испытаний. При $n: 10000, p: 0.001$ в двух из трех экспериментов возникает переполнение при подсчете точного значения, Питон не может такое большое число хранить в float.

Судя по таблице с данными можно сделать вывод, что при больших значениях n разница между точным и приближенным значениями уменьшается, особенно это заметно в первом пункте. Это и логично, так как ЦПТ подразумевает, что берется предел по n . В третьем пункте также видно что с ростом n , разница между значениями уменьшается.

Также можно заметить серьезное преимущество приближенных вычислений: там, где точные значения уже не помещаются в память, ЦПТ не переполняется и выдает результат.