

Расчёт доверительного интервала через бета-распределение

#гипотезы, #доверительный-интервал, #статистика

В Excel и google docs есть встроенная функция бета-распределения. Формула имеет вот такой вид:

$\text{Beta.inv}(\text{границы_точности}; \text{КОНВЕРСИЙ}; \text{ВЫБОРКА} - (\text{КОНВЕРСИЯ} \times \text{ВЫБОРКУ}))$

Если мы вернёмся к задачке с баннеров, то у нас получатся вот такие формулы

Для Баннера А

$\text{CTR}(\text{min}) = \text{beta.inv}(0.20 / 2; 5\% \times 10000; 10000 - 5\% \times 10000)$ $\text{CTR}(\text{max}) = \text{beta.inv}(1 - 0.20 / 2; 5\% \times 10000; 10000 - 5\% \times 10000)$

Для Баннера Б

$\text{CTR}(\text{min}) = \text{beta.inv}(0.1; 5.5\% \times 10000; 10000 - 5.5\% \times 10000)$ $\text{CTR}(\text{max}) = \text{beta.inv}(0.9; 5.5\% \times 10000; 10000 - 5.5\% \times 10000)$

Получается:

Для баннера А

$\text{CTR}(\text{min}) = 4.72\%$ $\text{CTR}(\text{max}) = 5.28\%$

Для баннера Б

$\text{CTR}(\text{min}) = 5.21\%$ $\text{CTR}(\text{max}) = 5.79\%$

Результаты расчёта доверительного интервала через **стандартную формулу** и через формула бета-распределения одинаковые. В случае биномиального расчёта можно использовать любую из этих формул.

Аналогично можно сделать и через Python. Разберём на примере баннера Б:

python

```
import scipy.stats as ss

dist = ss.beta(550, 9450)

ctrmin = dist.ppf(0.1)
ctrmax = dist.ppf(0.9)
```

```
print("CTRmin: {:.2%} \nCTRmax: {:.2%}".format(ctrmin, ctrmax))
```

Вывод кода выше:

CTRmin: 5.21%

CTRmax: 5.79%

#доработать-заметку-или-удалить Добавить про бета распределение:
<https://habr.com/ru/post/172101/>