```
!pip3 install -U -q PyDrive

from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials


import numpy as np


def collect_data_via_gd(link: str, filenm: str) -> np.array:
    auth.authenticate_user()
    gauth = GoogleAuth()
    gauth.credentials = GoogleCredentials.get_application_default()
    drive = GoogleDrive(gauth)
    fluff, id = link.split('=')
    downloader = drive.CreateFile({'id':id})
    downloader.GetContentFile(filenm)
    return np.load(filenm)


import matplotlib.pyplot as plt
from scipy.special import factorial
from scipy import stats
import warnings

warnings.simplefilter('ignore')

%matplotlib inline
```

Задача №1

$X_1, \ldots, X_n$ --- выборка из распределения $\mathcal{N}(a_1, \sigma^2)$, $Y_1, \ldots, Y_m$ --- выборка из распределения $\mathcal{N}(a_2, \sigma^2)$, а $Z_1, \ldots, Z_k$ --- выборка из распределения $\mathcal{N}(a_3, \sigma^2)$. Постройте F-критерий размера α для проверки гипотезы $H_0: a_1 = a_2$ и $a_1 + a_2 = a_3$ при неизвестном σ^2.

Протестируйте построенный вами критерий. Рассмотрите $\sigma^2 = 1$.

Рассмотрите три установки с различными значениями a_i на ваш вкус:

- когда гипотеза выполняется,
- когда гипотеза "почти" выполняется,
- когда гипотеза не выполняется.

а) Зафиксируйте значения $n = 100, m = 150, k = 300$. Для каждого эксперимента численно определите минимальный размер критерия, при котором гипотеза H_0 отвергается. Визуализируйте соответствующие квантили на графике распределения Фишера. В этом задании вам может помочь обратная функция распределения, реализованная в `scipy.stats`.

б) Положите $n = m = k = N$, где N изменяется в промежутке от 1 до 1000. Для каждого из трёх экспериментов постройте график $\alpha(N)$, где $\alpha(N)$ --- минимальный размер критерия, при котором гипотеза H_0 отвергается. В этой задаче можно использовать цикл по N.

Матрицы для проверки критерия можно посчитать численно. Не забудьте описать алгоритмы вычислений и сделать выводы.

Решение

Вывод №1

Double-click (or enter) to edit

Задача №2

Пусть $X_1, \ldots, X_n$ --- выборка из распределения $\mathcal{N}(\theta, 1)$. Построить (то есть, в том числе построить её график) функцию мощности критерия Стьюдента проверки гипотезы $H_0 : \theta = 0$ уровня значимости 0.05 для $\theta \in [-10, 10]$, при нескольких различных значениях n. Как объяснить её изменения при растущих n?

Найти такое минимальное n, что при $|\theta_0 - \theta_1| = 1$ при проверке гипотезы $H_0 : \theta = \theta_0$ против альтернативы $H_1 : \theta = \theta_1$ критерием Стьюдента уровня значимости 0.05 вероятность ошибки второго рода станет меньше вероятности ошибки первого рода.

В этой задаче можно использовать циклы, но предпочтительной является конструкция} $[\textbf{... for ... in ...}]$ или аналогичная ей.

Решение

t-критерий Стьюдента:

Применяется для проверки основной гипотезы

$$H_0 : E(X) = \theta.$$

Используя несмещенную оценку дисперсии $s(X)$ получаем *t-критерий*:

$$t_n = \frac{\bar{X} - E_\theta X}{s(X)} \sqrt{n},$$

при принятии основной гипотезы:

$$E(\bar{X}) = \theta.$$

Функцию мощности *t-критерия* проверки гипотезы

$$H_0 : \theta = 0 \; vs \; H_1 : \theta = \theta_1,$$

где:

$$\theta_1 \in [-10; 0) \cup (0; 10]$$

$$t_n = \frac{\bar{X} - E_\theta(X)}{s(X)} \sqrt{n}.$$

Пусть $\theta_0 = 0$, z_α: $\alpha-$квантиль распределения Стьюдента.

Рассмотрим два случая, т.к. у нас двусторонняя альтернатива:

1) $\theta_1 > 0$:

$$P_{\theta_0}(t_{\theta_0} > z_{1-\alpha}) = \alpha,$$
$$S = \{t_{\theta_0} > z_{1-\alpha}\},$$

где:

$$\beta(\theta_1, S) = P_{\theta_1}(t_{\theta_0} > z_{1-\alpha}) =$$
$$= P_{\theta_1}\left(t_{\theta_1} > z_{1-\alpha} - \frac{E_{\theta_1}(X) - E_{\theta_0}(X)}{s(X)}\sqrt{n}\right) =$$
$$= 1 - P_{\theta_1}\left(t_{\theta_1} \le z_{1-\alpha} - \frac{E_{\theta_1}(X) - E_{\theta_0}(X)}{s(X)}\sqrt{n}\right) =$$
$$= 1 - F_{T_n}\left(z_{1-\alpha} - \frac{E_{\theta_1}(X) - E_{\theta_0}(X)}{s(X)}\sqrt{n}\right).$$

2) $\theta_1 < 0$:

$$P_{\theta_0}(t_{\theta_0} \le z_\alpha) = \alpha,$$
$$S = \{t_{\theta_0} \le z_\alpha\},$$

где:

$$\beta(\theta_1, S) = P_{\theta_1}(t_{\theta_0} \le z_\alpha) =$$
$$= P_{\theta_1}\left(t_{\theta_1} \le z_\alpha - \frac{E_{\theta_1}(X) - E_{\theta_0}(X)}{s(X)}\sqrt{n}\right) =$$
$$= F_{T_n}\left(z_\alpha - \frac{E_{\theta_1}(X) - E_{\theta_0}(X)}{s(X)}\sqrt{n}\right)$$

```
alpha = 0.05
theta = np.linspace(-10, 10, 1000)
n = [10**2, 10**3, 10**4, 10**5, 10**6]
```

Несмещенную оценка дисперсии $s(X)$:

```
def var(sample):
    sample_size = len(sample)
    return np.sqrt(sample_size / (sample_size-1) * np.var(sample))
```

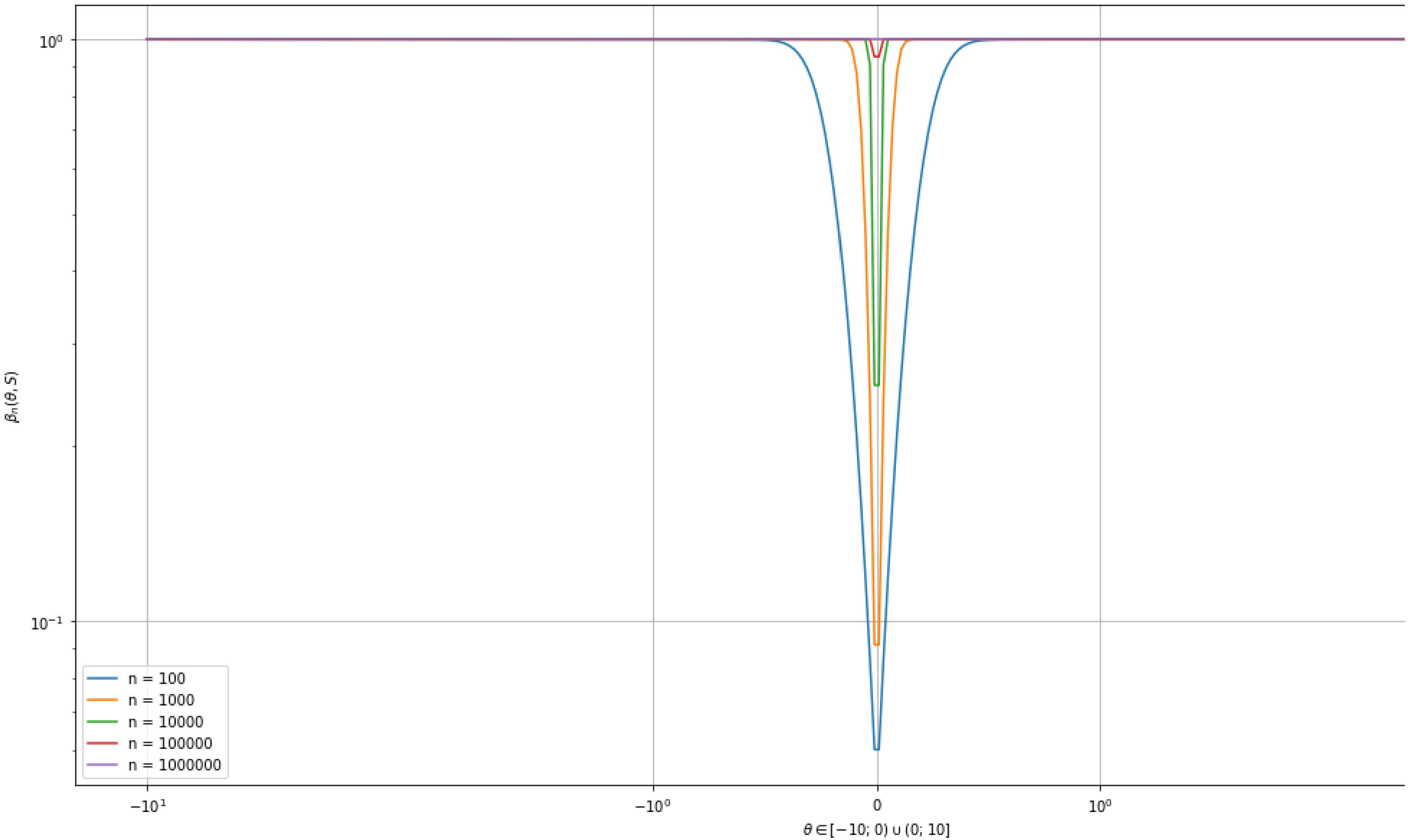
Напишем функцию, которая по размеру выборки, генерирует саму выборку и для каждого θ_1 подсчитывает мощность критерия $\beta(\theta_1, S)$:

```
def vcalc_t_test_criteria_power(sample_size):
    sample = stats.norm.rvs(size=sample_size)
    def scalar_calc(theta_):
        nonlocal sample, sample_size
        if theta_ < 0:
            z = stats.t.ppf(alpha, df=sample_size)
            return stats.t.cdf(
                z - theta_*np.sqrt(sample_size)/var(sample),
                df=sample_size)
        else:
            z = stats.t.ppf(1 - alpha, df=sample_size)
            return 1 - stats.t.cdf(
                z - theta_*np.sqrt(sample_size)/var(sample),
                df=sample_size)
    return np.vectorize(scalar_calc)
```

Теперь исследуем поведение функции мощности при разных размерах выборки при каждом θ_1:

```
plt.figure(figsize=(20, 10))
plt.xlabel(r'$\theta \in [-10;0) \cup (0;10]$')
plt.ylabel(r'$\beta_{n}(\theta, S)$')
plt.xscale('symlog')
plt.yscale('log')
for sample_size in n:
    plt.plot(
        theta,
        vcalc_t_test_criteria_power(sample_size)(theta),
```

```
        vcalc_t_test_criteria_power(sample_size)(theta),
        label=f"n = {sample_size}")
plt.legend()
plt.grid()
plt.show()
```



Полученный график даёт нам понять, что при $n \geq 10000$ мощность *t-критерия* равна единице, значит, вероятность совершить ошибку 2 рода практически нулевая. Это обьясняется тем, что при увеличении обьема выборки улучшается качество проверки гипотезы.

Теперь найдем такое минимальное n, что при проверке гипотезы:

$$H_0 : \theta = 0 \; vs \; H_1 : \theta = \theta_1,$$

где:

$$|\theta_0 - \theta_1| = 1,$$

t-критерием уровня значимости 0.05 вероятность ошибки 1 рода меньше вероятности ошибки 2 рода.
Рассмотрим два случая:

1.

$$H_0 : \theta = \theta_0 \; vs \; H_1 : \theta = \theta_1 = \theta_0 + 1$$

$$\beta(\theta, S) = P_{\theta_1}(t_{\theta_0} > z_{1-\alpha}) =$$
$$= P_{\theta_1}\left(t_{\theta_1} > z_{1-\alpha} - \frac{E_{\theta_1}(X) - E_{\theta_0}(X)}{s(X)}\sqrt{n}\right) =$$
$$= 1 - F_{T_n}\left(z_{1-\alpha} - \frac{\sqrt{n}}{s(X)}\right),$$

где T_n — распределение Стьюдента.

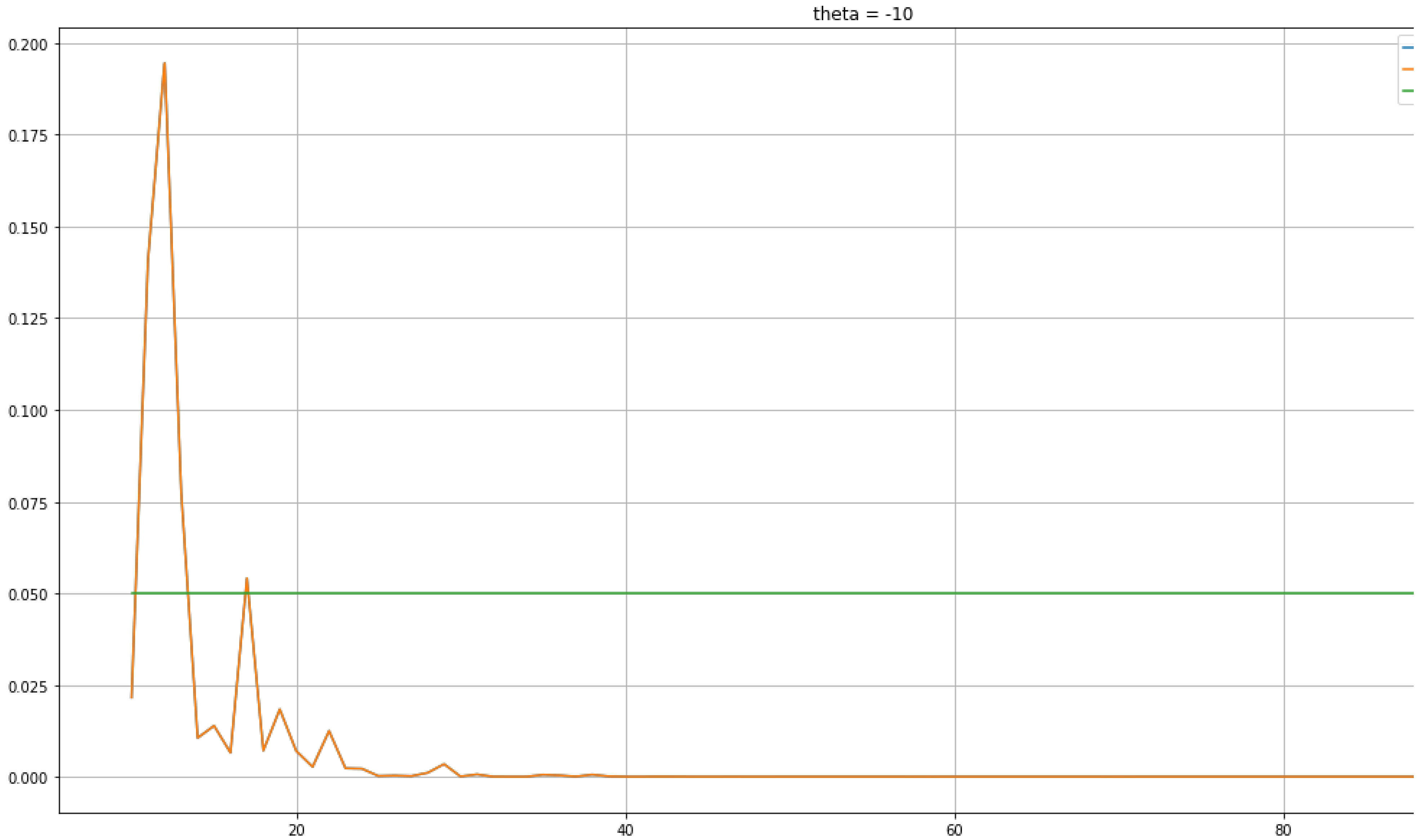
2.

$$H_0 : \theta = \theta_0 \; vs \; H_1 : \theta = \theta_1 = \theta_0 - 1$$

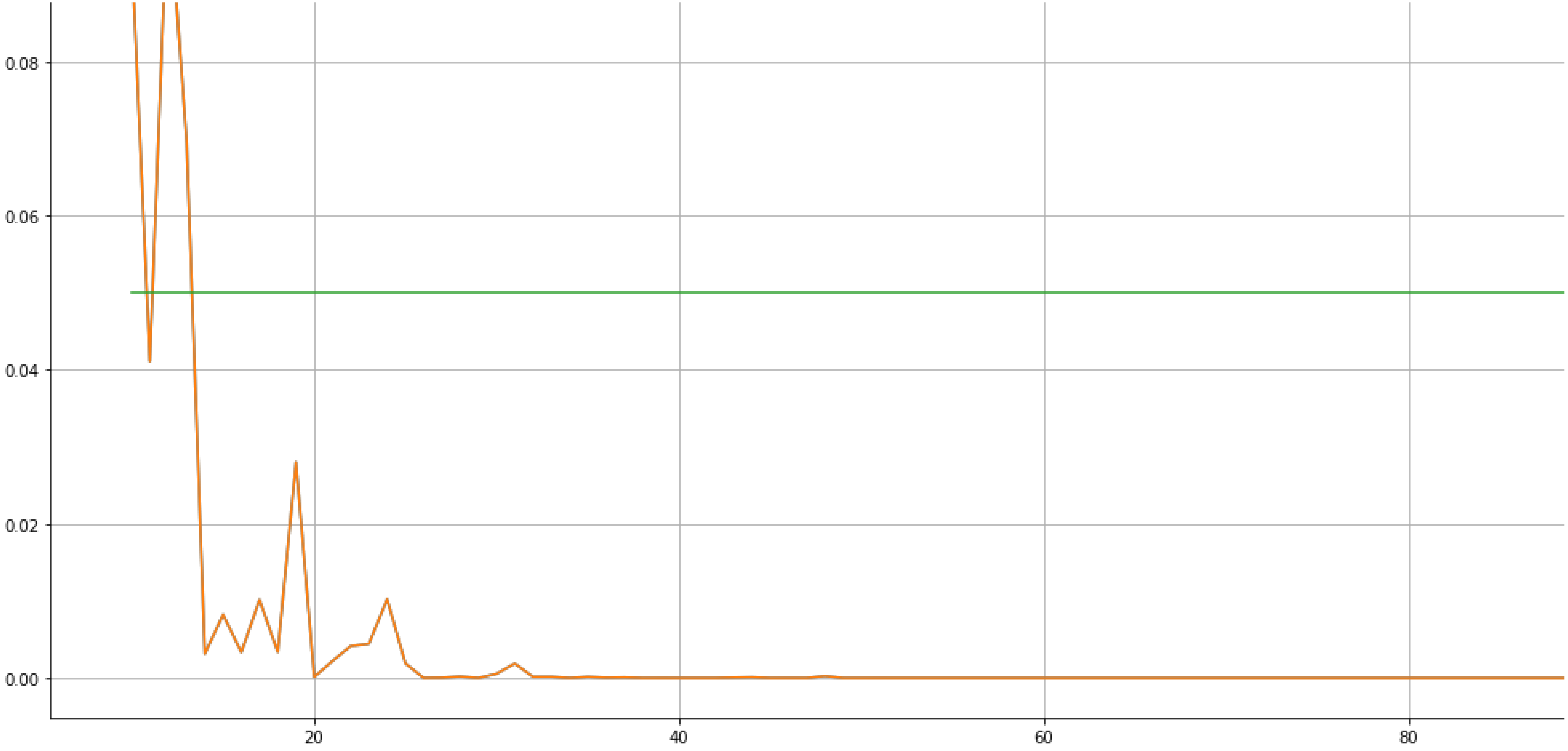
$$\beta(\theta, S) = P_{\theta_1}(t_{\theta_0} \leq z_{\alpha}) =$$
$$= P_{\theta_1}\left(t_{\theta_1} \leq z_{\alpha} - \frac{E_{\theta_1}(X) - E_{\theta_0}(X)}{s(X)}\sqrt{n}\right) =$$
$$= F_{T_n}\left(z_{\alpha} + \frac{\sqrt{n}}{s(X)}\right)$$

Определим минимальное n тривиальным перебором для $\theta \in \overline{-10, 10}$. Для большей наглядности построим графики и определим по ним:

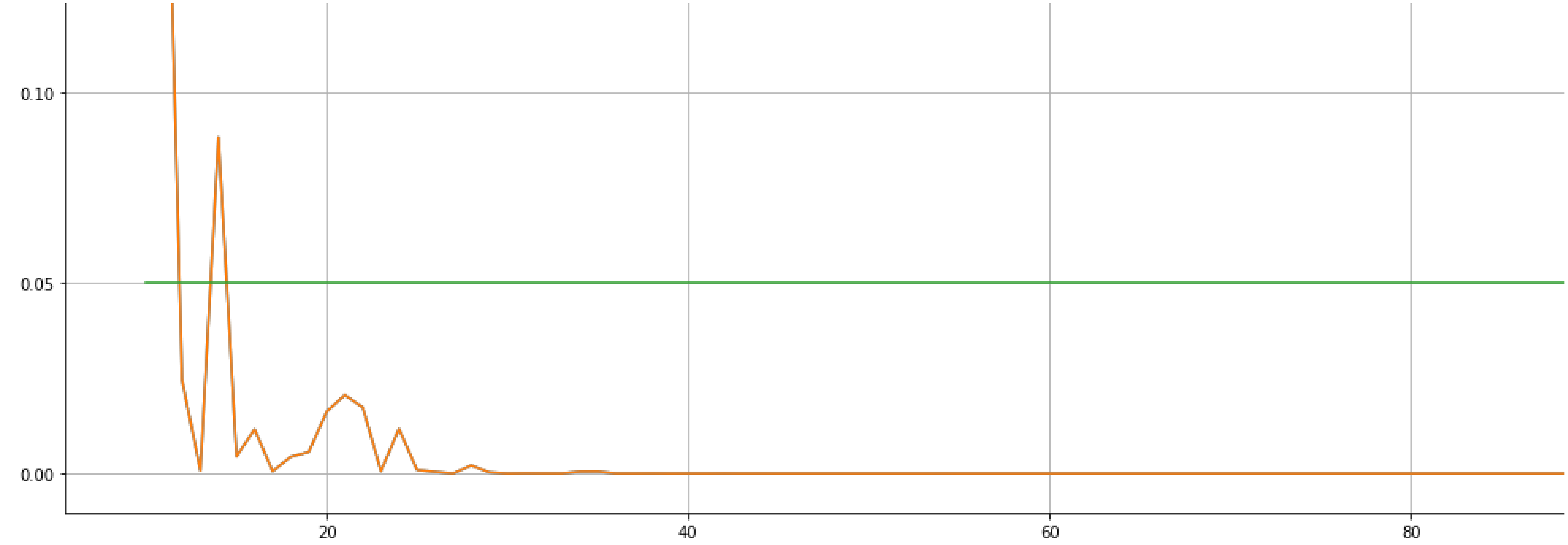
```
theta = np.array(list(range(-10,11)))
grid = np.array([i for i in range(10, 100)])
for theta_ in theta:
    error2nd1var = []
    error2nd2var = []
    for sample_size in grid:
        sample = stats.norm.rvs(loc=theta_, size=sample_size)
        z1 = stats.t.ppf(1-alpha, df=sample_size)
        error2nd1var.append(stats.t.cdf(z1 - np.sqrt(sample_size) / var(sample),
                                          df=sample_size))
        z2 = stats.t.ppf(alpha, df=sample_size)
        error2nd2var.append(1 - stats.t.cdf(z2 + np.sqrt(sample_size) / var(sample),
                                              df=sample_size))
    plt.figure(figsize=(20,10))
    plt.plot(
        grid,
        error2nd1var,
        label="Ошибка 2го рода 1 вариант")
    plt.plot(
        grid,
        error2nd2var,
        label="Ошибка 2го рода 2 вариант")
    plt.plot(
        grid,
        alpha*np.ones(len(grid)),
        label="Ошибка 1го рода")
    plt.title(f"theta = {theta_}")
    plt.legend()
    plt.grid()
    plt.show()
```


theta = -10
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
20
40
60
80

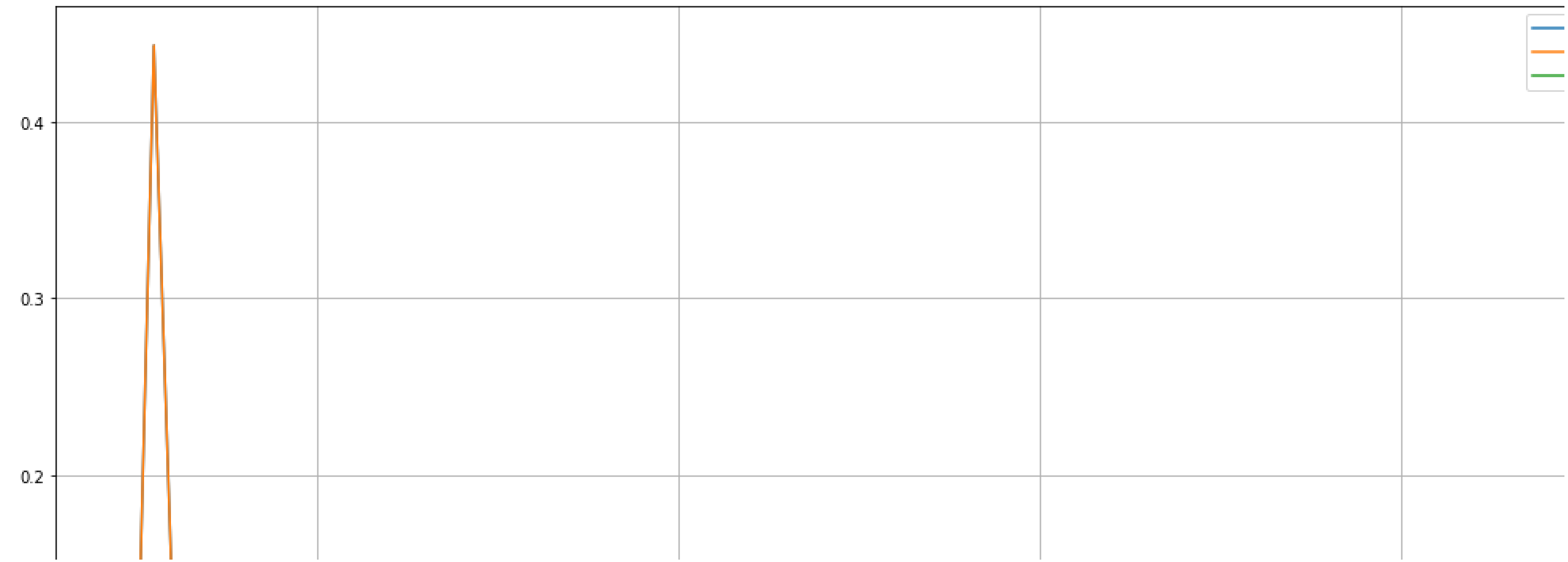

theta = -9
0.10

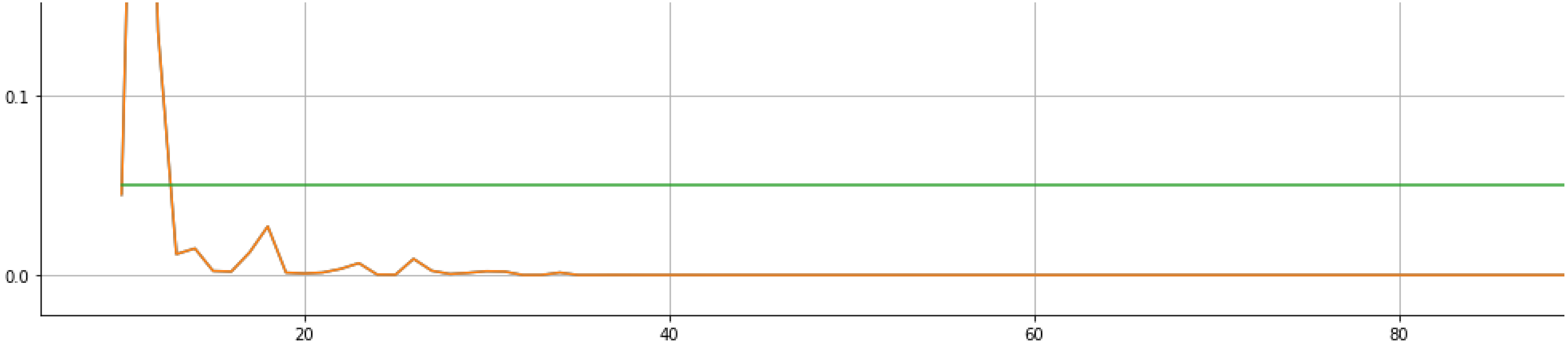

0.08
0.06
0.04
0.02
0.00
20
40
60
80


theta = -8
0.20
0.15

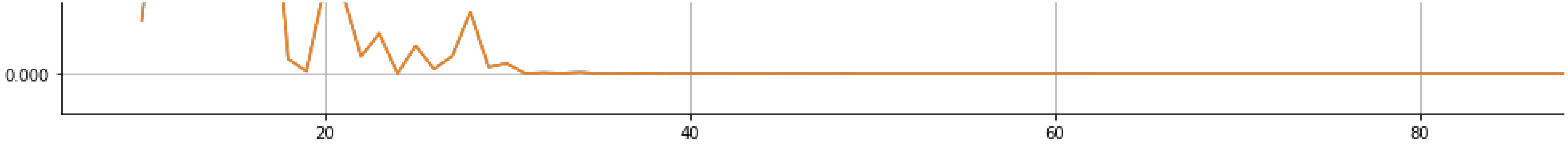


theta = -7

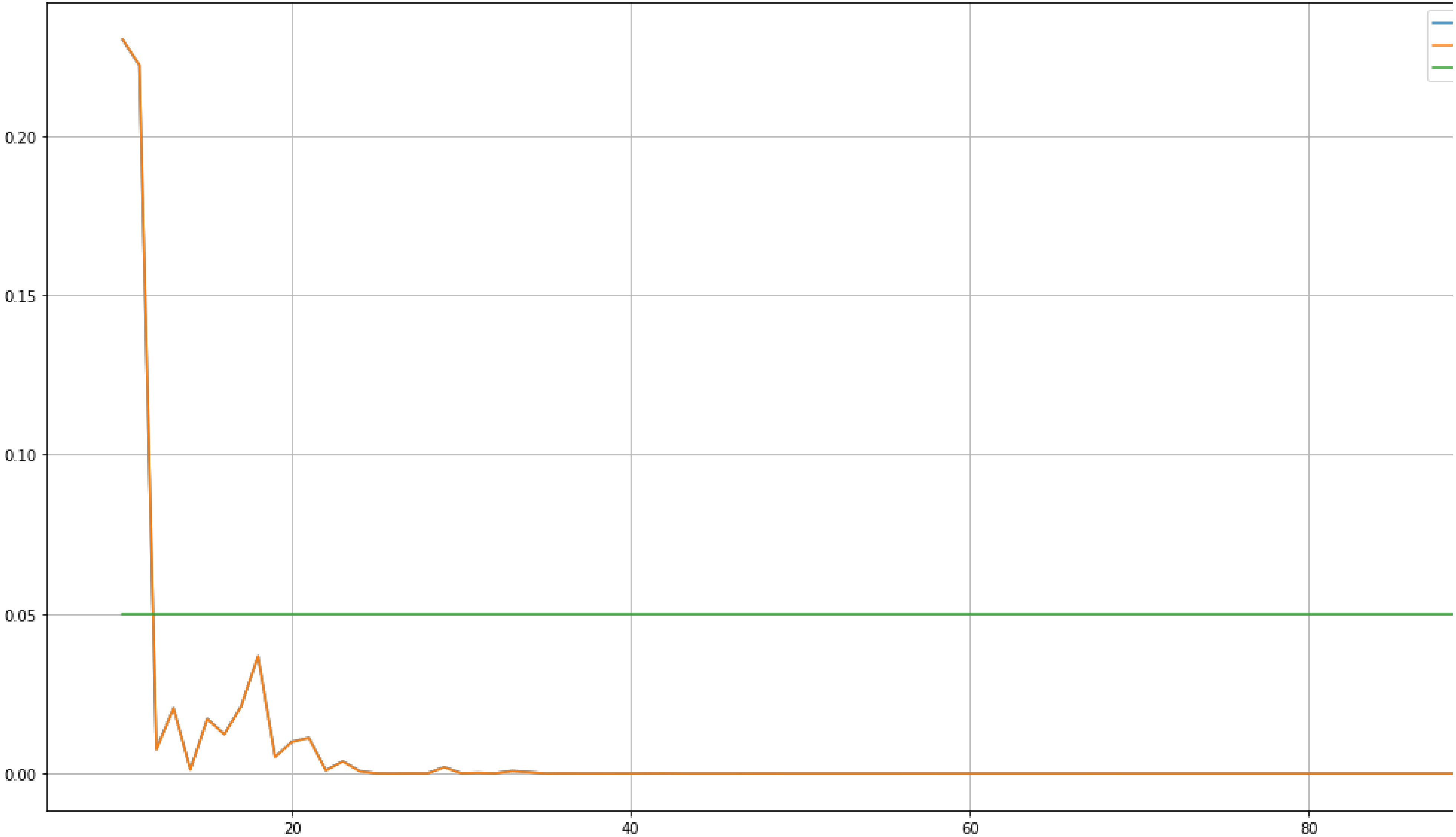



0.1
0.0
20
40
60
80


theta = -6
0.175
0.150
0.125
0.100
0.075
0.050
0.025



theta = -5



theta = -4