Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Финансовый университет при Правительстве Российской Федерации»

Кафедра «Теория вероятностей и математическая статистика»

Курсовая работа на тему:

«Проверка гипотезы о нормальном распределении логарифмической доходности по критерию Дэвида-Хартли-Пирсона»

Вид исследуемых данных:

Котировки акций компаний, входящих в индекс S&P 500

Выполнила:

студентка группы ПМ20-2

Короткова В.В.



Руководитель:

доцент., к.ф.-м.н.

Игудесман К.Б.

Москва, 2022

**Оглавление**

[**Введение** 2](#_Toc104030695)

[**Предварительный анализ данных** 3](#_Toc104030696)

[**Теоретическая справка** 8](#_Toc104030697)

[**1.** **Основные понятия проверки гипотез** 8](#_Toc104030698)

[**2.** **Критерий Дэвида-Хартли-Пирсона** 10](#_Toc104030699)

[**3.** **Критерий Колмогорова** 10](#_Toc104030700)

[**Проверка гипотезы для модельных данных** 12](#_Toc104030701)

[**Выбор альтернативной гипотезы и оценка мощности критерия** 15](#_Toc104030702)

[**Проверка гипотезы для реальных данных** 17](#_Toc104030703)

[**Заключение** 22](#_Toc104030704)

[**Литература** 23](#_Toc104030705)

[**Приложение** 24](#_Toc104030706)

# **Введение**

Цель данной курсовой работы - проверить гипотезу о нормальности логарифмической доходности. Она будет осуществляться с помощью статистического критерия Дэвида-Хартли-Пирсона. Для этого буду анализировать котировки акций нескольких компаний, входящих в индекс S&P 500 в период с 01.01.2015 по 01.01.2022 гг.

Я выбрала данный индекс, содержащий около 500 акций, так как он является одним из основных американских фондовых индексов и имеет большую капитализацию. Для того чтобы подобрать компании, чьи акции будут использоваться, в следующей части курсовой проведу предварительный анализ данных и исключу те компании, которые не подходят.

Далее будет приведена вся теоретическая справка по выбранному критерию и альтернативному. Для дальнейшего анализа проверим их сначала на сгенерированных в Python данных, причем заранее известно, что данные распределены нормально. Только после этого проанализируем котировки компаний индекса S&P 500 и сравним их с модельными. В итоге сделаем вывод об эффективности критерия Дэвида-Хартли-Пирсона и результатах данной работы.

Данная тема является актуальной как для меня, так и для ознакамливающихся с ней, так как по полученным результатам можно будет делать выводы о дальнейшем поведении котировок компаний. Это позволит возможным инвесторам лучше анализировать состояние ценных бумаг на рынке.

# **Предварительный анализ данных**

Для проверки гипотезы о нормальности логарифмической доходности, как уже было обозначено выше, я выбрала котировки акций компаний индекса S&P 500 в период с 01.01.2015 по 01.01.2022 гг. Но в этот индекс входит более 500 компаний, поэтому для удобства дальнейшего анализа я выбрала 18 из них, наиболее меня интересующих. Для этого использовала сайт <https://www.slickcharts.com/sp500>. Для удобства также представлю название тикера каждой компании.

Таблица 1. Названия компаний

|  |  |
| --- | --- |
| Тикер | Название Компании |
| AAPL | Apple Inc. |
| MSFT | Microsoft Corporation |
| AMZN | Amazon.com Inc. |
| TSLA | Tesla Inc. |
| NVDA | NVIDIA Corporation |
| JNJ | Johnson & Johnson |
| PG | Procter & Gamble Company |
| PFE | Pfizer Inc. |
| DIS | Walt Disney Company |
| CSCO | Cisco Systems Inc. |
| KO | Coca-Cola Company |
| PEP | PepsiCo Inc. |
| ADBE | Adobe Inc. |
| INTC | Intel Corporation |
| MCD | McDonald’s Corporation |
| NFLX | Netflix Inc. |
| NKE | NIKE Inc. |
| PM | Philip Morris International Inc. |

Для дальнейшей работы с котировками нужно проверить, все ли компании подходят нам для работы с ними. Сперва узнаем, все ли компании торговались нужное количество дней (> 240) в исследуемый период. Вывела количество торговых дней в таблицу, используя программу “Анализ котировок.ipynb”. В ней я делаю следующее: открываю файл с данными о торгах для каждой компании, а затем, группируя по годам, считаю количество строк в файле, то есть количество дней, в которые акции торговались.

Таблица 2. Количество торговых дней

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ticker | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 |
| US1.AAPL | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.ADBE | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.AMZN | 252 | 252 | 251 | 251 | 252 | 252 | 252 |
| US1.CSCO | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.DIS | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.INTC | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.JNJ | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.KO | 252 | 252 | 251 | 251 | 252 | 252 | 252 |
| US1.MCD | 0 | 0 | 0 | 0 | 202 | 253 | 252 |
| US1.MSFT | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.NFLX | 252 | 252 | 251 | 251 | 252 | 252 | 252 |
| US1.NKE | 252 | 252 | 251 | 251 | 252 | 253 | 252 |
| US1.NVDA | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.PEP | 252 | 252 | 251 | 251 | 252 | 253 | 252 |
| US1.PFE | 252 | 252 | 251 | 251 | 252 | 252 | 252 |
| US1.PG | 252 | 252 | 251 | 251 | 252 | 251 | 252 |
| US1.PM | 252 | 252 | 251 | 251 | 252 | 253 | 252 |
| US1.TSLA | 252 | 252 | 251 | 251 | 252 | 252 | 252 |

По таблице видно, что компания McDonald’s Corporation начала торговать своими акциями только в 2019 году, поэтому исключаем ее из исследуемых данных. Все остальные ячейки таблицы больше или равны 251, поэтому ничего больше убирать на этом этапе не нужно.

Теперь нужно посмотреть на максимальные относительные изменения цен для каждой компании. Сгенерируем таблицы с такими значениями на каждую компанию для каждого года используя программу “Относительные изменения цен.ipynb”. Делаю я это следующим способом:

1. Для каждого дня, кроме первого, считаю значение столбца “Changes” по формуле: . Это относительные изменения цен.
2. Для каждого года и для каждого тикера ищу минимальное и максимальное значение столбца “Changes” и вставляю их в таблицы.

Таблица 3. Относительные скачки цен вверх

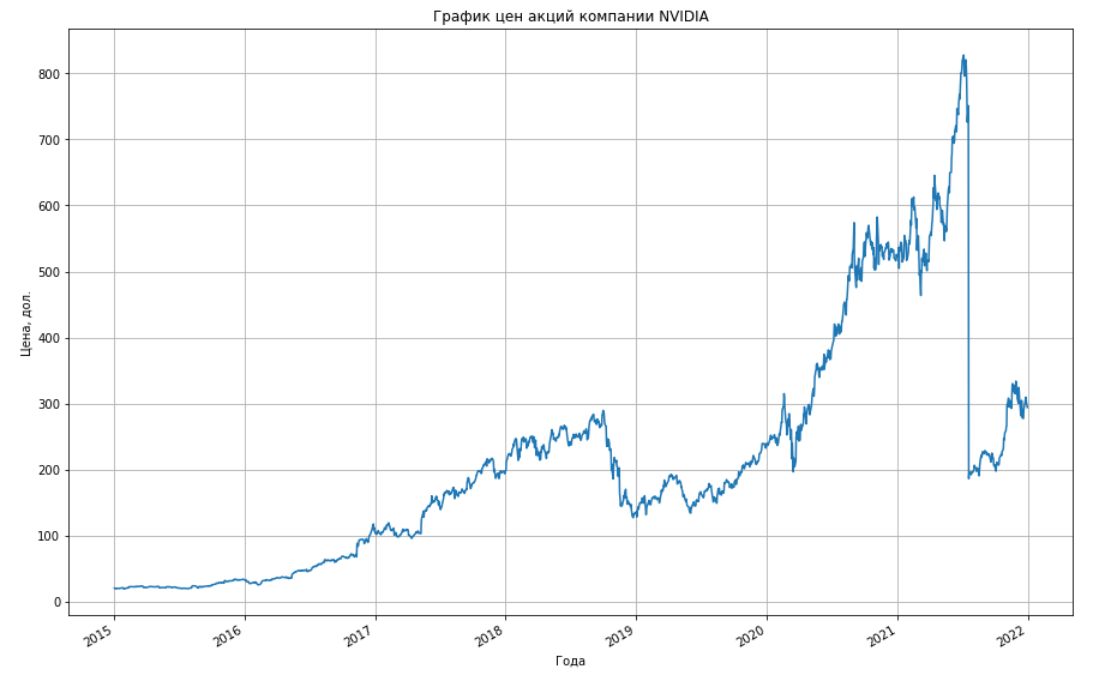
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ticker | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 |
| US1.AAPL | 5,73 | 6,51 | 6,14 | 7,17 | 6,86 | 12 | 5,36 |
| US1.ADBE | 3,77 | 7,08 | 12,24 | 9,5 | 5,19 | 17,47 | 4,63 |
| US1.AMZN | 14,15 | 9,33 | 13,21 | 10,22 | 5,11 | 7,99 | 4,74 |
| US1.CSCO | 9,47 | 9,69 | 5,21 | 5,83 | 6,68 | 12,54 | 4,08 |
| US1.DIS | 7,67 | 3,07 | 4,75 | 5,64 | 11,5 | 14,52 | 6,35 |
| US1.INTC | 6,38 | 3,52 | 7,3 | 10,53 | 8,2 | 19,13 | 6,97 |
| US1.JNJ | 4,79 | 4,86 | 3,5 | 3,55 | 2,98 | 7,9 | 2,82 |
| US1.KO | 2,89 | 2,46 | 1,85 | 2,96 | 6,03 | 6,37 | 2,65 |
| US1.MSFT | 10,85 | 5,74 | 6,44 | 7,62 | 4,69 | 14,31 | 4,25 |
| US1.NFLX | 18,19 | 19 | 13,56 | 10,23 | 9,73 | 11,58 | 16,74 |
| US1.NKE | 8,85 | 3,88 | 10,94 | 11,2 | 4,63 | 15,16 | 15,57 |
| US1.NVDA | 13,86 | 29,81 | 17,89 | 9,37 | 7,19 | 17,52 | 12,02 |
| US1.PEP | 3,14 | 2,75 | 1,67 | 4,76 | 3,76 | 12,96 | 3,29 |
| US1.PFE | 3,89 | 7,1 | 3,28 | 4,73 | 3,14 | 9,29 | 10,78 |
| US1.PG | 3,58 | 3,51 | 3,73 | 8,75 | 4,87 | 12,03 | 3,23 |
| US1.PM | 8,72 | 3,17 | 3 | 4,23 | 8,12 | 10,16 | 4,2 |
| US1.TSLA | 11,2 | 8,88 | 7,26 | 17,24 | 17,55 | 19,99 | 19,66 |

Таблица 4. Относительные скачки цен вниз

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ticker | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 |
| US1.AAPL | -6,11 | -6,55 | -3,84 | -6,57 | -10,03 | -74,21 | -4,13 |
| US1.ADBE | -5,85 | -8,28 | -6,16 | -8,04 | -4,59 | -14,52 | -10,22 |
| US1.AMZN | -6,29 | -7,85 | -3,56 | -7,84 | -5,38 | -7,94 | -7,46 |
| US1.CSCO | -5,93 | -5,07 | -7,21 | -5,35 | -8,69 | -11,19 | -5,6 |
| US1.DIS | -9,21 | -5,23 | -4,42 | -5,33 | -4,98 | -12,99 | -7,01 |
| US1.INTC | -4,82 | -9,22 | -3,52 | -8,63 | -8,96 | -17,83 | -11,69 |
| US1.JNJ | -3,25 | -3,28 | -3,18 | -10,14 | -6,22 | -7,18 | -3,52 |
| US1.KO | -2,88 | -4,83 | -2,08 | -4,03 | -8,42 | -9,66 | -3,85 |
| US1.MSFT | -9,31 | -7,2 | -3,79 | -5,34 | -3,72 | -14,84 | -3,63 |
| US1.NFLX | -86,05 | -13,13 | -5,57 | -9,4 | -10,32 | -11,73 | -7,43 |
| US1.NKE | -50,95 | -5,03 | -7,05 | -6,81 | -6,66 | -11,67 | -6,21 |
| US1.NVDA | -7,42 | -6,88 | -9,28 | -18,82 | -13,86 | -18,15 | -75,23 |
| US1.PEP | -4,65 | -3,72 | -2,03 | -3,78 | -2,8 | -11,59 | -2,77 |
| US1.PFE | -4,08 | -3,05 | -1,82 | -5,25 | -6,36 | -8,02 | -5,11 |
| US1.PG | -3,94 | -3,34 | -3,72 | -4,25 | -3,85 | -8,61 | -3,97 |
| US1.PM | -3,28 | -4,1 | -3,86 | -15,56 | -7,7 | -12,46 | -4,74 |
| US1.TSLA | -8,86 | -10,45 | -7,09 | -13,9 | -13,67 | -77,49 | -11,96 |

По таблице видно, что максимальный скачок цены вверх был у тикера NVDA. А максимальные скачки цены вниз наблюдаются у тикеров AAPL, NFLX, NKE, NVDA и TSLA.

Чтобы лучше понять, что в этот период происходило с ценами компаний, построим графики для вышеперечисленных тикеров со значениями котировок по годам. Сделаем это, используя программу “Визуализация подозрительных тикеров.ipynb”. Для этого считаю данные о ценах акций из столбца “Close” для каждой из компаний по годам и визуализирую их с помощью функции plt.plot() библиотеки matplotlib.pyplot.



Риc 1. График цен акций NVIDIA

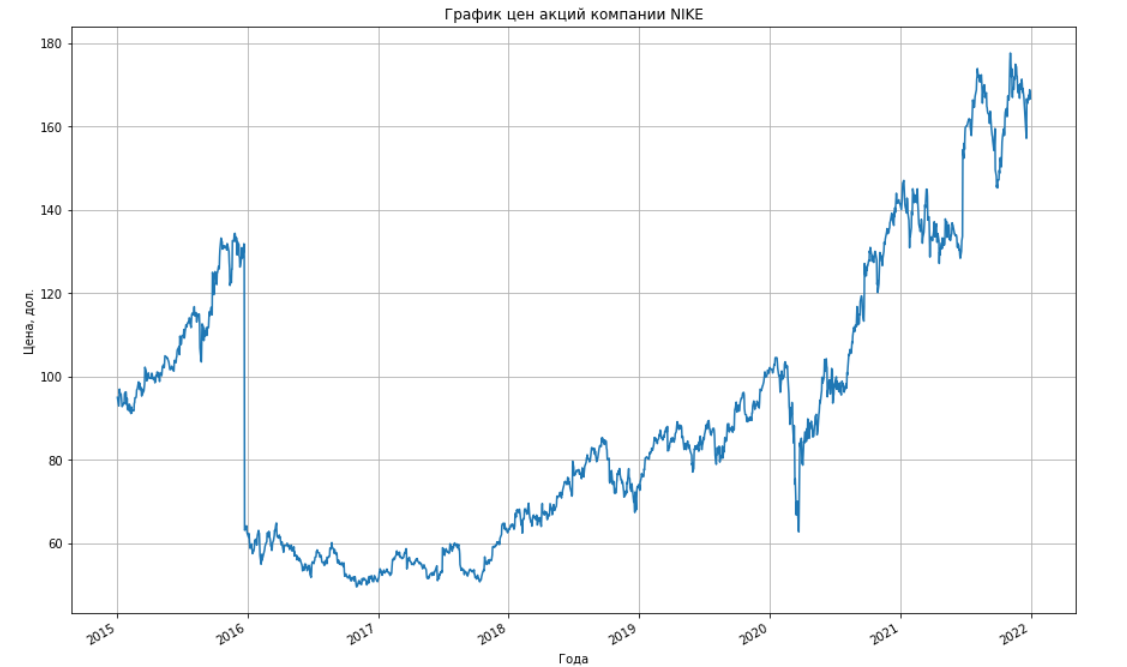
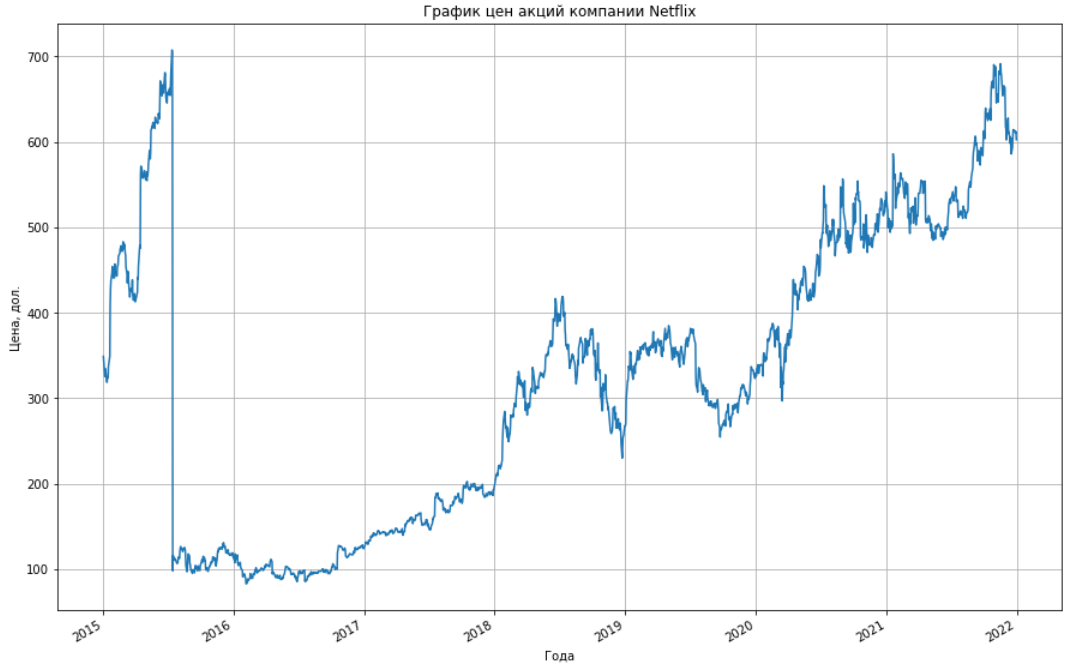


Рис 2. График цен акций NETFLIX Рис 3. График цен акций NIKE

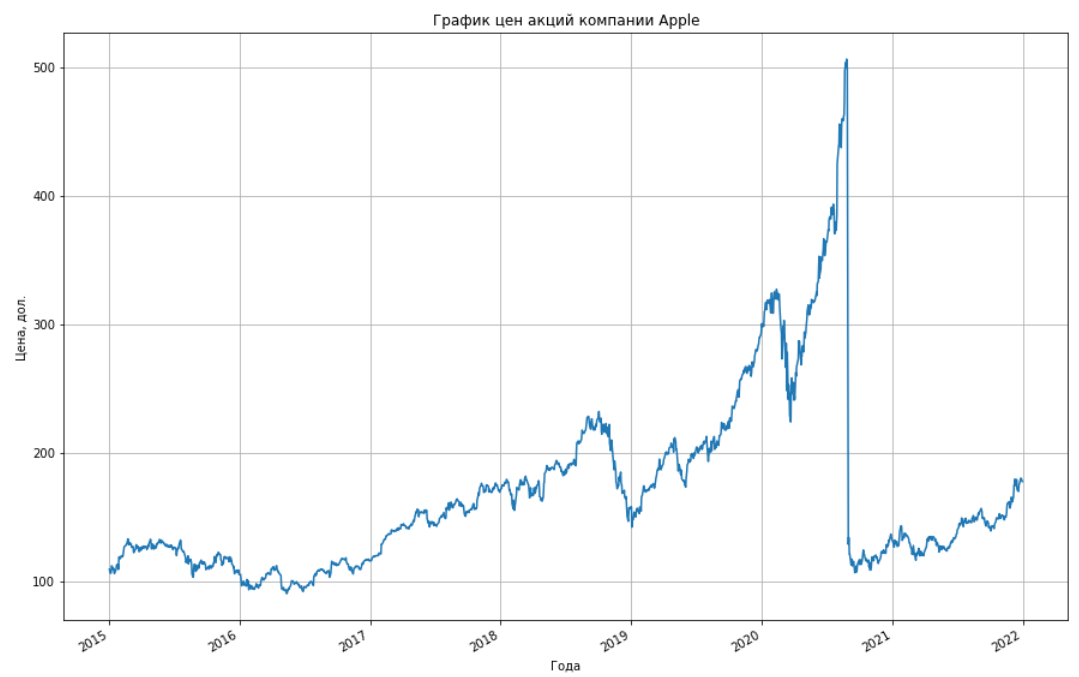
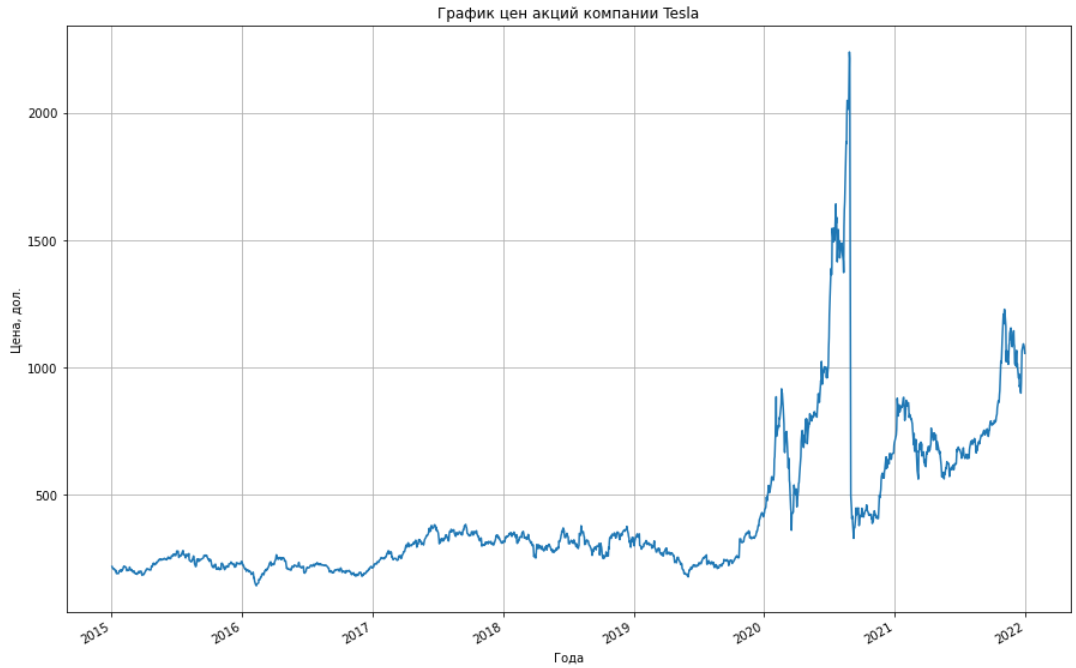


Рис 4. График цен акций Tesla Рис 5. График цен акций Apple

На графике этих трех компаний присутствуют слишком ярко выраженные скачки в 2015 году для Netflix и Nike, в 2020 году для Tesla и Apple, а в 2021 году для NVIDIA.

Разберемся, почему это произошло:

1. У компании Netflix был сплит «7 к 1» 15 июля 2015 года для привлечения новых инвесторов, то есть все прежние владельцы акций получили еще 6, а сами ценные бумаги стали дешевле.
2. 23 декабря 2015 года компания Nike начала программу обратного выкупа акций, из-за чего был осуществлен сплит.
3. Для того, чтобы сделать акции Tesla более доступными и выплатить дивиденды, 31 августа 2020 года был проведен сплит «5 к 1».
4. В 2021 году в июле компания NVIDIA произвела сплит своих акций в соотношении 4 к 1. Это было сделано с целью привлечения новых клиентов и роста спроса на акции компании.
5. К дроблению акций прибегла также и крупнейшая технологическая компания Apple в августе 2020 года. Тогда сплит ценных бумаг составил 4 к 1.

Такие компании нельзя анализировать при дальнейшем анализе, поэтому исключаем их из нашего списка.

# **Теоретическая справка**

## **Основные понятия проверки гипотез**

Теперь, после проверки и фильтра данных, для продолжения работы, нужно ввести основные понятия о гипотезах и методах их проверки.

Вообще, проверка гипотезы о виде генерального распределения – это одна из главных задач математической статистики.

Статистическая гипотеза – это любое предположение о параметрах случайной величины или о виде генерального распределения.

Первое, что стоит сделать – определить саму гипотезу H0. В своей работе я рассматриваю гипотезу о нормальном распределении логарифмической доходности акций.

Гипотеза, которую я определила, не параметрическая, так как ее точные параметры неизвестны. Но она является полностью простой (определенной), потому что предположение о ее справедливости однозначно определяет распределение случайной величины.

Также нужно выдвинуть альтернативную гипотезу H1, которая является отрицанием H0.

Статистический критерий – правило, на основании которого по данным (реализация выборки можно оценить достоверность гипотезы.

В качестве статистического критерия я выбрала критерий Дэвида-Хартли-Пирсона, но о нем будет написано ниже. Именно с помощью него будет приниматься решение, принять или отвергнуть гипотезу.

Применение какого-либо критерия при проверке гипотез может привести к ошибкам двух видов:

1. Ошибка первого рода - это такое событие, когда правильное утверждение считают неверным, то есть отвергается верная гипотеза H0. Вероятность этого события называется уровнем значимости и обозначается как α.
2. Ошибка второго рода происходит, когда ложное утверждение принимается за правильное, то есть отвергается гипотеза H1, когда она на самом деле верна. Вероятность этого события – β. А 1 - β – это мощность критерия.

Поэтому на третьем шаге нужно выбрать значение уровня значимости, с помощью него в дальнейшем будет сделан вывод о верности гипотезы.

Отклонение эмпирических значений от гипотетических характеризует статистика ). Тогда Tнабл – значение статистики полученной реализации выборки. Ее значения делятся на критическую область критерия К и дополнение Д.

К определяется следующим образом: , где

PH0(A) – вероятность, что гипотеза верна.

То есть, если значения статистики ϵ Д, то можно предположить, что гипотеза верна. Но это условие не является достаточным.

Для задания области К есть три способа:

1. Правостостороняя критическая область:
2. Левосторонняя критическая область:
3. Двусторонняя критическая область:

Значения с – константы – критические значения статистического критерия.

Теперь необходимо ввести определение P-значения. Это минимальное значение из совокупности значений уровня значимости, при которых нулевая гипотеза H0 отвергается.

,

H0 нужно отклонить, если P-value мало.

Перед тем, как вводить формулы, по которым будет рассчитываться P -значения, отмечу, что wq – 100q% процентная точка статистики критерия.

Итак:

- В 1) случае: гипотеза отклоняется, если Tнабл > wα

- Во 2) случае: гипотеза отклоняется, если Tнабл < w1-α

- В 3) случае: гипотеза отклоняется, если Tнабл ϵ (ꝏ; w1-α/2) (wα/2; ꝏ)

## **Критерий Дэвида-Хартли-Пирсона**

S – стандартное отклонение; R = xmax – xmin – размах выборки

Гипотезу можно счиать верной, если T1(α) < T < T2(α). То есть критическая область для данного критерия является двусторонней.

В учебниках приводятся критические границы критерия, но я не буду ими пользоваться, так как буду находить значения p-value вручную, используя таблицу квантилей.

Данный график показывает зависимость распределений статистики T от объёма выборки при справедливости проверяемой гипотезы.

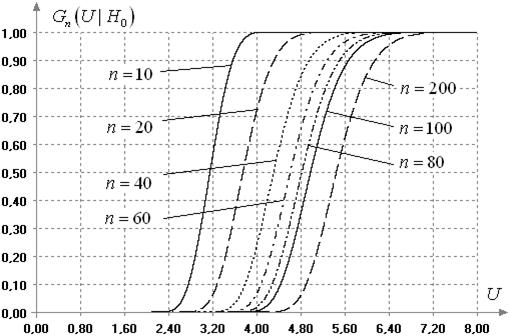


Рис 6. Зав-ть статистики от выборки

## **Критерий Колмогорова**

Также в данной работе будет рассмотрен критерий согласия Колмогорова, в качестве вспомогательного критерия. Я выбрала именно его, так как гипотеза, которую я исследую на достоверность, является простой, а распределение – нормальным. Он будет использоваться во время проверки нормальности распределения P-значений.

где – теоретическая, а эмпирическая функция распределения.

При работе с данным критерием нужно сделать следующее:

1. Проверить, является ли n > 20
2. Вычислим значения границ
3. Посчитаем выборочное значение статистики
4. В данном случае H0 будет отклоняться, если Dn набл > Dn α, проверим это неравенство.

# **Проверка гипотезы для модельных данных**

Тестирование гипотезы на модельных данных является вторым шагом в процессе проверки гипотезы на достоверность. Оно начинается с генерирования 100000 выборок, состоящих из 251 элемента, распределенных по нормальному закону с параметрами и обозначу, что уровень значимости для этой части работы равен 0,05. Код данной программы приведен в файле “Проверка гипотезы на модельных и реальных данных.ipynb”.

Для каждой выборки посчитаю, чему равно значение статистики, используя формулу статистики для критерия Дэвида-Хартли-Пирсона:

S – стандартное отклонение; R = xmax – xmin – размах выборки.

Так, я получила 1000000 значений статистики (список T\_list), теперь нужно построить таблицу 999 квантилей. Уточню, что квантиль — это такое число, которое больше случайно заданного числа с заданной вероятностью. Например, квантиль 0,5 – это медианное значение всей выборки. В своей работе их значения я нахожу с помощью встроенной функции библиотеки numpy np.quantile(), нахожу значения квантилей.

Полная версия таблицы представлена в файле “quantiles.csv”.

Изображение выглядит как текст

Автоматически созданное описание

Рис 7. Значения квантилей

Очевидно, что сделать вывод о том, как распределены p-значения нельзя, если их просто вывести. Данные значения нужно как-то сгруппировать для наглядности. Существует теорема о том, что гипотеза не отвергается, если p-значения распределены равномерно. Поэтому для того, чтобы делать вывод о том, стоит ли принять гипотезу на модельных данных, нужно построить гистограмму 10000 p-values.

Для лучшего понимания, еще раз приведу определение p-значений:

p-value – это наименьшая величина уровня значимости, при котором нулевая гипотеза H0 отвергается, то есть

,

где

Посчитаем эти значения вручную с помощью критерия Дэвида-Хартли-Пирсона так:

1. Как и для квантилей, для каждой выборки нахожу значение статистики.
2. Для моего критерия используется двусторонняя критическая область, поэтому p-значения я должна рассчитать дважды:

Для каждой выборки я сначала перебираю все квантили, нахожу кол-во квантилей, значение которых больше статистики и делю его на общее количество квантилей, то есть на 999.

Затем считаю кол-во квантилей, значение которых меньше критической области и тоже делю это число на 999.

В итоге, нахожу минимум между ними и умножаю его на 2.

1. Теперь у меня есть список pvalues.
2. Гистограмму p-values строю с помощью встроенной функции библиотеки matplotlib.pyplot hist(), в которую передаю список pvalue.

На ней по оси x расположены сами p-значения, а на оси y количество их появлений. То есть этот график показывает, что все значения p-value встречаются примерно одинаковое количество раз.

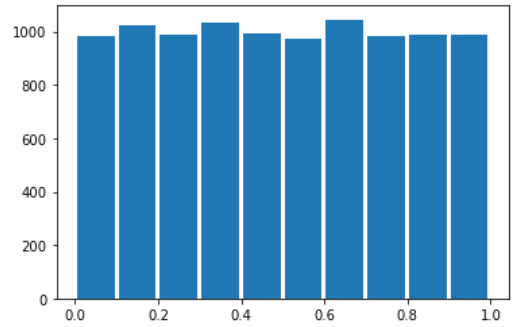


Рис 8. p-значения, модельные данные

Теперь с помощью теста Колмогорова проверю, насколько данное распределение соответствует равномерному. Для этого воспользуюсь встроенной функцией библиотеки scipy.stats ks\_test(). Она возвращает два значения: статистику и p-значение, нас интересует второе, выведем его. P-value по данному тесту, равно 0,5908. Известно, что гипотеза не отвергается, если Так как 0,5908 > 0,05 гипотеза принимается.

# **Выбор альтернативной гипотезы и оценка мощности критерия**

Чтобы понять, какую гипотезу можно выбрать как альтернативную, я перебрала основные распределения математической статистики и вывела графики их плотностей распределения с помощью функции библиотеки scipy.stats pdf(). Как самые похожие на нормальное распределение, были выбраны следующие распределения:

1. Распределение Коши
2. Распределение Стьюдента с 5 и 15 степенями свободы
3. Логистическое распределение

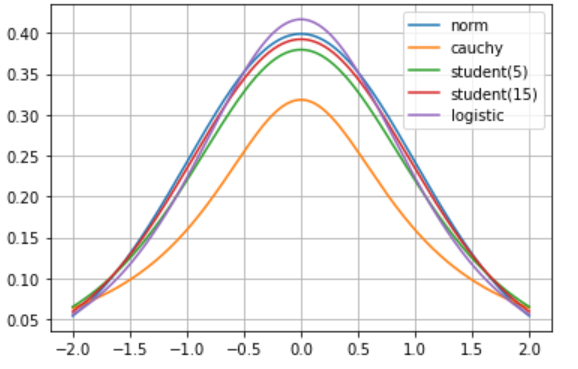


Рис 9. Плотности распределений

Теперь рассчитаем значения мощности критерия, учитывая, что оно равняется 1 – вероятность принятия неверной гипотезы за верную. Поэтому для 10000 выборок из 251 элемента посчитаем, какую часть от общего объема занимают значения, которые меньше α или больше 1-α.

Мощности критерия я считаю по следующему алгоритму:

1. Считаю значения квантилей и p-values по алгоритмам, описанным выше.
2. Затем, для каждого уровня значимости α, считаю сколько p-значений входят в интервал: [0; α) (1- α; 1].
3. Делю полученное значение на количество p-значений. Получившееся число является мощностью критерия.

Таблица 5. Мощности критерия для разных распределений

|  |  |  |  |
| --- | --- | --- | --- |
|  | **0,01** | **0,05** | **0,1** |
| Р-ие Коши | 0.0186 | 0.0999 | 0.1995 |
| Логистическое р-ие | 0.0231 | 0.1014 | 0.1994 |
| Р-ие Стьюдента с 5 степенями свободы | 0.0188 | 0.0918 | 0.1965 |
| Р-ие Стьюдента с 15 степенями свободы | 0.0202 | 0.0972 | 0.2025 |

По таблице видно, что мощность критерия повышается с увеличением уровня значимости. Но в целом данный показатель достаточно низкий, что говорит о том, что выбранный критерий недостаточно четкий и подходящий для такого вида работы.

# **Проверка гипотезы для реальных данных**

Выполнив все предыдущие шаги, можно начинать проверку гипотезы о нормальности логарифмической доходности на реальных данных. Используя все ту же программу “Проверка гипотезы на модельных и реальных данных.ipynb”, построим таблицу p-значений, используя критерий Дэвида-Хартли-Пирсона. Для более четкого понимания, распределены ли данные нормально, была построена гистограмма p-значений. Алгоритм моих действий был следующим:

1. Нашла все значения логарифмических доходностей компаний по формуле: , где Si – цена акции в i-ый день, а Si-1 – цена акции в i-1 день.
2. Теперь для каждого списка со значениями логаримических доходностей, по которому мы находим значений p-values используя тот же алгоритм, что и в части с ислледованием модельных данных.
3. Полученные данные заносим в таблицу, а затем по ним строим гистограмму, используя встроенную функцию бибилиотеки matplotlib.pyplot hist().

Таблица 6. p-значения для реальных данных

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ticker | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 |
| US1.ADBE | 0.09 | 0.0 | 0.0 | 0.0 | 0.106 | 0.0 | 0.0 |
| US1.AMZN | 0.0 | 0.0 | 0.0 | 0.0 | 0.006 | 0.092 | 0.0 |
| US1.CSCO | 0.0 | 0.0 | 0.0 | 0.062 | 0.0 | 0.0 | 0.0 |
| US1.DIS | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| US1.INTC | 0.008 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| US1.JNJ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.03 |
| US1.KO | 0.14 | 0.0 | 0.034 | 0.008 | 0.0 | 0.002 | 0.054 |
| US1.MSFT | 0.0 | 0.0 | 0.0 | 0.008 | 0.064 | 0.0 | 0.468 |
| US1.PEP | 0.0 | 0.0 | 0.432 | 0.008 | 0.002 | 0.0 | 0.132 |
| US1.PFE | 0.208 | 0.0 | 0.012 | 0.0 | 0.0 | 0.0 | 0.0 |
| US1.PG | 0.004 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| US1.PM | 0.0 | 0.016 | 0.002 | 0.0 | 0.0 | 0.0 | 0.0 |

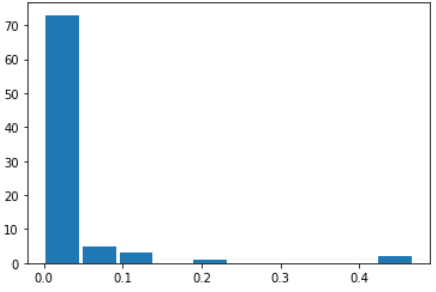


Рис 10. Гистограмма p-values

На графике ясно видно, что p-values не распределены равномерно. Более того, можно заметить, что большая часть p-значений меньше уровня значимости, поэтому принятая гипотеза отвергается. Доходности распределены не нормально.

Таблица 7. Доля проверок, в которых гипотеза принималась

|  |  |  |  |
| --- | --- | --- | --- |
| **Уровень значимости (α)** | **1%** | **5%** | **10%** |
| Частота принятия гипотезы | 0.179 | 0.131 | 0.071 |

Таблица показывает, в скольких процентах случаев принималась гипотеза. Чтобы найти это число, я считаю количество p-значений, входящих в интервал [α; 1- α] и делю его на общее количество p-значений. Значение в таблице уменьшается, из-за того, что количество ошибок второго рода увеличивается при уменьшении уровня значимости.

Также приведу средние значения p-values по компаниям и по годам. Их я считаю так:

1. Сначала нахожу p-значения по вышеописанному алгоритму.
2. Затем нахожу их среднее значение с помощью встроенной функции библиотеки pandas DataFrame.mean().

Таблица 8. p-values по годам

|  |  |
| --- | --- |
| Year | p-value |
| 2015 | 0,038 |
| 2016 | 0,001 |
| 2017 | 0,040 |
| 2018 | 0,007 |
| 2019 | 0,015 |
| 2020 | 0,008 |
| 2021 | 0,057 |

Таблица 9. p-значения по компаниям

|  |  |
| --- | --- |
| Ticker | p-value |
| US1.ADBE | 0,028 |
| US1.AMZN | 0,014 |
| US1.CSCO | 0,009 |
| US1.DIS | 0 |
| US1.INTC | 0,001 |
| US1.JNJ | 0,004 |
| US1.KO | 0,034 |
| US1.MSFT | 0,077 |
| US1.PEP | 0,082 |
| US1.PFE | 0,031 |
| US1.PG | 0,001 |
| US1.PM | 0,003 |

Для компании с максимальным средним значением - Pepsi, выведу график зависимости p-значений от года. Для этого я беру p-значения для тикера US1.PEP и визуализирую их по годам с помощью функции plt.plot().

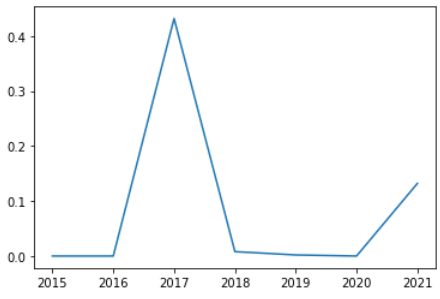


Рис 11. p-vales для тикера US1.PEP

Этот скачок вызван ростом чистой прибыли компании в 2017 году.

Для точности проанализирую то, как часто принималась гипотеза для разных компаний на разных уровнях значимости. Это значение рассчитывается так же, как рассчитывались данные для таблицы 7, только теперь считаем для каждой компании по отдельности.

По таблице видно, что почти для всех данных гипотеза не принимается, что говорит о не нормальности распределения доходностей.

Таблица 10. Доля проверок, в которых гипотеза принималась по компаниям

|  |  |  |  |
| --- | --- | --- | --- |
| **TIcker** | **alpha=0.01** | **alpha=0.05** | **alpha=0.1** |
| US1.ADBE | 0.2857142857142857 | 0.2857142857142857 | 0.1428571428571429 |
| US1.AMZN | 0.1428571428571429 | 0.1428571428571429 | 0.0 |
| US1.CSCO | 0.1428571428571429 | 0.1428571428571429 | 0.0 |
| US1.DIS | 0.0 | 0.0 | 0.0 |
| US1.INTC | 0.0 | 0.0 | 0.0 |
| US1.JNJ | 0.1428571428571429 | 0.0 | 0.0 |
| US1.KO | 0.4285714285714286 | 0.2857142857142857 | 0.1428571428571429 |
| US1.MSFT | 0.2857142857142857 | 0.2857142857142857 | 0.1428571428571429 |
| US1.PEP | 0.2857142857142857 | 0.2857142857142857 | 0.2857142857142857 |
| US1.PFE | 0.2857142857142857 | 0.1428571428571429 | 0.1428571428571429 |
| US1.PG | 0.0 | 0.0 | 0.0 |
| US1.PM | 0.1428571428571429 | 0.0 | 0.0 |

Теперь, для того чтобы убедиться в верности выводов, проверим и значения индекса на нормальность. Для этого скачаем с сайта finam.ru данные для индекса S&P 500 для периода с 01.01.2015 по 01.01.2022 гг.

Для расчета данного индекса применяется следующая формула:

где Pi – цена i-ой акции в индексе, Qi – кол-во i-ых акций, а Divisor – нормирующий фактор, равный 8,3 млрд (это изменяющейся параметр, который помогает быть значению index level постоянным, несмотря на уходящие и приходящие в индекс компании).

Далее, как и для всех тикеров при анализе реальных данных выше, построим для значений индекса такую же таблицу и гистограмму p-values, чтобы проверить нормальность распределения index level.

Изображение выглядит как стол

Автоматически созданное описание

Рис 12. P-значения index level по годам

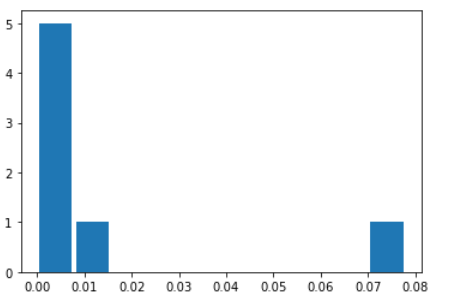


Рис 13. Гистограмма p-значений index level

По таблице видно, что в 2021 году p-значение больше уровня значимости, и для этого года можно было бы принять нулевую гипотезу. Но так как мне важно, чтобы гипотеза была верной для всех данных и очевидно, что p-values для индекса распределены неравномерно и сгруппированы около нуля, я ее не принимаю. Значения индекса не распределены нормально и для самого индекса нулевая гипотеза также отвергается.

# **Заключение**

Итак, в данной работе были достигнуты все цели, описанные во введении: проведен предварительный анализ данных, написана теория, гипотеза была проверена на модельных и реальных данных.

Стоит отметить, что уже на модельных данных критерий Дэвида-Хартли-Пирсона давал достаточно низкое значение на тесте Колмогорова-Смирнова, а во время выбора альтернативной гипотезы выяснилось, что значения его мощности также низкие. Однако он удобен простотой формулы для вычисления статистики.

В своей работе я, так же, как и студенты до меня, показала, что логарифмические доходности котировок акций индекса S&P 500 не распределены нормально. Гипотеза отвергалась как за весь период по всем компаниям, так и по годам.

В результате проведённой работы я изучила и программно описала все требуемые для анализа алгоритмы, изучила методику проверки гипотез и применила на практике полученные знания.

Новизна работы состоит в том, что большая часть кода была написана самостоятельно и на нормальность распределения были исследованы не только акции компаний индекса, но и значения index level. Также я заметила, что при подсчете p-значений ранее использовались встроенные функции Python, я же сделала это вручную и для модельных, и для реальных данных.

# **Литература**

1. А. В. Браилов, В. И. Глебов, С. Я. Криволапов, П. Е. Рябов - Теория вероятностей и математическая статистика – 2016 г – 414 с.
2. Кобзарь А. И. - Прикладная математическая статистика. Для инженеров и научных работников. – 2006 г – 814 с.
3. Лемешко Б.Ю., Рогожников А.П. “Исследование особенностей и мощности некоторых критериев нормальности” // Метрология. – 2009. – № 4. – С. 3-24
4. “How to interpret p-value of Kolmogorov-Smirnov test (python)?”. [Электрон. ресурс]. URL: <https://stats.stackexchange.com/questions/57885/how-to-interpret-p-value-of-kolmogorov-smirnov-test-python> (дата обращения 15.04.2022)
5. “Netflix Is Splitting Its Stock 7 For 1”. [Электрон. ресурс]. URL: <https://clck.ru/ghbm7> (дата обращения 29.04.2022)
6. “Nike проведет обратный выкуп акций на $12 млрд, сплит акций”. [Электрон. ресурс]. URL: <https://clck.ru/ghbjs> (дата обращения 29.04.2022)
7. “Акции Tesla снова «разделятся» ”. [Электрон. ресурс]. URL: <https://clck.ru/ghbmR> (дата обращения 29.04.2022)

# **Приложение 1**

Изображение выглядит как стол

Автоматически созданное описание

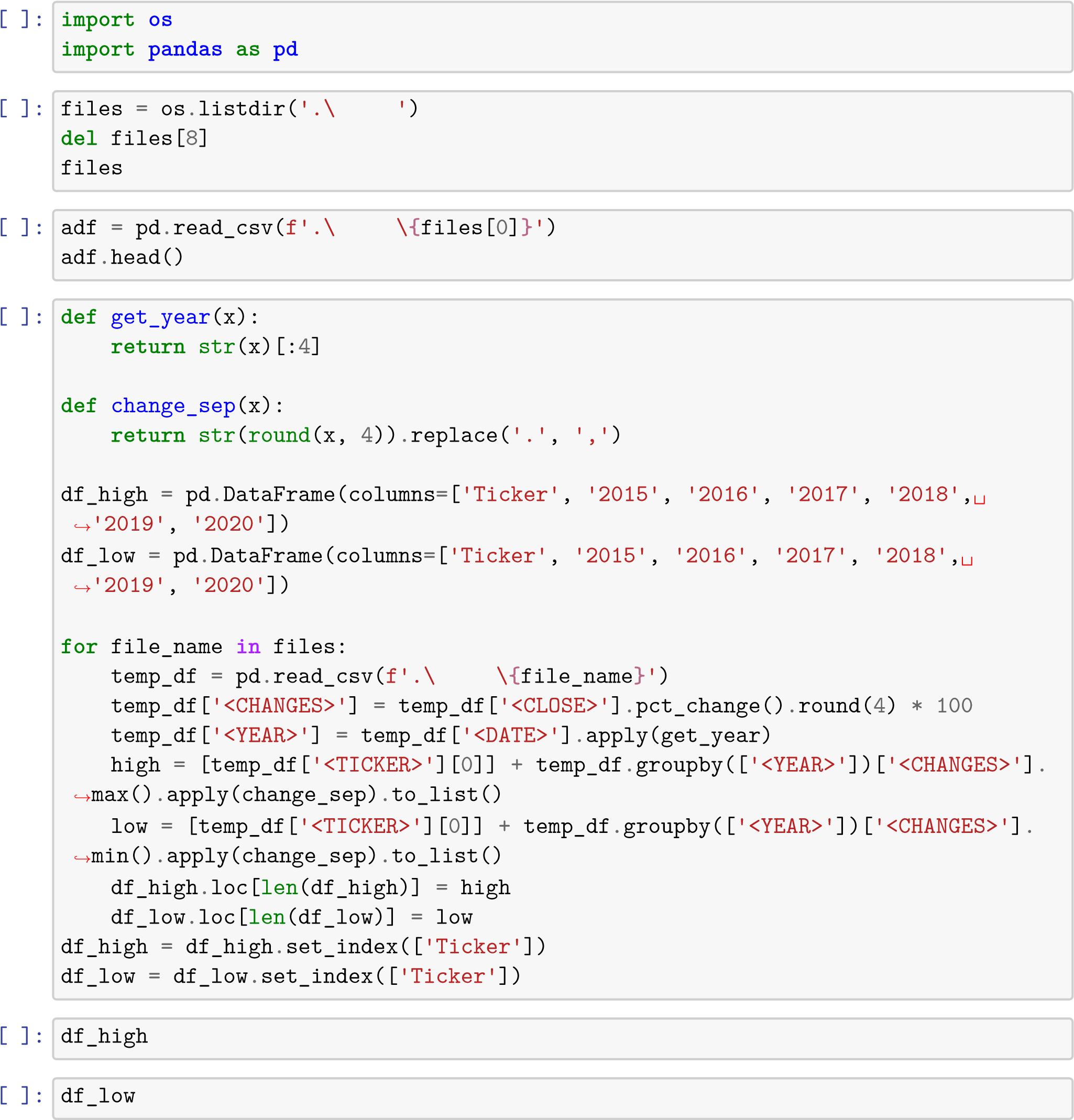
**Приложение 2**

Анализ котировок.ipynb

Изображение выглядит как текст

Автоматически созданное описание

Относительные изменения цен.ipynb



[ ]: df\_high.to\_csv('.\\ (high).csv', sep=';') df\_low.to\_csv('.\\ (low).csv', sep=';')

Визуализация подозрительных тикетов.ipynb

[ ]:

**import**

**matplotlib**

**.**

**pyplot**

**as**

**plt**

**import**

**pandas**

**as**

**pd**

[ ]:

df\_NVDA

=

pd

.

read\_csv(

r

'

.

\

\

US1\_NVDA\_150101\_210101.csv

'

)

df\_NVDA[

'

>

<

DATE

'

]

=

pd

.

to\_datetime(df\_NVDA[

'

<

DATE

>

'

]

,

format

=

"

%

Y

%

m

**%d**

"

)

df\_NVDA

=

df\_NVDA

.

set\_index([

'

DATE

>

<

'

])

df\_NVDA

.

head()

[ ]:

plt

.

figure(

1

,(

15

,

10

))

df\_NVDA[

'

<

CLOSE

>

'

]

.

plot()

plt

.

title(

'

NVIDIA

'

)

plt

.

xlabel(

'

'

)

plt

.

ylabel(

'

, .

'

)

plt

.

grid()

plt

.

show()

[ ]:

df\_NFLX

=

pd

.

read\_csv(

r

'

.

\

\

US1\_NFLX\_150101\_210101.csv

'

)

df\_NFLX[

'

>

<

DATE

'

]

=

pd

.

to\_datetime(df\_NFLX[

'

<

DATE

>

'

]

,

format

=

"

%

Y

%

m

**%d**

"

)

df\_NFLX

=

df\_NFLX

.

set\_index([

'

DATE

>

<

'

])

df\_NFLX

.

head()

[ ]:

plt

.

figure(

1

,(

15

,

10

))

df\_NFLX[

'

<

CLOSE

>

'

]

.

plot()

plt

.

title(

'

Netflix

'

)

plt

.

xlabel(

'

'

)

plt

.

ylabel(

'

, .

'

)

plt

.

grid()

plt

.

show()

[ ]:

df\_NKE

=

pd

.

read\_csv(

r

'

.

\

\

US1\_NKE\_150101\_210101.csv

'

)

print

(

df\_NKE

.

dtypes)

df\_NKE

.

head()

[ ]:

plt

.

figure(

1

,(

15

,

10

))

df\_NKE[

'

<

CLOSE

>

'

]

.

plot()

plt

.

title(

'

Nike

'

)

plt

.

xlabel(

'

'

)

plt

.

ylabel(

'

, .

'

)

plt

.

grid()

plt

.

show()

[ ]:

df\_TSLA

=

pd

.

read\_csv(

r

'

.

\

\

US1\_TSLA\_150101\_210101.csv

'

)

df\_TSLA[

'

>

<

DATE

'

]

=

pd

.

to\_datetime(df\_TSLA[

'

<

DATE

>

'

]

,

format

=

"

%

Y

%

m

**%d**

"

)

df\_TSLA

=

df\_TSLA

.

set\_index([

'

DATE

>

<

'

])

df\_TSLA

.

head()

[ ]:

plt

.

figure(

1

,(

15

,

10

))

df\_TSLA[

'

<

CLOSE

>

'

]

.

plot()

plt

.

title(

'

Tesla

'

)

plt

.

xlabel(

'

'

)

plt

.

ylabel(

'

, .

'

)

plt

.

grid()

plt

.

show()

Проверка гипотезы на модельных и реальных данных.ipynb

[ ]:

**import**

**scipy**

**.**

**stats**

**as**

**sps**

**import**

**numpy**

**as**

**np**

**import**

**pandas**

**as**

**pd**

norm

=

sps

.

norm(

0

,

1

)

cauchy

=

sps

.

cauchy()

logistic

=

sps

.

logistic()

student5

=

sps

.

t(

5

)

student15

=

sps

.

t(

15

)

**def**

quant

(

distr

=

norm):

T\_list

=

[]

**for**

i

**in**

range

(

100000

):

sample

=

distr

.

rvs(

251

)

R

=

max

(

sample

)

-

min

(

sample

)

std

=

np

.

std(sample)

T

=

R

/

std

T\_list

.

append(T)

quantiles

=

[]

**for**

q

**in**

range

(

1

,

1000

):

quantiles

.

append(np

.

quantile(T\_list, q

/

1000

))

df

=

pd

.

DataFrame(columns

=

[

'

q

'

,

'

x\_q

'

])

**for**

q

**in**

range

(

1

,

1000

):

df

.

loc[

len

(

df

)]

=

[

q

/

1000

, quantiles[q

-

1

]]

df

.

to\_csv(

'

quantiles.csv

'

, index

=

**False**

)

**return**

df, quantiles

df, quantiles

=

quant()

print

(

df

)

[ ]:

*# p\_values --*

**import**

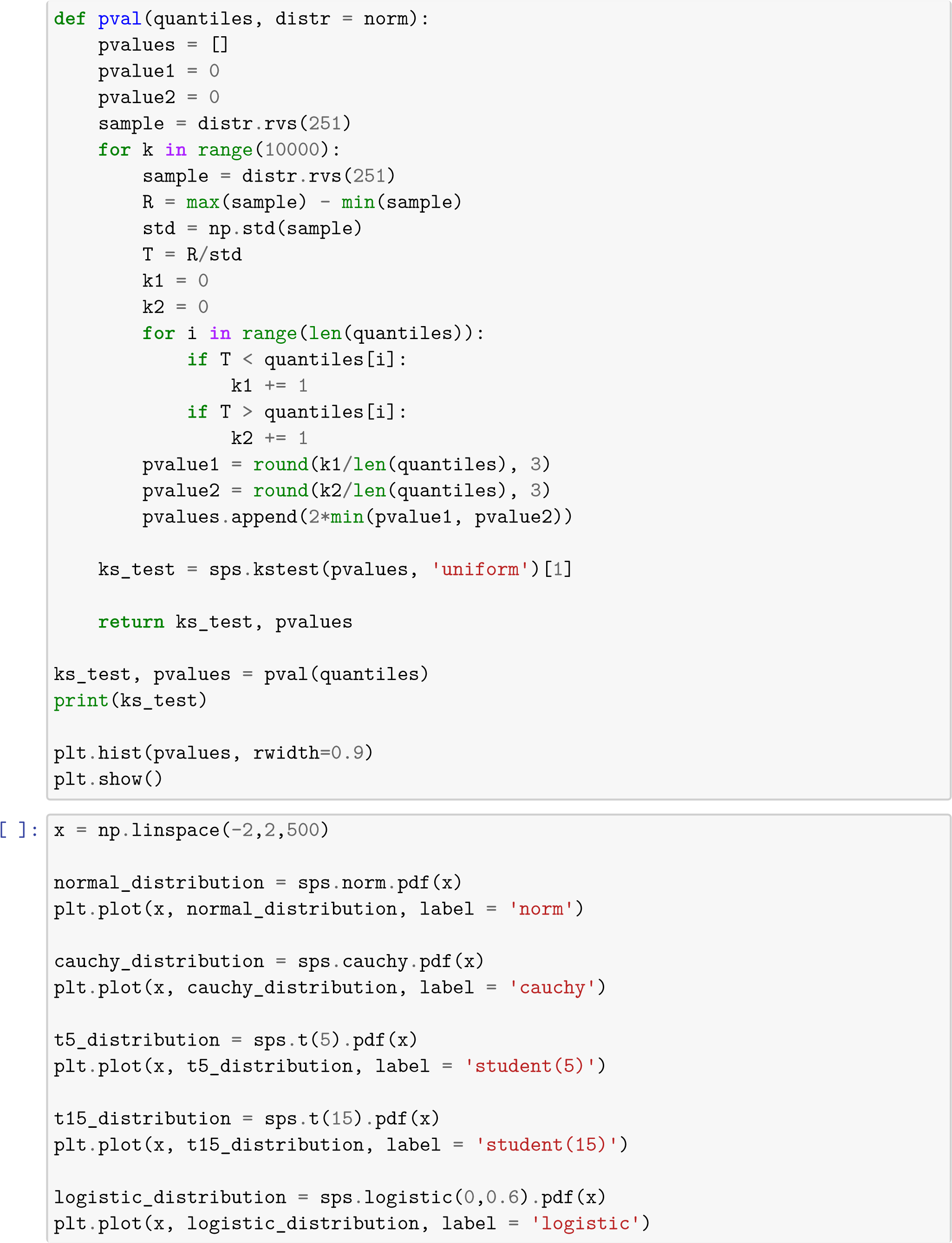
**matplotlib**

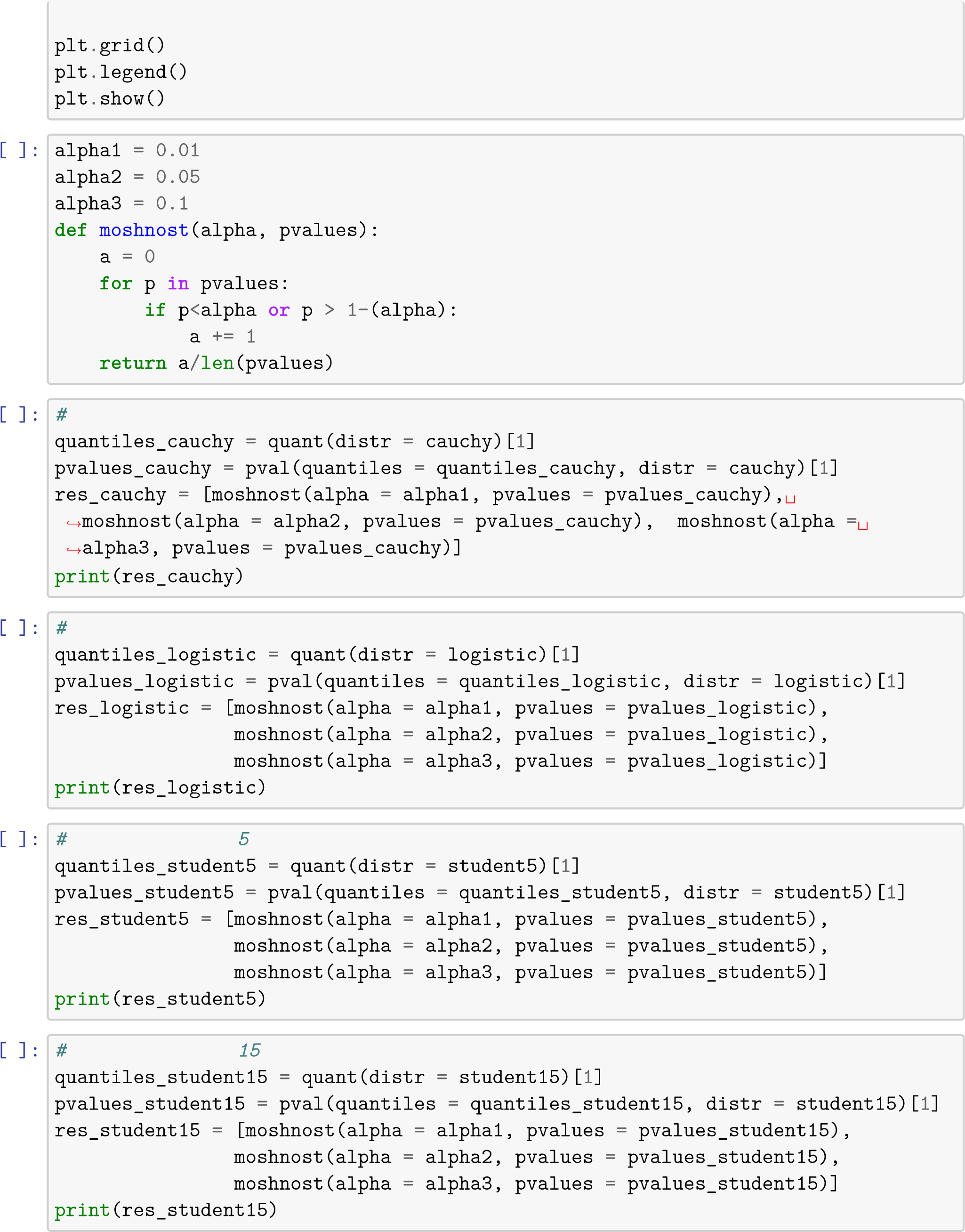
**.**

**pyplot**

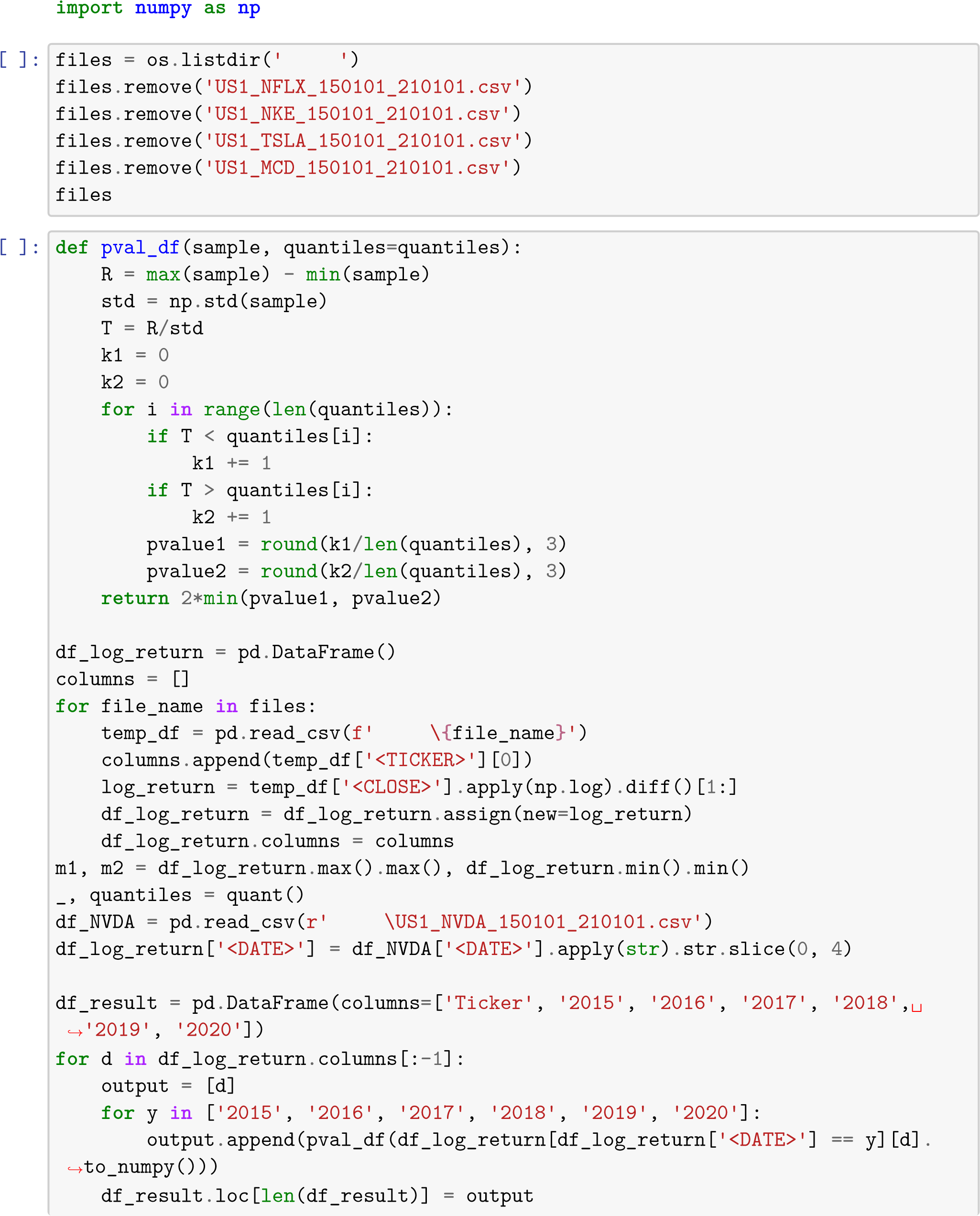
**as**

**plt**

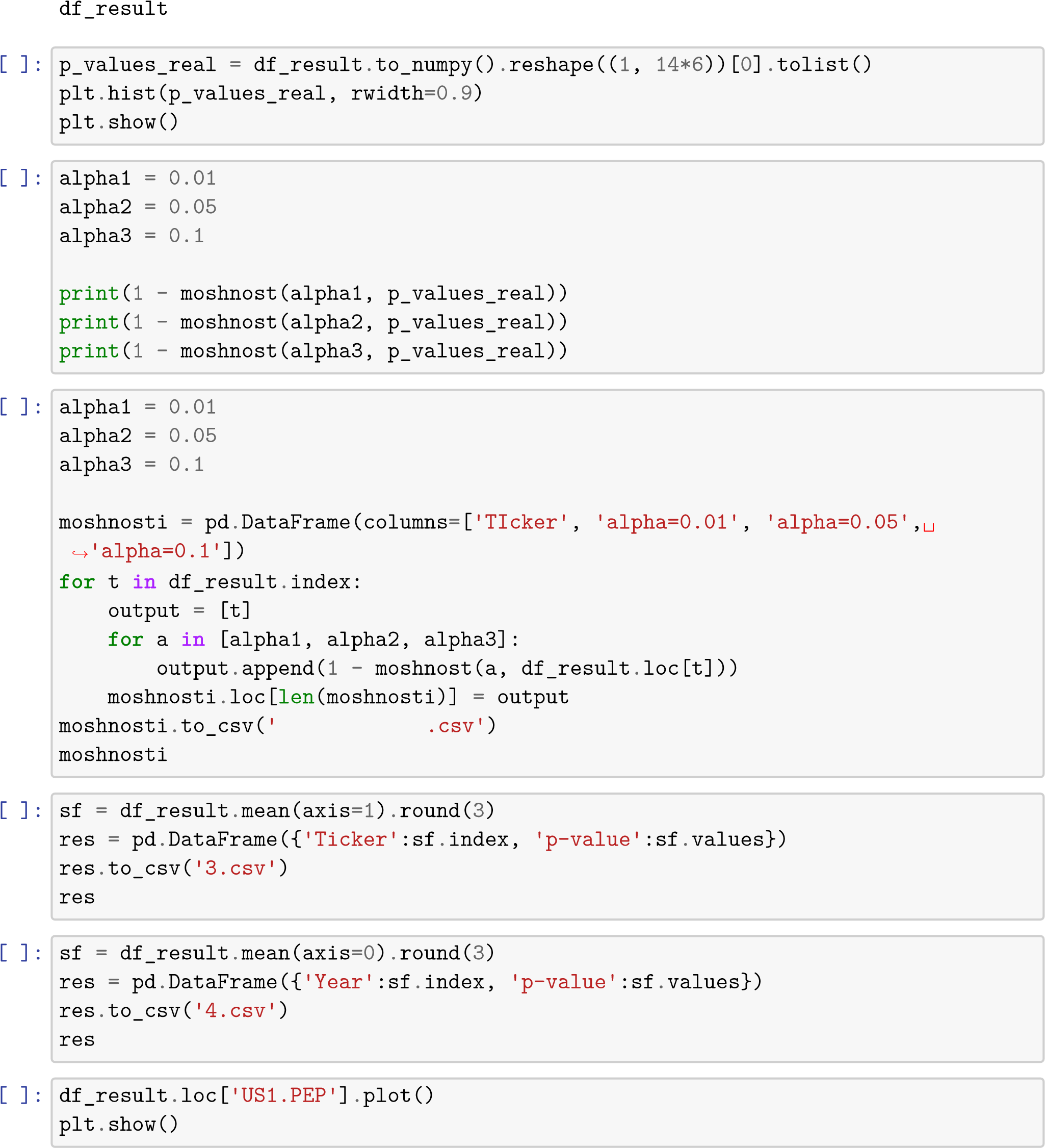




[ ]: **import os import pandas as pd**



df\_result = df\_result.set\_index(['Ticker']) df\_result.to\_csv('1.csv')



**S&P 500**

[ ]:

index\_name

=

os

.

listdir(

'

'

)[

0

]

temp\_df

=

pd

.

read\_csv(

f

'

\

**{**

index\_name

**}**

'

, sep

=

'

;

'

)

temp\_df

.

head()

[ ]: ticker = temp\_df['<TICKER>'][0] log\_return = temp\_df['<CLOSE>'].apply(np.log).diff()[1:] log\_return

[ ]:

log\_df

=

pd

.

DataFrame(log\_return

.

to\_numpy(), columns

=

[

ticker

])

log\_df[

'

<

DATE

>

'

]

=

temp\_df[

'

<

DATE

>

'

]

.

apply(

str

)

.

str

.

slice(

0

,

4

)

log\_df

[ ]:

\_, quantiles

=

quant()

output

=

[

ticker

]

result

=

pd

.

DataFrame(columns

=

[

'

TICKER

>

<

'

,

'

2015

'

,

'

2016

'

,

'

2017

'

,

'

2018

'

,

␣

*,*

→

'

2019

'

,

'

2020

'

])

**for**

y

**in**

[

'

2015

'

,

'

2016

'

,

'

2017

'

,

'

2018

'

,

'

2019

'

,

'

2020

'

]:

output

.

append(pval\_df(log\_df[log\_df[

'

<

DATE

>

'

]

==

y][ticker]

.

to\_numpy()))

result

.

loc[

0

]

=

output

result

=

result

.

set\_index([

'

<

TICKER

>

'

])

result

.

to\_csv(

'

5

.csv

'

)

result

[ ]:

plt

.

hist(result

.

to\_numpy()[

0

]

, rwidth

=

0.9

)

plt

.

show()