

## 9\_3

May 13, 2016

```
In [154]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D
```

```
In [155]: N = 100
sample = stats.norm.rvs(size=N)
gamma = 0.95
```

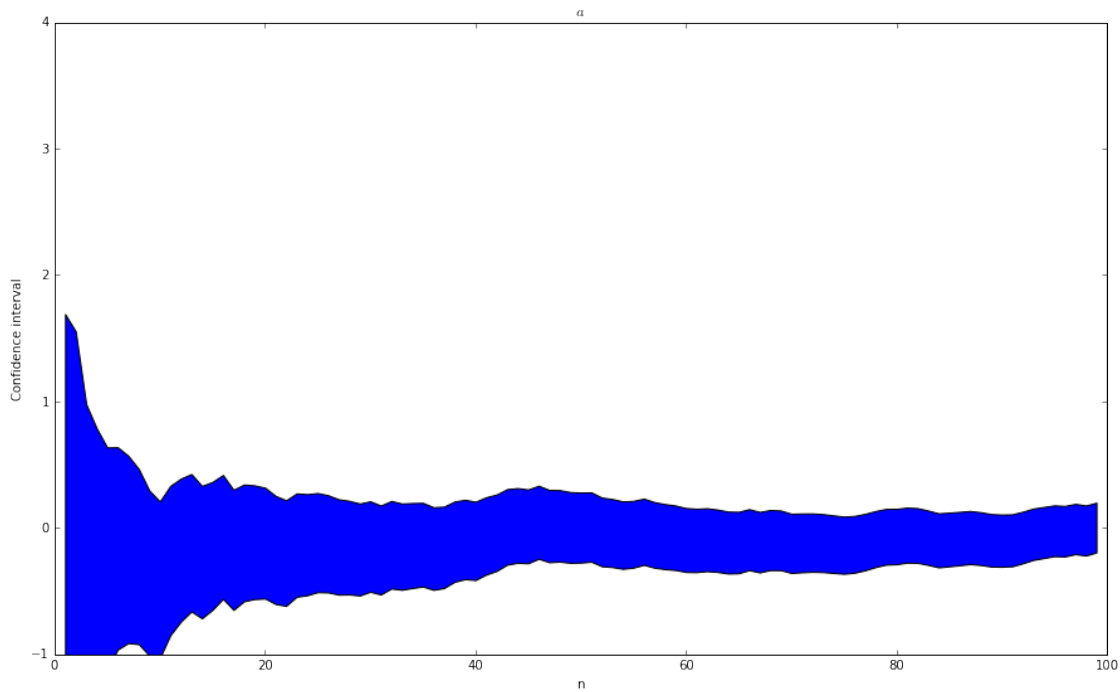
### 0.1 Точный доверительный интервал для $a$ при известном $\sigma^2$

Обозначим  $Z_\gamma$  квантиль уровня  $\gamma$  нормального стандартного распределения. Доверительным интервалом для  $a$  будет  $(\bar{X} - Z_\alpha \frac{\sigma}{\sqrt{n}}, \bar{X} + Z_\alpha \frac{\sigma}{\sqrt{n}})$ , где  $\alpha = \frac{1+\gamma}{2}$

```
In [156]: z = stats.norm.ppf((1+gamma)/2)
def get_interval_a_with_sigma_lower_bound(sample):
    return np.average(sample) - z*(1.0/len(sample))*0.5
def get_interval_a_with_sigma_upper_bound(sample):
    return np.average(sample) + z*(1.0/len(sample))*0.5

In [157]: def draw_plot(get_interval_lb, get_interval_ub, title=''):
    plt.figure(figsize=(15, 9))
    plt.title(title)
    plt.ylabel('Confidence interval')
    plt.xlabel('n')
    plt.ylim(-1, 4)
    plt.fill_between(range(1, len(sample)),
                     [get_interval_lb(sample[:n]) for n in range(1, N)],
                     [get_interval_ub(sample[:n]) for n in range(1, N)])
    plt.show()
```

```
In [158]: draw_plot(get_interval_a_with_sigma_lower_bound, get_interval_a_with_sigma_upper_bound, r'$a$')
```



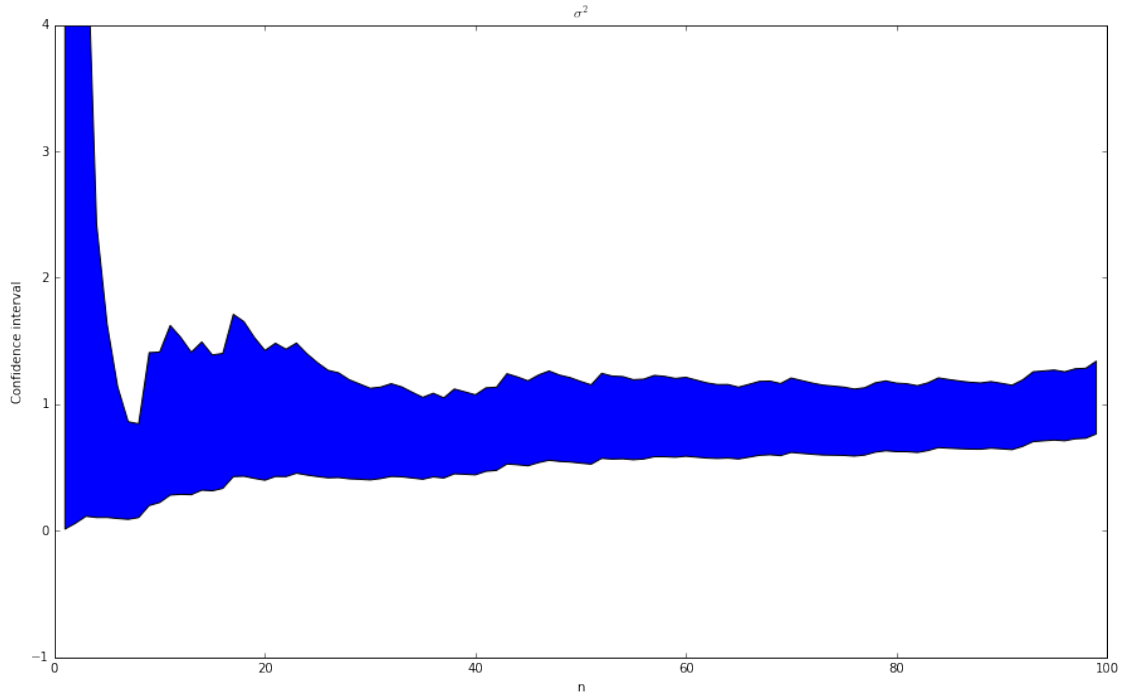
## 0.2 Точный доверительный интервал для $\sigma^2$ при известном $a$

Как известно, в этом случае доверительный интервал будет  $(\frac{n(\overline{X}-a)^2}{z_\alpha^2}, (\frac{n(\overline{X}-a)^2}{z_\beta^2}))$ , где  $\alpha = \frac{1+\gamma}{2}$ ,  $\beta = \frac{1-\gamma}{2}$ ,  $z_\epsilon$  — квантиль хи-квадрат распределения с  $n$  степенями свободы.

```
In [159]: def get_interval_sigma_with_a_lower_bound(sample):
            alpha = (1+gamma)/2
            return (np.sum((sample)**2))/stats.chi2.ppf(alpha, df = len(sample))

            def get_interval_sigma_with_a_upper_bound(sample):
                beta = (1-gamma)/2
                return (np.sum((sample)**2))/stats.chi2.ppf(beta, df = len(sample))
```

```
In [160]: draw_plot(get_interval_sigma_with_a_lower_bound, get_interval_sigma_with_a_upper_bound, r'\sigma')
```



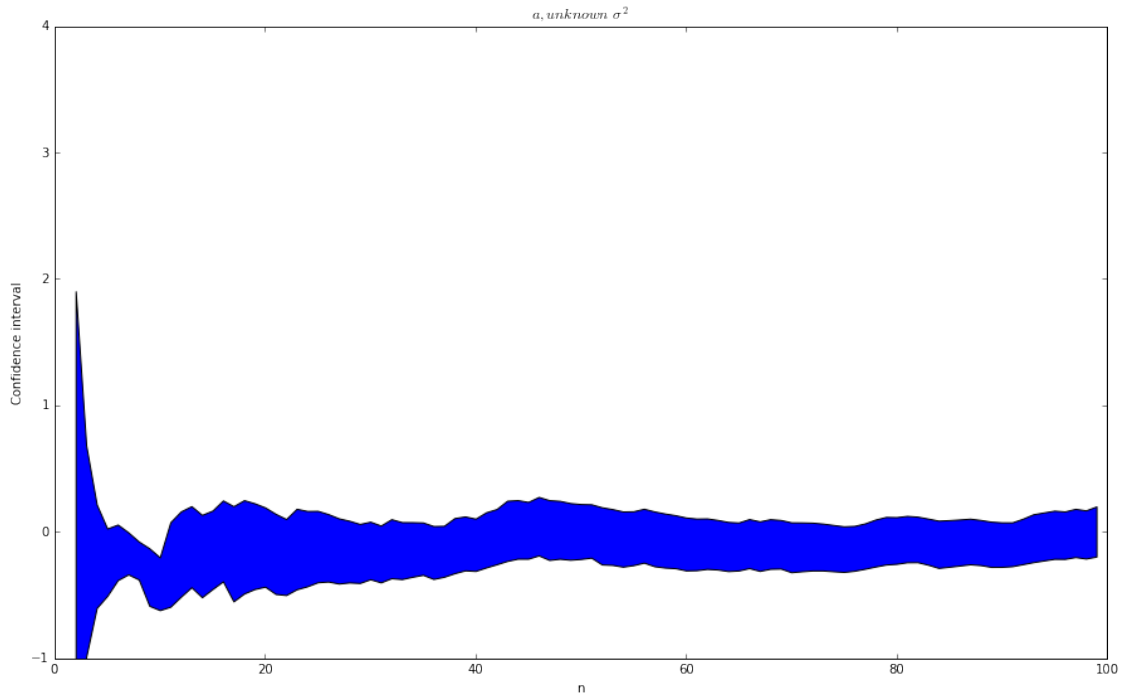
### 0.3 Точный доверительный интервал для $a$ при неизвестном $\sigma^2$

Точный доверительный интервал равен  $(\bar{X} - Z_\alpha \frac{S(X)}{\sqrt{n}}; \bar{X} + Z_\alpha \frac{S(X)}{\sqrt{n}})$ , где  $Z_\alpha$  — квантиль уровня  $\alpha$  распределение Стьюдента, причем  $\alpha = \frac{1+\gamma}{2}$

```
In [161]: alpha = (1+gamma)/2
def get_interval_a_with_unknown_sigma_lower_bound(sample):
    n = len(sample)
    Z = stats.t.ppf(alpha, df = n-1)
    S = np.average(sample**2) - (np.average(sample))**2
    return np.average(sample) - Z * S / np.sqrt(n)

def get_interval_a_with_unknown_sigma_upper_bound(sample):
    n = len(sample)
    Z = stats.t.ppf(alpha, df = n-1)
    S = np.average(sample**2) - (np.average(sample))**2
    return np.average(sample) + Z * S / np.sqrt(n)
```

```
In [162]: draw_plot(get_interval_a_with_unknown_sigma_lower_bound, get_interval_a_with_unknown_sigma_upper_bound)
```



#### 0.4 Точный доверительный интервал для $\sigma^2$ при неизвестном $a$

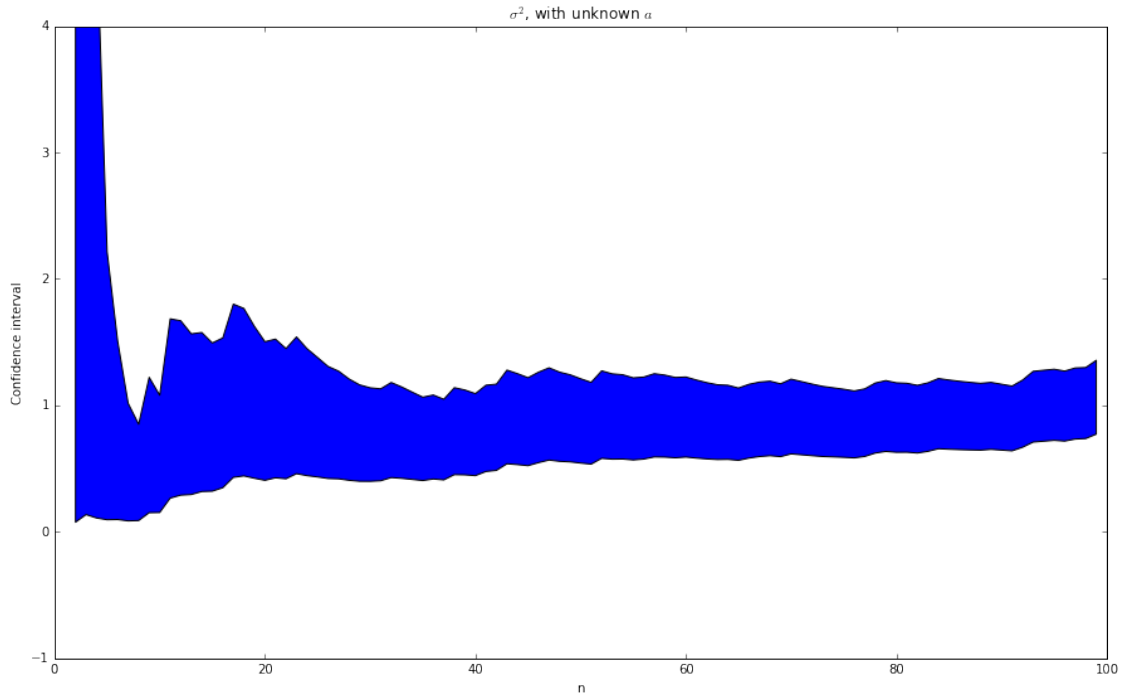
Обозначим квантиль уровня  $\epsilon$  распределения хи-квадрат  $(n - 1)$  степени свободы через  $Z_\epsilon$ . Доверительный интервал равняется  $(\frac{(n-1)S^2(X)}{Z_\alpha}, \frac{(n-1)S^2(X)}{Z_\beta})$ ,  $\beta = \frac{1-\gamma}{2}$ ,  $\alpha = \frac{1+\gamma}{2}$

```
In [163]: def S(X):
            return np.average(X**2) - (np.average(X))**2

def get_interval_sigma_unknown_a_lower_bound(sample):
    n = len(sample)
    s = S(sample)
    z = stats.chi2.ppf((1+gamma) / 2, df = n-1)
    return (n * s / z)

def get_interval_sigma_unknown_a_upper_bound(sample):
    n = len(sample)
    s = S(sample)
    z = stats.chi2.ppf((1-gamma) / 2, df = n-1)
    return (n * s / z)
```

```
In [176]: draw_plot(get_interval_sigma_unknown_a_lower_bound, get_interval_sigma_unknown_a_upper_bound,
```



## 0.5 Доверительная область для $(a; \sigma^2)$

Построить точную доверительную область не получилось.

```
In [183]: # fig = plt.figure()
# ax = fig.gca(projection='3d')

# grid = np.array([i for i in range(1, N)])
# a_s = [get_interval_a_with_unknown_sigma_lower_bound(sample[:n]) for n in range(1, N)]
# sigmas = [get_interval_sigma_unknown_a_lower_bound(sample[:n]) for n in range(1, N)]

# ax.contourf(grid[1:], a_s[1:], sigmas[1:])
# plt.show()
```

**Вывод.** Линейная регрессия помогла найти квантили для доверительных интервалов оценок параметром  $N(a, \sigma^2)$ . С ростом  $n$  вне зависимости от метода доверительный интервал сосредоточен вокруг  $a = 0$  и  $\sigma^2 = 1$ . Причем при  $n \geq 20$  доверительные интервалы для соответствующих оценок с известным или неизвестным вторым параметром практически не отличаются.