

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

N = 10**4
sample = np.random.normal(size=N)
x_row = np.linspace(-5, 5, 1000)
```

Подсчет эмпирической функции распределения для части выборки sample размера n.

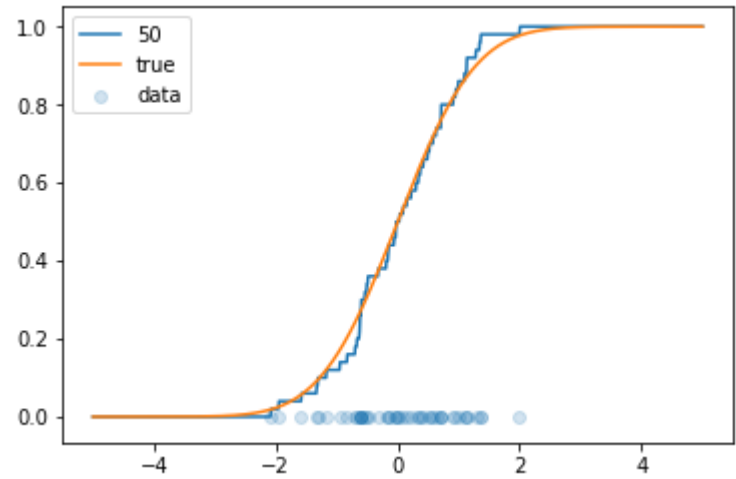
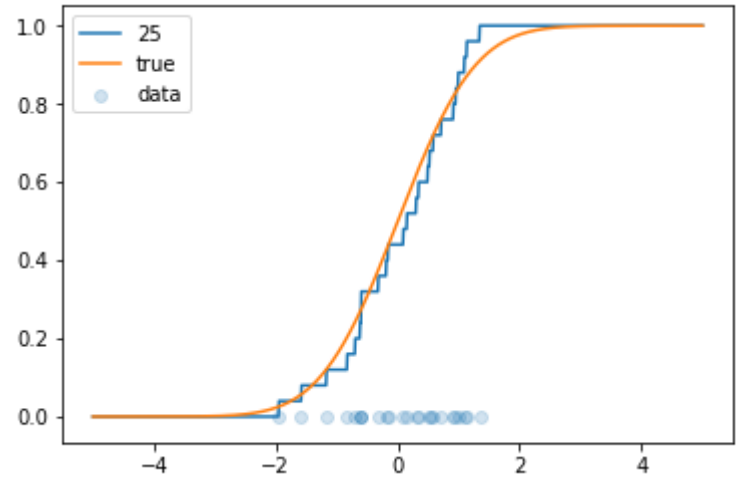
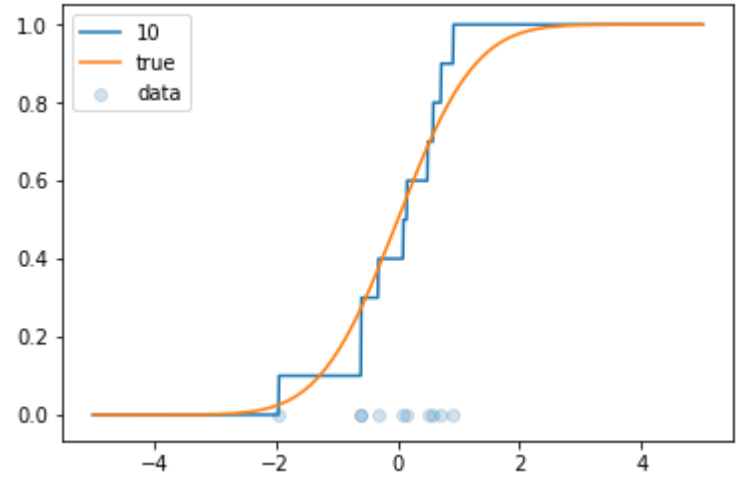
In [2]:

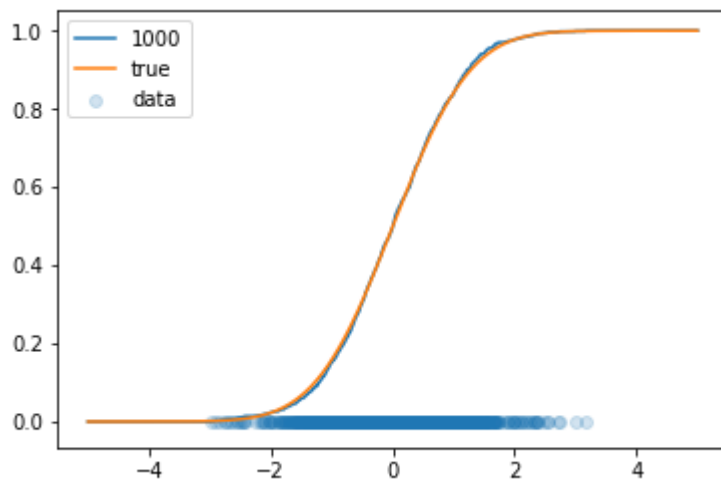
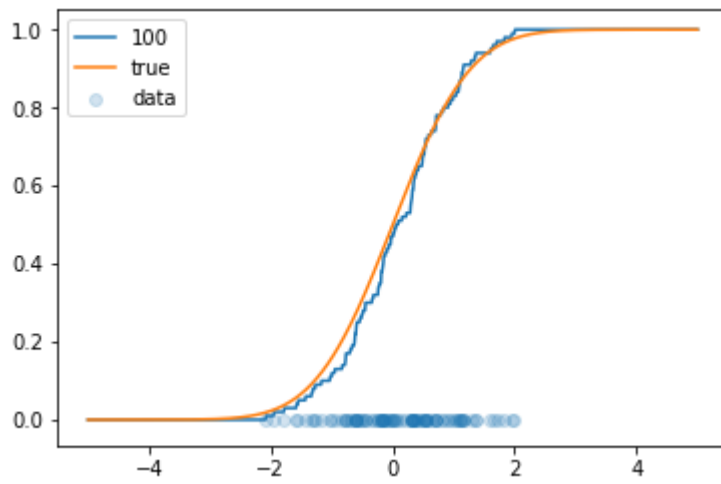
```
def emp_funk(n):
    funk = []
    data = list(np.flipud(np.sort(sample[:n])))
    for x in enumerate(x_row):
        if funk:
            funk.append(funk[-1])
        else:
            funk.append(0)
        while data:
            if x[1] < data[-1]:
                break
            funk[-1] += 1
            del data[-1]
    return np.array(funk)/float(n)
```

Выводим графики эмпирических функций распределения для выборок размера  $n \in [10, 25, 50, 100, 1000]$ . Точками на оси X обозначены реализации св.

In [3]:

```
n_row = [10, 25, 50, 100, 1000]
real_funk = norm.cdf(x_row)
for n in n_row:
    plt.plot(x_row, emp_funk(n), label=str(n))
    plt.plot(x_row, real_funk, label='true')
    plt.scatter(sample[:n], np.zeros(n), alpha=0.2, label='data')
    plt.legend()
    plt.show()
```





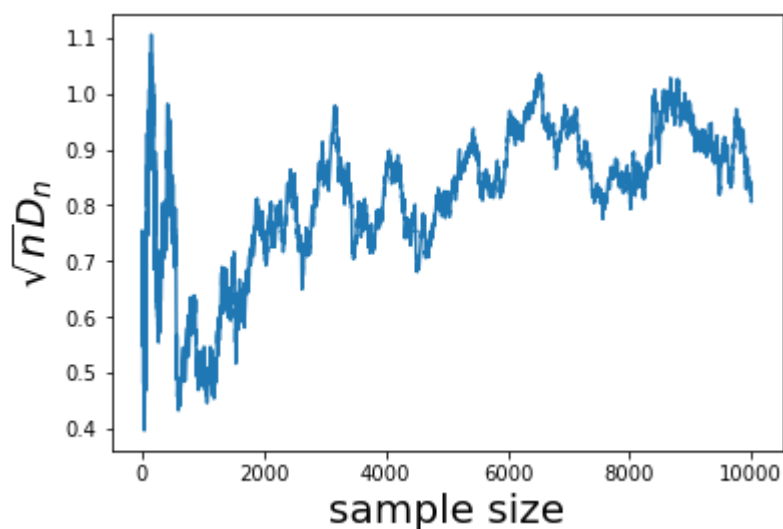
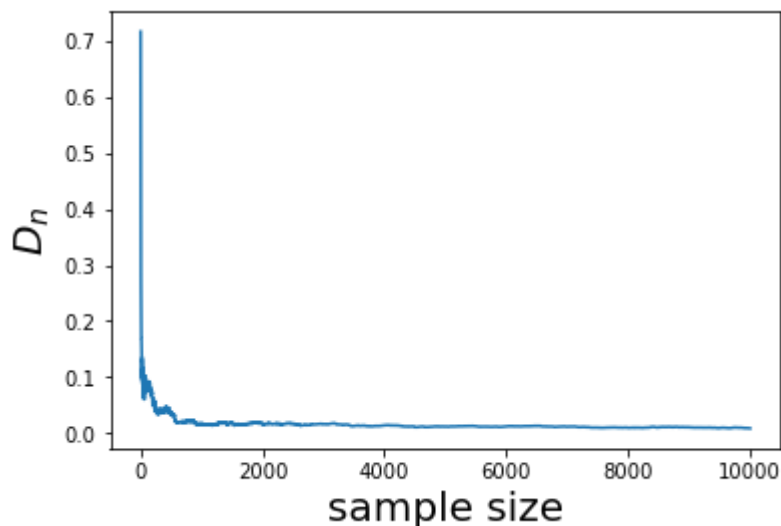
Считаем  $D_n = \sup |F(x) - F_{emp}(x)|$  Выводим графики  $D_n(n)$  и  $\sqrt{n} D_n(n)$

In [7]:

```

D_row = [max(np.abs(real_funk-emp_funk(n))) for n in range(N)[1:]]
plt.plot(range(N)[1:], D_row)
plt.xlabel('sample size', fontsize=20)
plt.ylabel('$D_n$', fontsize=20)
plt.show()
plt.plot(range(N)[1:], [D_row[i-1]*np.sqrt(i) for i in range(N)[1:]])
plt.xlabel('sample size', fontsize=20)
plt.ylabel('$\sqrt{n}D_n$', fontsize=20)
plt.show()

```



In [ ]: