

Санкт-Петербургский государственный политехнический
университет Петра Великого

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

Лабораторная работа

Шум

Выполнил студент гр. 3530901/80201

И.С. Иванов

Преподаватель:

Н.В. Богач

Санкт-Петербург
2021

Содержание

1	Упражнение №1: Изучение шума	5
2	Упражнение №2: метод Бартлетта	11
3	Упражнение №3: Получение спектра курса валюты Bitcoin	12
4	Упражнение №4: UncorrelatedPoissonNoise	14
5	Упражнение №5: Алгоритм Voss-McCartney	16
6	Выводы	18

Список иллюстраций

1	Спектр сегмента	5
2	Спектр сегмента в логарифмическом масштабе	6
3	Наложенные спектры двух сегментов	6
4	Наложенные спектры двух сегментов в логарифмическом масштабе	7
5	Спектрограмма сегмента	7
6	Спектр сегмента	8
7	Спектр сегмента в логарифмическом масштабе	8
8	Наложенные спектры двух сегментов	9
9	Наложенные спектры двух сегментов в логарифмическом масштабе	9
10	Спектрограмма сегмента	10
11	Спектры после использования метода Бартлетта	11
12	График курса Bitcoin	12
13	Логарифмический спектр курса Bitcoin	13
14	Спектрограмма полученного звука	15
15	Логарифмический спектр полученного звука	15
16	График полученного сигнала	16
17	Логарифмический график полученного сигнала	17

Листинги

1	Чтение файла, выделение фрагмента, вывод спектра	5
2	Метод Бартлетта	11
3	Считывание файла	12
4	Построение графика	12
5	Класс UncorrelatedPoissonNoise	14
6	Создание и прослушивание сигнала	14
7	Сравнение ожидаемых частиц и полученных	14
8	Алгоритм Voss-McCartney	16

1 Упражнение №1: Изучение шума

Во первом упражнении необходимо скачать файлы с шумом природы, например дождь или морские волны. Выделить из этих сигналов спектры и установить, на какой шум похож каждый сигнал.

Для выполнения были скачаны аудио файлы звука шторма и моря.

Прочитаем файл. Выделим сегмент длиной в 1 секунду. Посмотрим на спектр выделенного сегмента.

```
1     from thinkdsp import read_wave
2
3     wave = read_wave('Sounds/127596__juskiddink__wind-in-birch-trees-a-passing-
4     -sheep.wav')
5
6     segment = wave.segment(start=1.5, duration=1.0)
7
8     spectrum = segment.make_spectrum()
9     spectrum.plot_power()
```

Листинг 1: Чтение файла, выделение фрагмента, вывод спектра

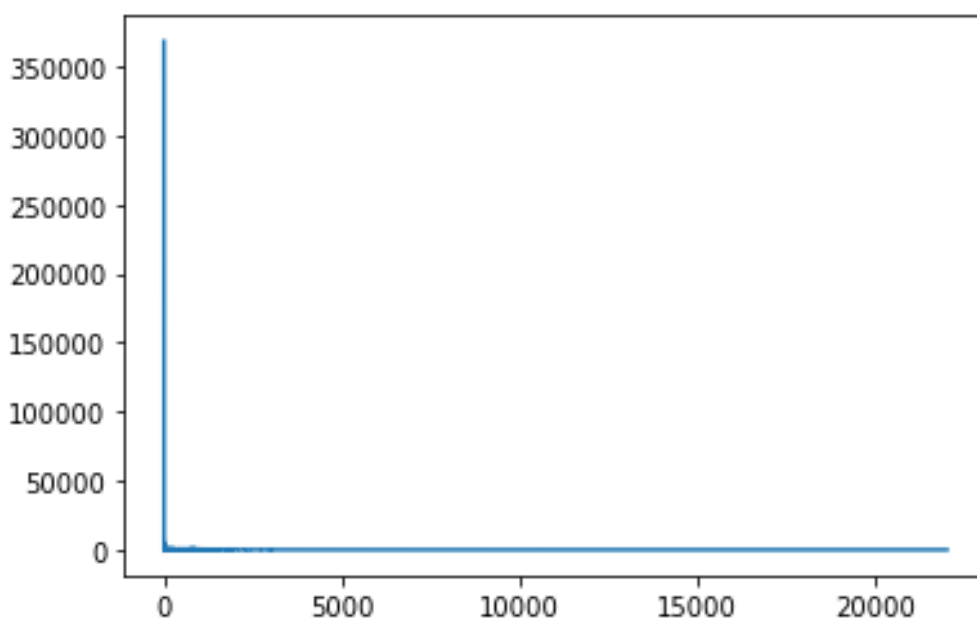


Рис. 1: Спектр сегмента

Так как большей амплитуде соответствует меньшее значение частоты, можно сказать, что это красный или розовый шум.

Посмотрим на спектр сегмента в логарифмическом масштабе.

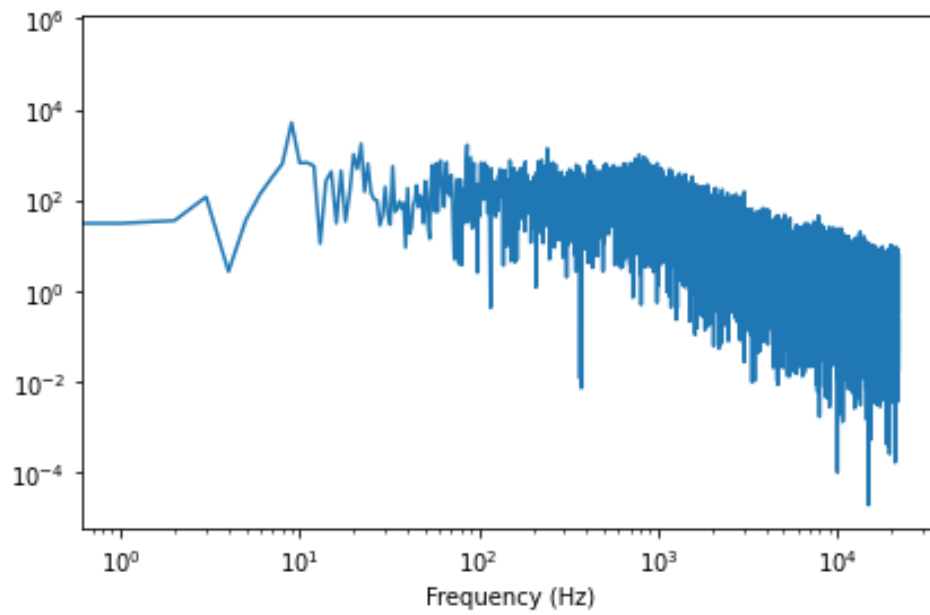


Рис. 2: Спектр сегмента в логарифмическом масштабе

Так же проверим другой сегмент сигнала.

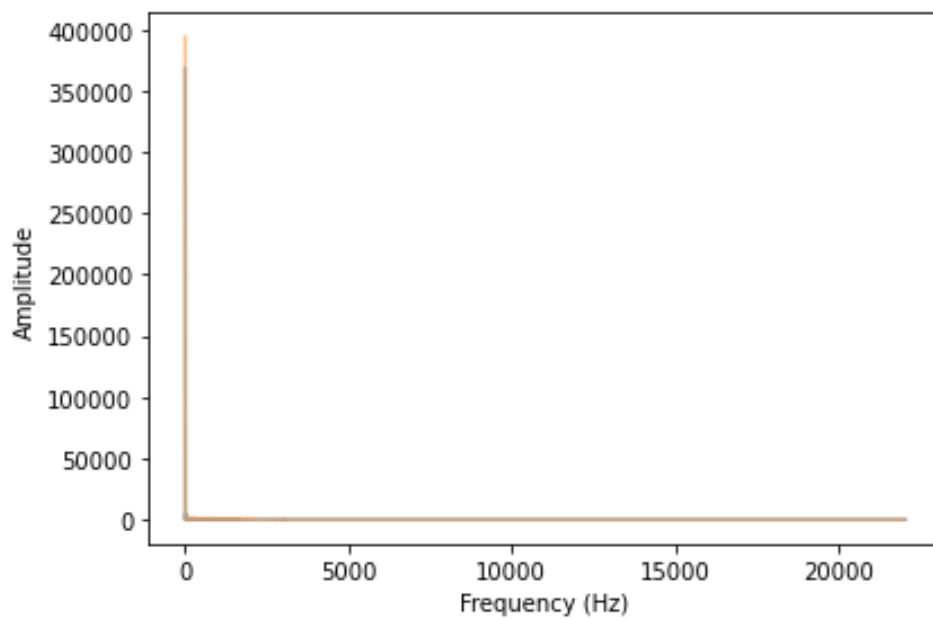


Рис. 3: Наложенные спектры двух сегментов

На основании спектра этого сегмента так же можно сказать, что это красный или розовый шум.

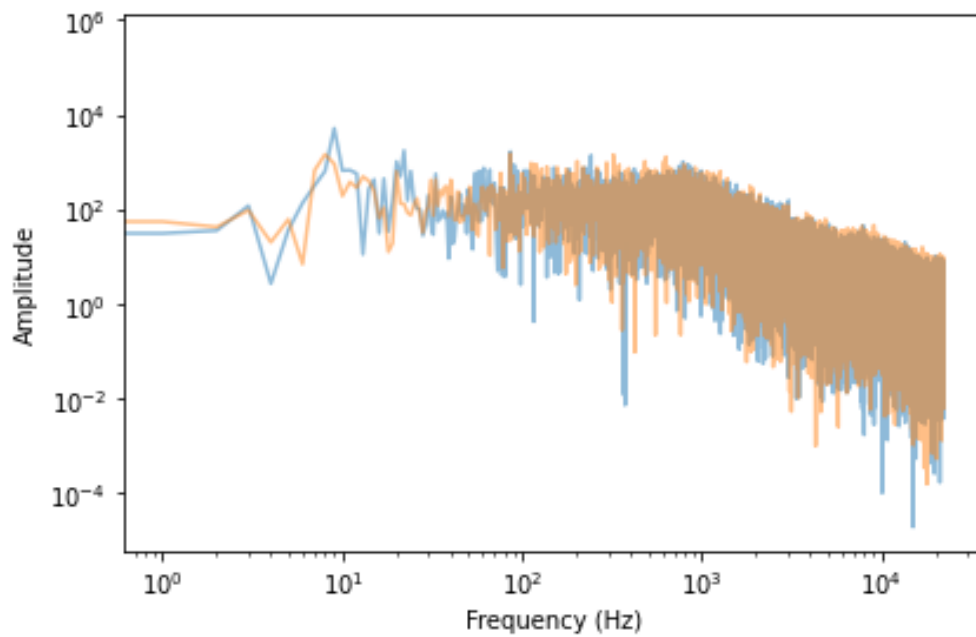


Рис. 4: Наложенные спектры двух сегментов в логарифмическом масштабе

На графике видно, что сигнал не сильно меняется с течением времени.

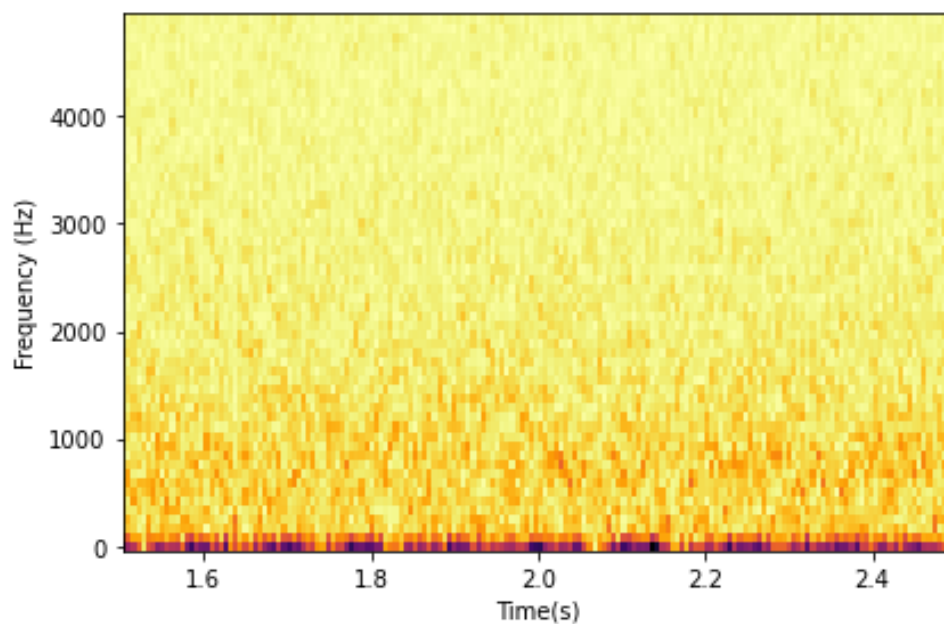


Рис. 5: Спектрограмма сегмента

Изучим файл со звуками морских волн. Выделим сегмент длиной в 1 секунду. Посмотрим на спектр выделенного сегмента.

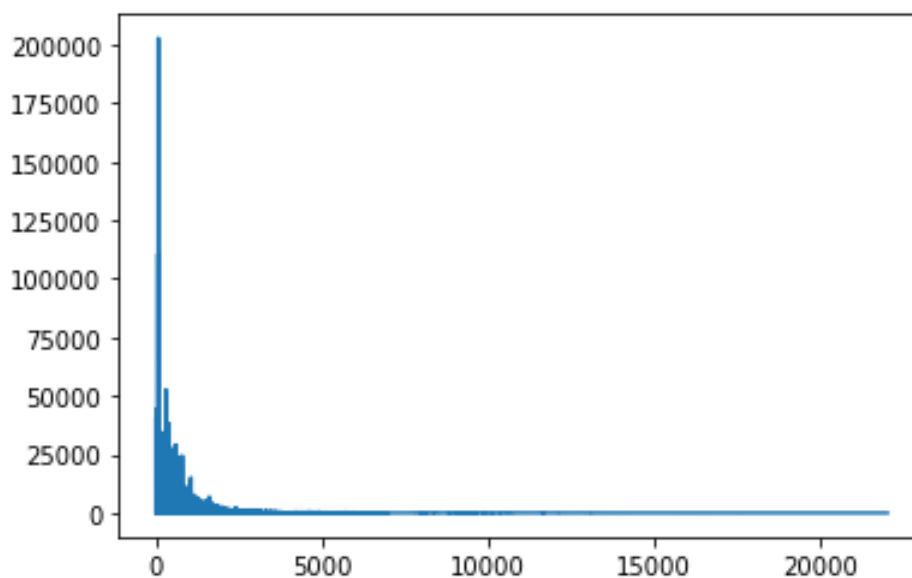


Рис. 6: Спектр сегмента

Так как большей амплитуде соответствует меньшее значение частоты, можно сказать, что это красный или розовый шум.

Посмотрим на спектр сегмента в логарифмическом масштабе.

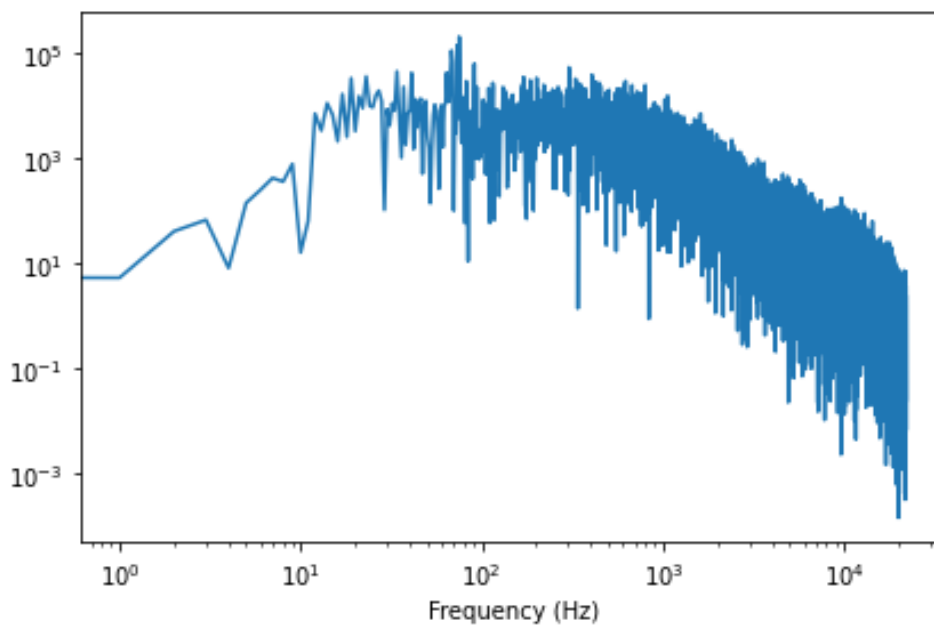


Рис. 7: Спектр сегмента в логарифмическом масштабе

Так же проверим другой сегмент сигнала.

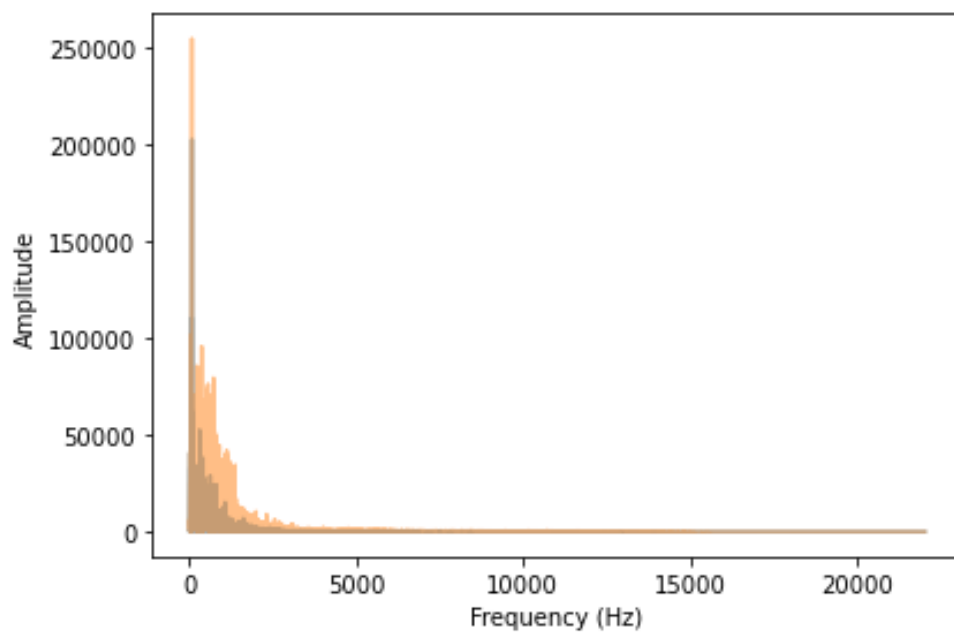


Рис. 8: Наложенные спектры двух сегментов

На основании спектра этого сегмента так же можно сказать, что это красный или розовый шум.

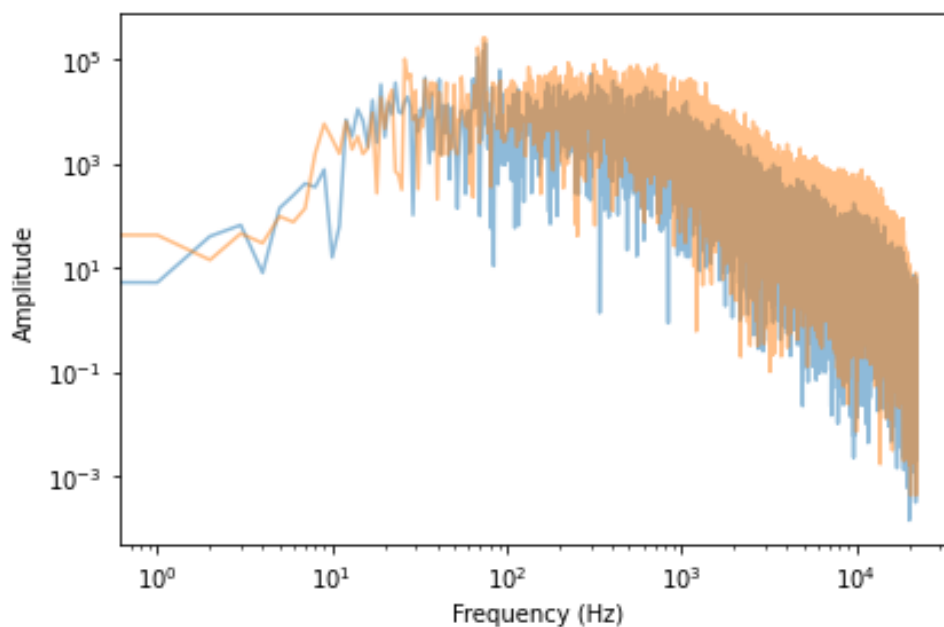


Рис. 9: Наложенные спектры двух сегментов в логарифмическом масштабе

На графике видно, что сигнал не сильно меняется с течением времени.

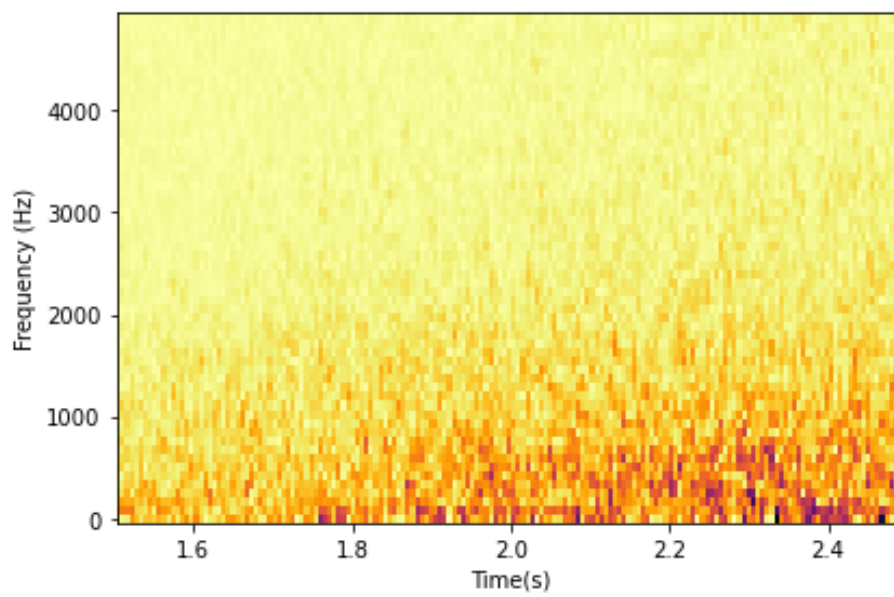


Рис. 10: Спектрограмма сегмента

2 Упражнение №2: метод Бартлетта

Во втором упражнении необходимо реализовать метод Бартлетта и использовать его для оценки спектра мощности шумового сигнала.

Реализуем метод Бартлетта:

```
1 from thinkdsp import Spectrum
2
3 def bartlett_method(wave, seg_length=512, win_flag=True):
4     spectro = wave.make_spectrogram(seg_length, win_flag)
5     spectrums = spectro.spec_map.values()
6     psds = [spectrum.power for spectrum in spectrums]
7     hs = np.sqrt(sum(psds) / len(psds))
8     fs = next(iter(spectrums)).fs
9     spectrum = Spectrum(hs, fs, wave.framerate)
10    return spectrum
11
```

Листинг 2: Метод Бартлетта

Проверим работу метода. Вызовем метод дважды для двух сегментов из предыдущего упражнения и выведем полученные спектры на график:

Выведем конец сигнала.

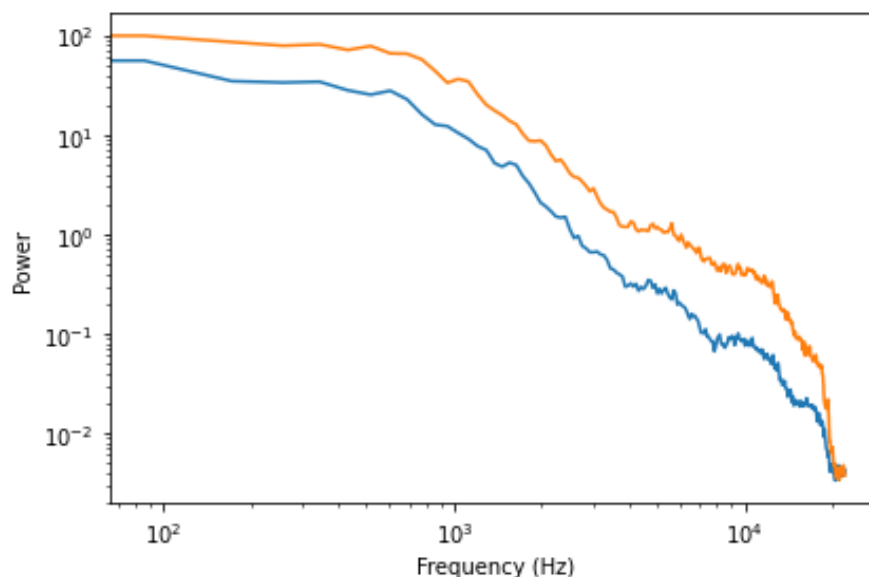


Рис. 11: Спектры после использования метода Бартлетта

Изучив полученные спектры, можно сделать вывод, что в них есть связь между частотой и амплитудой. Зависимость линейная.

3 Упражнение №3: Получение спектра курса валюты Bitcoin

В третьем упражнении нам необходимо скачать CSV файл с историческими данными курса Bitcoin. Необходимо вычислить спектр цен как функцию времени и установить, на какой шум похож спектр.

Считаем файл:

```
1 import pandas as pd
2
3 data = pd.read_csv('Res/BTC_USD_2013-10-01_2021-05-04-CoinDesk.csv',
4                   parse_dates=[0])
5
```

Листинг 3: Считывание файла

Посмотрим на график полученных данных:

```
1 ys = data['Closing Price (USD)']
2 ts = data.index
3
4 from thinkdsp import Wave
5
6 wave = Wave(ys, ts, framerate=1)
7 wave.plot()
8 decorate(xlabel='Time (days)')
9
```

Листинг 4: Построение графика

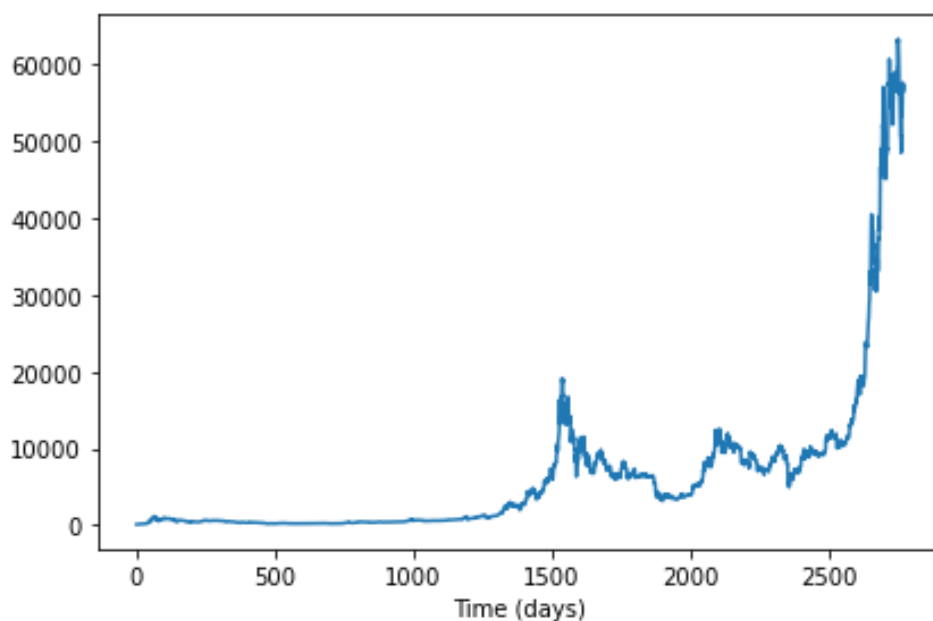


Рис. 12: График курса Bitcoin

Построим логарифмический спектр курса Bitcoin

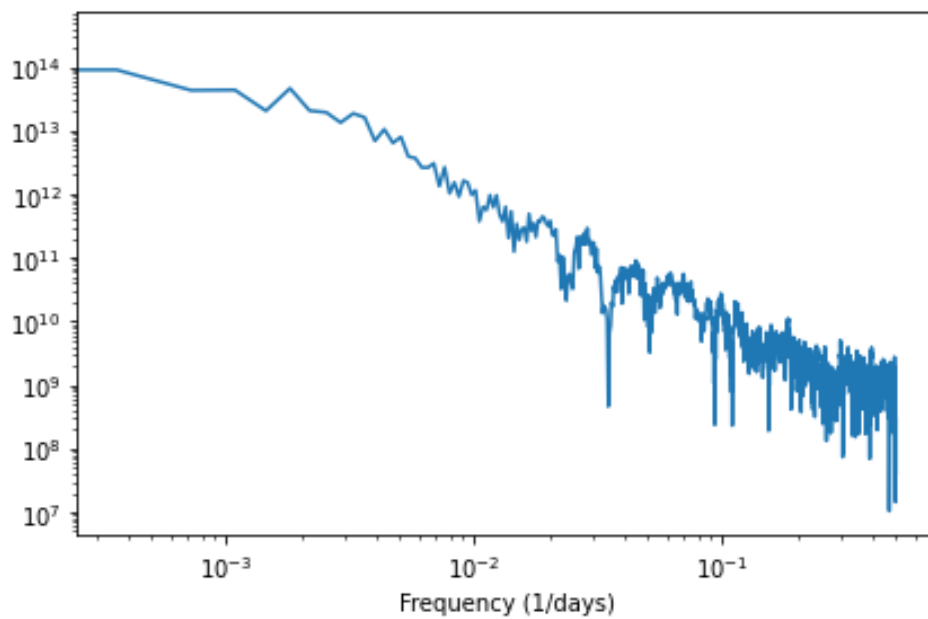


Рис. 13: Логарифмический спектр курса Bitcoin

Из спектра можно понять, что это розовый или красный шум.

Посмотрим на "slope" полученного спектра.

"Slope" равен -1.7835653687618445.

Для красного шума характерно значение -2, значит курс Bitcoin является розовым шумом.

4 Упражнение №4: UncorrelatedPoissonNoise

В четвертом упражнении необходимо реализовать класс `UncorrelatedPoissonNoise` наследующий `Noise` и предоставляющий `evaluate`. Необходимо сгенерировать случайные величины из распределения Пуассона, а так же пару секунд UP и прослушать. При малых значениях `amp` звук будет похож на счетчик Гейгера, а при больших на белый шум. Вычислить и вывести спектр для этих сигналов.

Напишем класс `UncorrelatedPoissonNoise`:

```
1 from thinkdsp import Noise
2
3 class UncorrelatedPoissonNoise(Noise):
4     def evaluate(self, ts):
5         ys = np.random.poisson(self.amp, len(ts))
6         return ys
7
```

Листинг 5: Класс `UncorrelatedPoissonNoise`

Создадим сигнал с `amp = 0.001` и прослушаем.

```
1 amp = 0.001
2 framerate = 10000
3 duration = 1
4
5 signal = UncorrelatedPoissonNoise(amp=amp)
6 wave = signal.make_wave(duration=duration, framerate=framerate)
7 wave.make_audio()
8
```

Листинг 6: Создание и прослушивание сигнала

Получившийся звук похож на счетчик Гейгера.

Сверим ожидаемое количество частиц и полученное.

```
1 expected = amp * framerate * duration
2 actual = sum(wave.ys)
3 print(expected, actual)
4
```

Листинг 7: Сравнение ожидаемых частиц и полученных

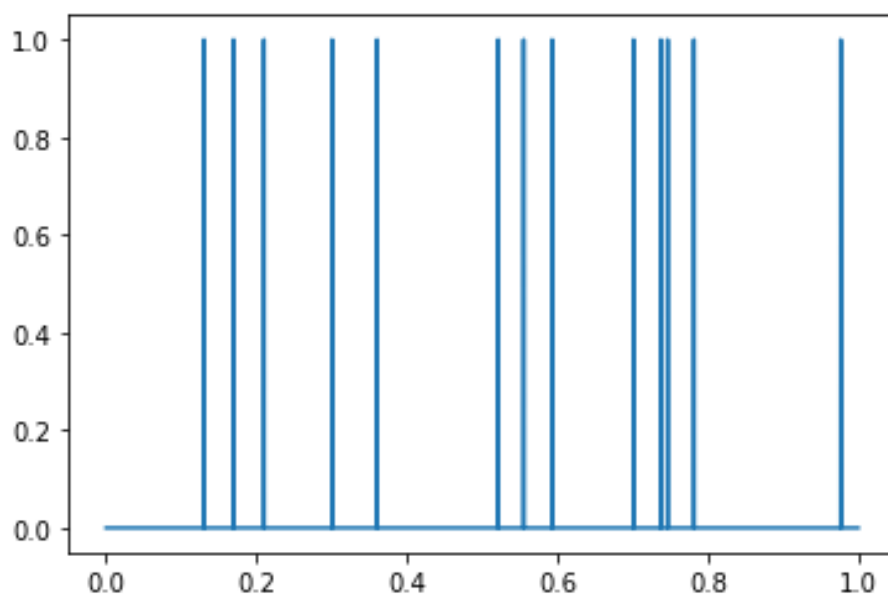


Рис. 14: Спектрограмма полученного звука

Посмотрим на логарифмический спектр.

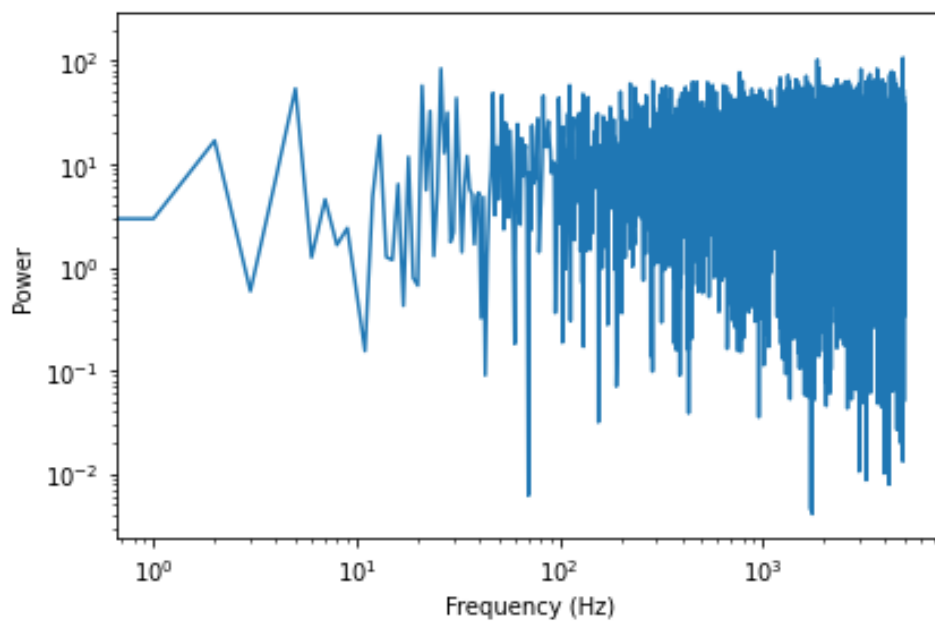


Рис. 15: Логарифмический спектр полученного звука

"Slope- 0.03737836229373747.

На основе полученных данных, можно сделать вывод, что это белый шум.

5 Упражнение №5: Алгоритм Voss-McCartney

В пятом упражнении необходимо реализовать алгоритм Voss-McCartney, вычислить спектр и убедиться, что соотношение между мощностью и частотой соответствующие.

Реализуем алгоритм Voss-McCartney:

```
1     def voss(nrows, ncols=16):
2         array = np.empty((nrows, ncols))
3         array.fill(np.nan)
4         array[0, :] = np.random.random(ncols)
5         array[:, 0] = np.random.random(nrows)
6
7         n = nrows
8         cols = np.random.geometric(0.5, n)
9         cols[cols >= ncols] = 0
10        rows = np.random.randint(nrows, size=n)
11        array[rows, cols] = np.random.random(n)
12
13        df = pd.DataFrame(array)
14        df.fillna(method='ffill', axis=0, inplace=True)
15        total = df.sum(axis=1)
16
17        return total.values
18
```

Листинг 8: Алгоритм Voss-McCartney

Создадим, выведем график и прослушаем сигнал, полученный с помощью этого алгоритма.

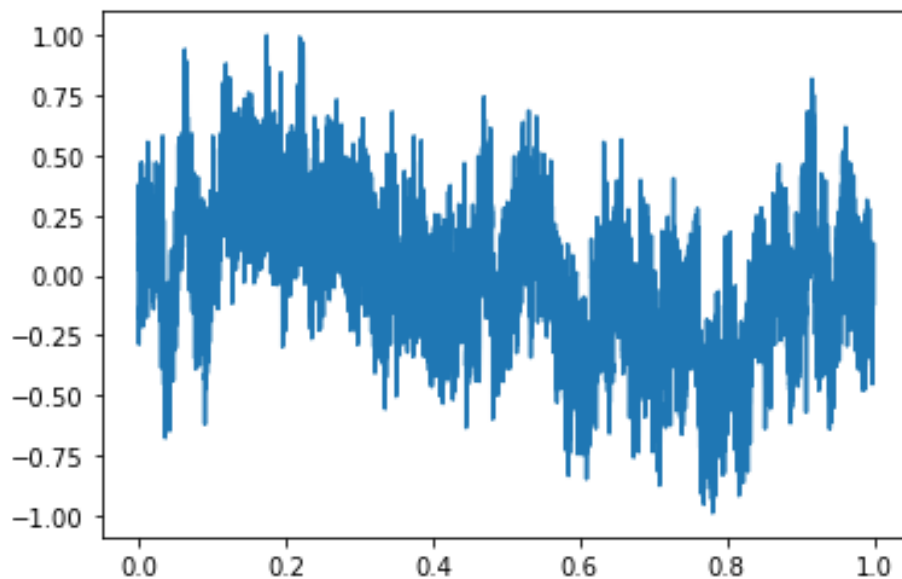


Рис. 16: График полученного сигнала

Посмотрим на логарифмический график, чтобы убедиться в правильности зависимости амплитуды от частоты.

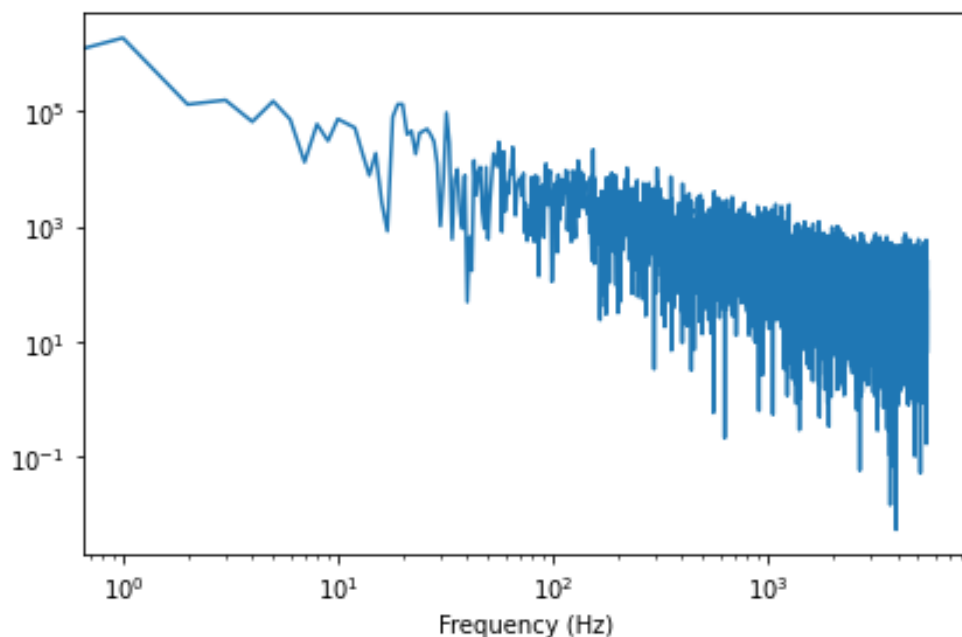


Рис. 17: Логарифмический график полученного сигнала

"Slope- -0.9913328610110239.

Так как "Slope" близок к -1, можно сделать вывод, что сигнал является розовым шумом.

6 Выводы

В результате выполнения данной лабораторной работы мы изучили, понятие шума и как работать с ним. Также научились строить логарифмические спектры шумов. Были преобразованы данные курса валюты Bitcoin и представлены в виде источника изучения шума. Был создан класс для генерации случайных величин из распределения Пуассона и создан методы, реализующий алгоритм Voss-McCartney.