Лабораторная работа №3
Выполнил: Кузнецов Павел М3207

Задача 1.2 Предъявите доверительный интервал уровня $1-\alpha$ для указанного параметра при данных предположениях (с обоснованиями). Сгенерируйте 2 выборки объёма объёма 25 и посчитайте доверительный интервал. Повторить 1000 раз. Посчитайте, сколько раз 95-процентный доверительный интервал покрывает реальное значение параметра. То же самое сделайте для объема выборки 10000. Как изменился результат? Как объяснить?

Задача представлена в 3 вариантах. Везде даны две независимые выборки X, Y из нормальных распределений $N(\mu_1, \sigma_1^2), N(\mu_2, \sigma_2^2)$ объема n, m соответственно. Сначала указывается оцениваемая функция, потом данные об остальных параметрах, затем параметры эксперимента и подсказки.

$\tau = \mu_1 - \mu_2; \sigma_1^2 = \sigma_2^2$ неизвестна; $\mu_1 = 2, \mu_2 = 1, \sigma_1^2 = \sigma_2^2 = 1$; воспользуйтесь функцией

$$\sqrt{\frac{mn(m+n-2)}{m+n}}\frac{\bar{X}-\bar{Y}-\tau}{\sqrt{nVar(X)+mVar(Y)}}$$

где Var(.) - выборочная смещенная дисперсия. Смотрите в сторону распределения Стьюдента.

Решение: Из функции ... приходим путём преобразований:

$$\sqrt{\frac{mn(m+n-2)}{m+n}}\frac{\bar{X}-\bar{Y}-\tau}{\sqrt{nVar(X)+mVar(Y)}} \sim N(0,1)\sqrt{\frac{n+m-2}{nVar(X)+mVar(Y)}}$$

Получившаяся функция имеет распределение Стьюдента с степенью свободы n+m-2

Построим доверительный интервал уровня $1 - \alpha$:

$$P(\bar{X} - \bar{Y} - \frac{q\sqrt{(nVar(X) + mVar(Y))(m+n)}}{\sqrt{mn(m+n-2)}} \leq \tau$$

$$\leq \overline{X} - \overline{Y} + \frac{q\sqrt{(nVar(X) + mVar(Y))(m+n)}}{\sqrt{mn(m+n-2)}}) = 1 - \alpha$$

где q - квантиль распределения Стьюдента для установленных степени свободы и уровня.

Эксперимент:

```
import numpy
import scipy.stats
nu_1=2
nu_2=1
disp=1
n1 = 25
n2 = 10000
k = 1000
counter=0

q = scipy.stats.t.ppf(q = 0.975, df = 2 * n1 - 2)
precalculated_half = q * numpy.sqrt(2.0/(n1*(2*n1-2)))
for i in range (k):
    X_sample = numpy.random.normal(size=n1, loc = nu_1, scale=disp)
    Y_sample = numpy.random.normal(size=n1, loc= nu_2, scale=disp)
    lower_limit = numpy.mean(X_sample) - numpy.mean(Y_sample) - q * precalculated_half*numpy.sqrt(n1 * (numpy.var(X_sample, ddof=1) + numpy.var(Y_sample, ddof=1)))
    higher_limit = numpy.mean(X_sample) - numpy.mean(Y_sample) + q * precalculated_half*numpy.sqrt(n1 * (numpy.var(X_sample, ddof=1) + numpy.var(Y_sample, ddof=1)))
    if(lower_limit < (nu_1 - nu_2) and (nu_1 - nu_2) < higher_limit):
        counter+=1
print(counter)

q = scipy.stats.t.ppf(q = 0.975, df = 2 * n2 - 2)
precalculated_half = q * numpy.sqrt(2.0/(n2*(2*n2-2)))
for i in range (k):
    X_sample = numpy.random.normal(size=n2, loc = nu_1)
    Y_sample = numpy.random.normal(size=n2, loc= nu_2)
    lower_limit = numpy.mean(X_sample) - numpy.mean(Y_sample) - q * precalculated_half*numpy.sqrt(n2 * (numpy.var(X_sample) + numpy.var(Y_sample)))
    higher_limit = numpy.mean(X_sample) - numpy.mean(Y_sample) + q * precalculated_half*numpy.sqrt(n2 * (numpy.var(X_sample) + numpy.var(Y_sample)))
```

```
    if(lower_limit < (nu_1 - nu_2) and (nu_1 - nu_2) < higher_limit
    ):
        counter+=1
print(counter)

```

Получившиеся результаты:

Частота попадания реального значения в построенный доверительный интервал для выборки объема 25: 965

Частота попадания реального значения в построенный доверительный интервал для выборки объема 10000: 973

Частота немного выросла, что связывается с увеличением n, но в целом вероятность покрытия доверительного интервала остаётся близкой к постоянной, и слабо зависит от величины выборки.

Задача 2.5 Постройте асимптотический доверительный интервал уровня $1-\alpha$ для указанного параметра. Проведите эксперимент по схеме, аналогичной первой задаче.

Класс распределений: $Geom(p)$, оценить параметр p. Эксперимент для p = 0.7;

$$E(X) = \frac{1}{p} => \widehat{p} = \frac{1}{\bar{X}}$$

$$\sigma(\bar{X}) = \sqrt{\frac{p(1-p)}{n}}.$$

Тогда асимптотический доверительный интервал для параметра p в $Geom(p)$ с уровнем доверия $1-\alpha$:

$$P(\widehat{p} - z_{\alpha/2} \cdot \sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}} \leq p \leq \widehat{p} + z_{\alpha/2} \cdot \sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}})$$

Эксперимент:

```python
import numpy
import scipy.stats
p = 0.7
n1 = 25
n2 = 10000
k = 1000
counter=0

q = scipy.stats.norm.ppf(q = 0.95)
for i in range (k):
    X_sample = scipy.stats.geom.rvs(p, size=n1)
    p_score = 1.0/numpy.mean(X_sample)
    lower_limit = p_score - q * numpy.sqrt(p_score * (1 - p_score)/n1)
    higher_limit = p_score + q * numpy.sqrt(p_score * (1 - p_score)/n1)
    if(lower_limit < p and p < higher_limit):
        counter+=1
print(counter)

for i in range (k):
    X_sample = scipy.stats.geom.rvs(p, size=n2)
    p_score = 1.0/numpy.mean(X_sample)
    lower_limit = p_score - q * numpy.sqrt(p_score * (1 - p_score)/n2)
    higher_limit = p_score + q * numpy.sqrt(p_score * (1 - p_score)/n2)
    if(lower_limit < p and p < higher_limit):
        counter+=1
print(counter)
```

Получившиеся результаты: Частота попадания реального значения в построенный доверительный интервал для выборки объема 25: 952

Частота попадания реального значения в построенный доверительный интервал для выборки объема 10000: 979