

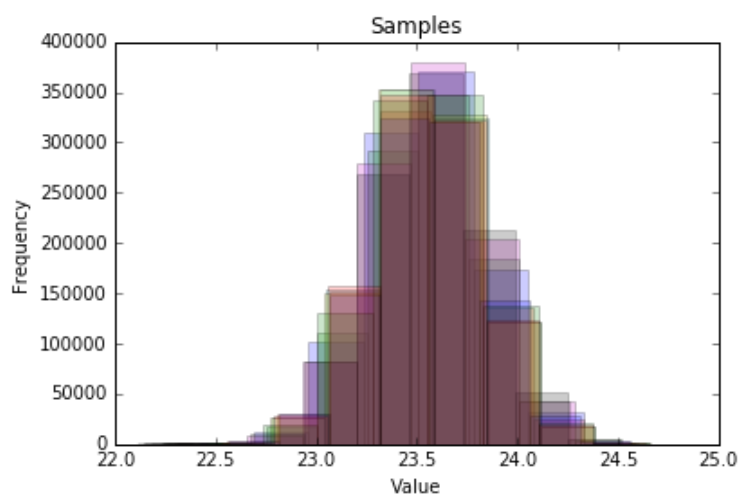
1.  $N(23.56, 0.27)$

```
In [9]: import numpy as np
import scipy.stats as sps
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [62]: a = 23.56
sigma = 0.27

N = 10 ** 6
num_of_samples = 10

samples = [sps.norm.rvs(loc=a, scale=sigma, size=N) for i in range(num_of_samples)]
for i in range(num_of_samples):
    plt.hist(samples[i], alpha = 1 / num_of_samples * 2)
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.title("Samples")
fig = plt.gcf()
```

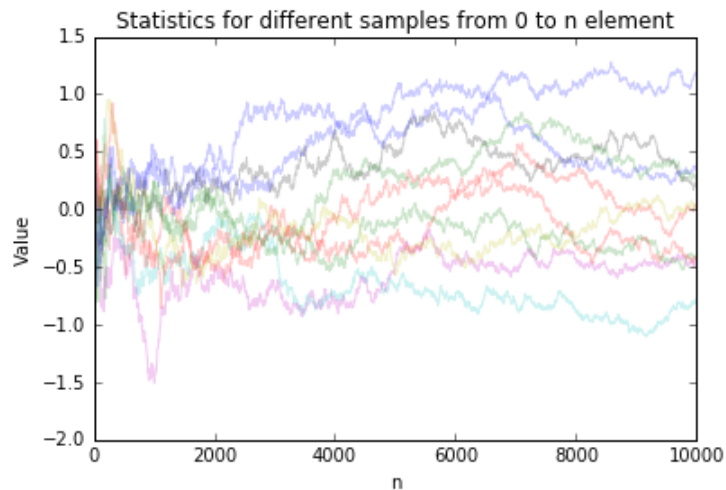


```
In [105]: def stat_CLT(X):
return (len(X) / sigma)**(1/2) * (np.mean(X) - a)
```

```
In [106]: Y = np.arange(0, N // 100, 5)
for i in range(num_of_samples):
    stat = [stat_CLT(samples[i][:j]) for j in Y]
    plt.plot(Y, stat, alpha = 1 / num_of_samples * 2)
plt.xlabel("n")
plt.ylabel("Value")
plt.title("Statistics for different samples from 0 to n element")
fig = plt.gcf()

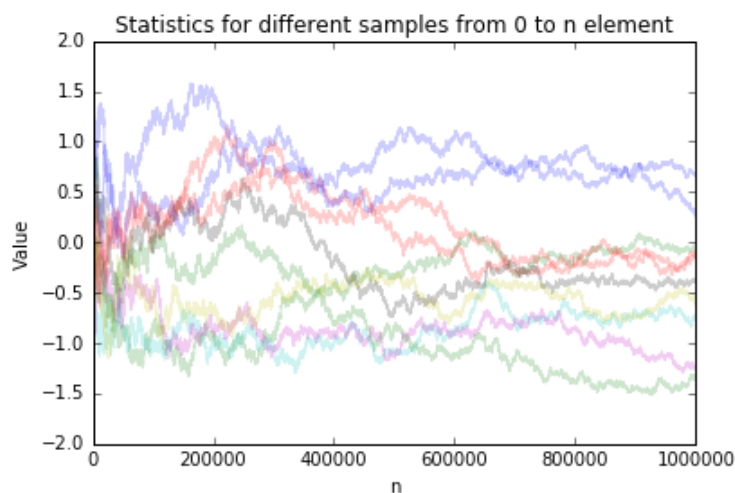
/home/riv/anaconda3/lib/python3.5/site-packages/numpy/core/_methods.p
y:59: RuntimeWarning:
```

Mean of empty slice.



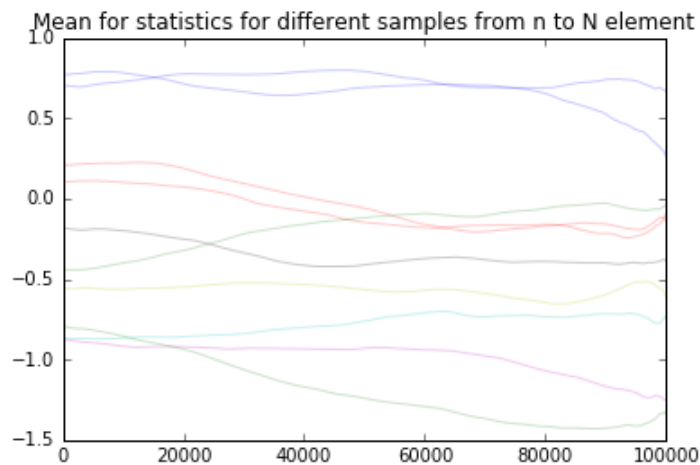
```
In [118]: CLT_stat = []
Y = np.arange(0, N , 10)
for i in range(num_of_samples):
    stat = [stat_CLT(samples[i][:j]) for j in Y]
    CLT_stat.append(stat)
    plt.plot(Y, CLT_stat[i], alpha = 1 / num_of_samples * 2)
plt.xlabel("n")
plt.ylabel("Value")
plt.title("Statistics for different samples from 0 to n element")
fig = plt.gcf()
#Результат статистики от n/10 до n/1000000
/home/riv/anaconda3/lib/python3.5/site-packages/numpy/core/_methods.p
y:59: RuntimeWarning:
```

Mean of empty slice.



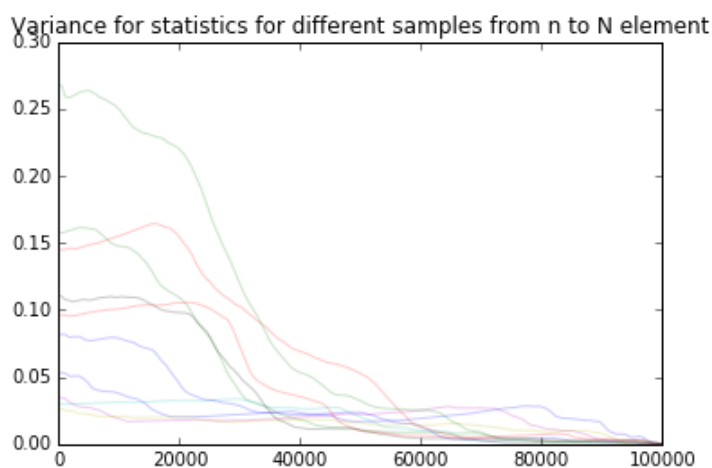
```
In [120]: Y = np.arange(0, N - 101, 100)
for i in range(num_of_samples):
    stat = [np.mean(CLT_stat[i][j:]) for j in Y]
    plt.plot(Y, stat, alpha = 1 / num_of_samples * 2)
plt.title("Mean for statistics for different samples from n to N element")
plt.show()
# ПОПЕРЕЧНОЕ МНОЖЕСТВО СРЕДНЕЙ СТАТИСТИКИ (от n до N)
/home/riv/anaconda3/lib/python3.5/site-packages/numpy/core/_methods.p
y:59: RuntimeWarning:
```

Mean of empty slice.

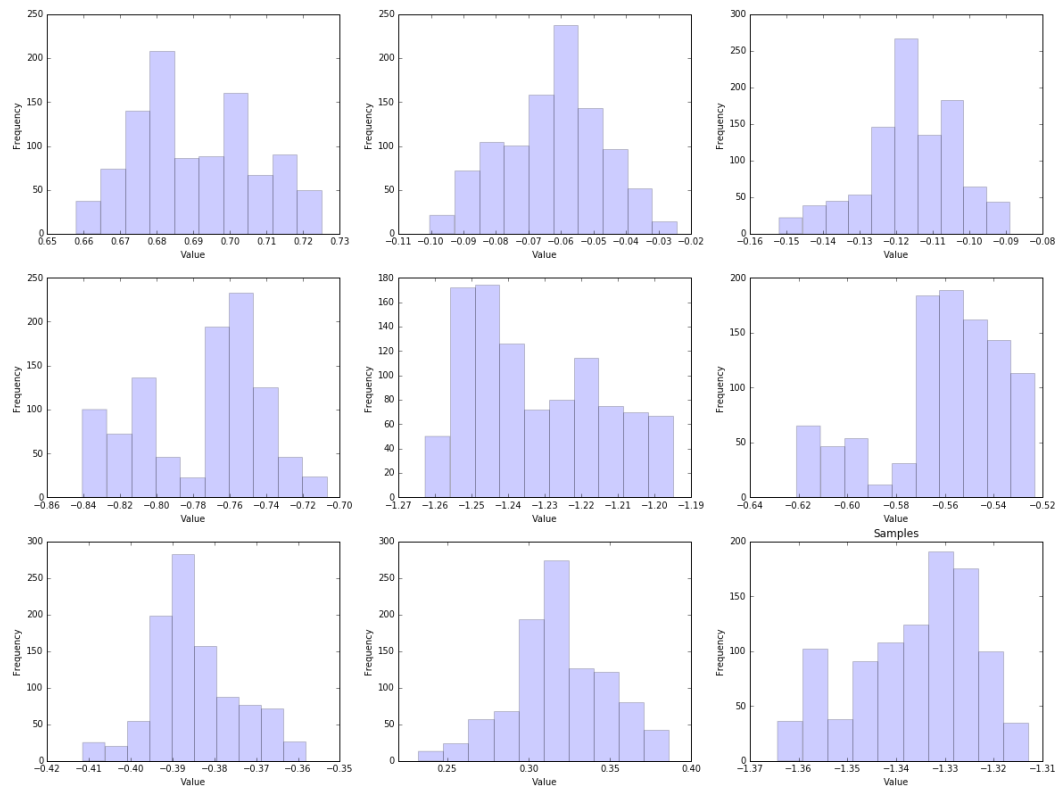


```
In [121]: Y = np.arange(0, N - 1000, 100)
for i in range(num_of_samples):
    stat = [np.var(CLT_stat[i][j:]) for j in Y]
    plt.plot(Y, stat, alpha = 1 / num_of_samples * 2)
plt.title("Variance for statistics for different samples from n to N element")
plt.show()
# ПОПЕРЕЧНОЕ МНОЖЕСТВО СРЕДНЕЙ СТАТИСТИКИ (от n до N)
/home/riv/anaconda3/lib/python3.5/site-packages/numpy/core/_methods.p
y:82: RuntimeWarning:
```

Degrees of freedom <= 0 for slice



100000



Статистика  $T_n$  ведёт себя в большинстве случаев близко к нормальному распределению с мат.ожиданием  $\sim 0$  и выборочной дисперсией близкой к нулю, откуда следует, что ЦПТ применима, но лишь с некоторыми ограничениями для произвольной выборки данного размера(например, нельзя строго утверждать о классификации данных статистик).