ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ФИНАНСВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»   
(ФИНАНСОВЫЙ УНИВЕРСИТЕТ)**

Департамент анализа данных и машинного обучения

**Дисциплина: «Теория вероятностей и математическая статистика»**Направление подготовки: «Прикладная математика и информатика» Профиль: «Анализ данных и принятие решений в экономике и финансах»   
Факультет информационных технологий и анализа больших данных   
Форма обучения очная   
Учебный 2021/2022 год, 4 семестр

Курсовая работа  
на тему:  
 **«Проверка гипотезы о нормальном распределении логарифмической доходности при условии, что накануне значение индикатора RSI находилось в определенном диапазоне.»**

Вид исследуемых данных:   
**Котировки акций компаний, входящих в индекс S&P 100**

Выполнил: студент группы ПМ20-4 Новохатний В. А.

Руководитель: к.э.н. Диденко А.С.

Москва 2022

**Содержание**

[**1.** **Введение** 2](#_Toc104986221)

[**2.** **Предварительный анализ выбранных данных** 3](#_Toc104986222)

[**2.1. Количество торговых дней** 4](#_Toc104986223)

[**2.2. Максимальное относительное изменение цены** 5](#_Toc104986224)

[**3. Теоретическая справка** 7](#_Toc104986225)

[**3.1. Статистическая гипотеза** 7](#_Toc104986226)

[**3.2. Критерий Пирсона** 7](#_Toc104986227)

[**3.3. Критерий Колмогорова** 8](#_Toc104986228)

[**3.4. Индикатор RSI** 8](#_Toc104986229)

[**4. Проверка гипотез на модельных данных** 9](#_Toc104986230)

[**5. Альтернативные гипотезы** 11](#_Toc104986231)

[**6. Проверка гипотезы на реальных данных** 13](#_Toc104986232)

[**7. Заключение** 16](#_Toc104986233)

[**8. Литература** 16](#_Toc104986234)

[**9. Приложения** 16](#_Toc104986235)

# **Введение**

В данной курсовой работе будет проведена проверка гипотезы о нормальном распределение логарифмической доходности при условии, что накануне значение индикатора RSI находилось в определенном диапазоне.

Данная курсовая работа будет состоять из нескольких частей. Для начала проанализируем данные с которыми будем работать и исключим неподходящие. Затем познакомимся с теоретической частью критериев и индикатора RSI. Далее на практике проверим гипотезы на модельных данных, а затем на реальных акциях из индекса S&P 100. В конце сделаем вывод об эффективности исследуемого индикатора.

Расчет данных основан на ценах закрытия акций, которые включены в индекс S&P 100. Даты исследуемого промежутка времени: с 01.01.2012 по 31.12.2021.

Данная работа является актуальной как для самого автора, так и для людей, интересующихся и знакомых с индикатором RSI и темой в целом. По выполнению курсовой работы будут получены результаты о эффективности индикаторы, которые можно сравнить с данными по другим индикаторам.

1. **Предварительный анализ выбранных данных**

Как было сказано ранее, в данной работе будут рассмотрены некоторые акции, входящие в индекс S&P 100 в период с 01.01.2012 по 31.12.2021.

Список компаний, а также информацию о ценах закрытия, открытия и о наибольших и наименьших значениях взял с сайта <https://finance.yahoo.com>.

Акции, которые были взяты для анализа приведены ниже в таблице с соответствующими им тикерами.

|  |  |
| --- | --- |
| Тикер | Компания |
| AAPL | Apple Inc. |
| BCO.HM | BOEING CO |
| COP | ConocoPhillips |
| KO | The Coca-Cola Comp. |
| FDX | FedEx Corporation |
| GM | General Motors Company |
| MA | Mastercard Incorporated |
| INTC | Intel Corp. |
| HPQ | HP Inc. |
| IBM | IBM Corp. |
| HON | Honeywell International Inc. |
| EBAY | eBay Inc. |
| HAL | Halliburton Company |
| MCD | McDonald's Corp. |
| ORCL | Oracle Corp. |

Таблица 1. Список компаний

## **2.1. Количество торговых дней**

Для начала нужно убедиться, что исследуемые акции торговались в заданный период, и они пригодны для дальнейшего анализа. Необходимо, чтобы количество дней по каждой акции было больше 240, также нужно чтобы количество между всеми тикерами за один год были равны. Воспользуемся программой “КОД 1”. Результат представлен в таблице 2.

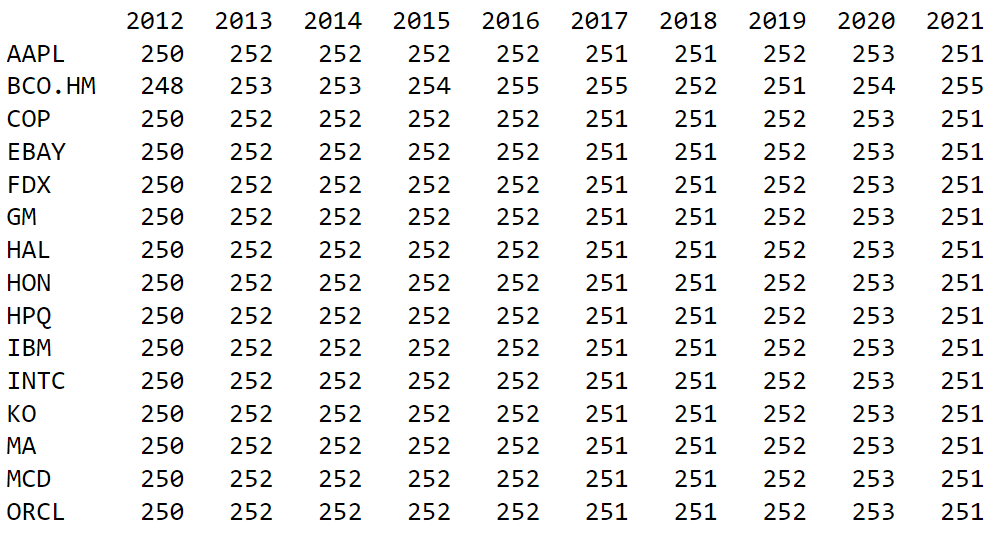


Таблица 2 Количество торговых дней

Заметим, что все компании, кроме “BOEING CO”, имеют одинаковое количество торговых дней в каждом году, также все другие акции удовлетворяют заданным требованиям. Поэтому следует убрать неподходящую компанию из нашей выборки.

## **2.2. Максимальное относительное изменение цены**

Следующим шагом следует рассмотреть максимальные отклонения цен акций за каждый год. Это нужно сделать, чтобы исключить компании с аномально большими скачками цен вверх или вниз. Для того чтобы это посчитать будет использован “КОД 2”. Результаты будут представлены в таблице 3 и 4



AAPL 5.31 3.96 3.72 8.70 3.08 2.67 5.98 3.95 6.26 3.32

COP 2.78 1.93 4.37 7.59 7.76 3.99 6.62 4.43 16.22 5.13

EBAY 4.96 4.19 4.43 6.25 4.01 3.60 5.69 4.34 6.58 7.84

FDX 4.59 4.56 2.61 2.64 4.34 3.29 3.99 3.62 12.98 3.30

GM 7.43 5.32 3.70 5.07 5.82 4.19 5.61 3.93 9.27 5.82

HAL 5.23 6.39 4.64 6.61 11.09 3.89 4.43 8.04 11.99 6.47

HON 3.32 2.70 3.18 3.40 2.77 1.98 3.64 2.39 8.65 3.62

HPQ 4.63 8.81 7.09 10.91 5.35 6.60 5.54 3.50 15.52 5.46

IBM 2.23 2.67 3.33 3.03 2.87 2.65 4.21 5.00 6.00 3.98

INTC 4.01 3.03 5.30 6.63 2.68 2.67 4.98 4.19 12.78 5.79

KO 2.02 2.34 2.74 2.10 2.46 2.67 2.86 2.30 7.15 2.25

MA 3.92 4.47 6.10 3.15 9.31 1.89 5.93 3.36 9.80 4.09

MCD 2.98 2.06 3.06 3.95 2.58 2.58 4.31 1.86 10.58 3.19

ORCL 3.09 3.04 4.10 3.30 2.65 8.57 4.37 3.00 8.84 3.95

Таблица 3 Максимальные относительные скачки цен вверх (в %)



AAPL -5.29 -4.16 -4.03 -6.63 -2.97 -4.00 -4.04 -3.24 -7.26 -3.92

COP -2.89 -2.70 -3.03 -6.29 -8.40 -3.16 -5.75 -4.19 -8.44 -5.37

EBAY -3.77 -3.98 -3.00 -4.02 -4.67 -3.61 -5.14 -4.19 -6.51 -5.34

FDX -3.72 -3.03 -3.29 -5.04 -3.71 -3.53 -6.20 -4.03 -7.17 -3.33

GM -4.13 -3.44 -4.92 -6.58 -4.21 -3.89 -4.55 -3.92 -10.78 -4.74

HAL -4.65 -3.90 -6.99 -5.16 -4.57 -5.64 -5.09 -5.61 -17.53 -6.12

HON -2.93 -2.26 -2.72 -4.49 -2.66 -2.68 -3.79 -3.41 -6.94 -3.03

HPQ -13.46 -4.22 -4.11 -5.48 -4.97 -6.13 -6.35 -9.57 -8.68 -4.72

IBM -2.85 -2.93 -3.04 -4.07 -3.85 -2.36 -4.22 -3.04 -6.36 -4.69

INTC -2.92 -2.81 -2.98 -4.33 -3.16 -2.85 -5.59 -3.60 -6.15 -6.82

KO -1.81 -2.83 -2.16 -2.91 -2.67 -1.83 -3.77 -2.38 -8.70 -2.79

MA -3.68 -2.75 -4.27 -4.41 -4.91 -4.33 -6.04 -4.96 -10.42 -6.24

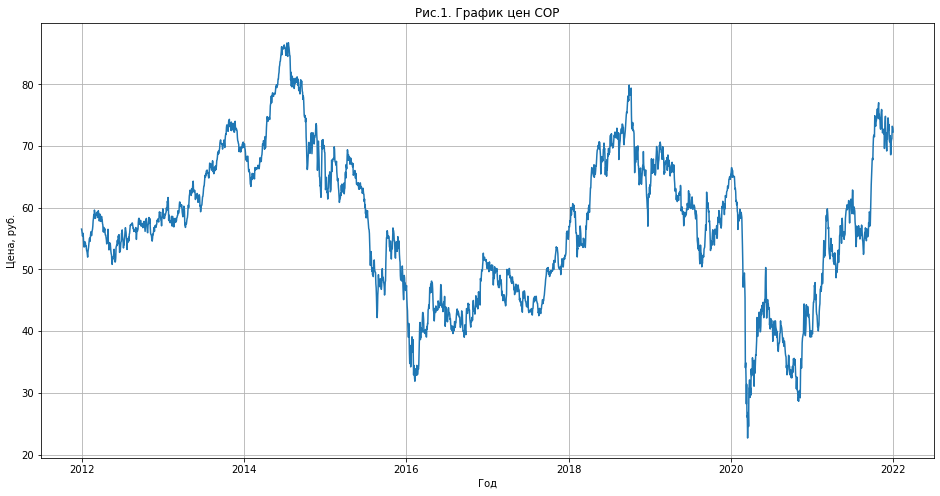
MCD -2.69 -1.74 -2.75 -4.46 -4.35 -2.42 -4.06 -2.84 -5.98 -2.54

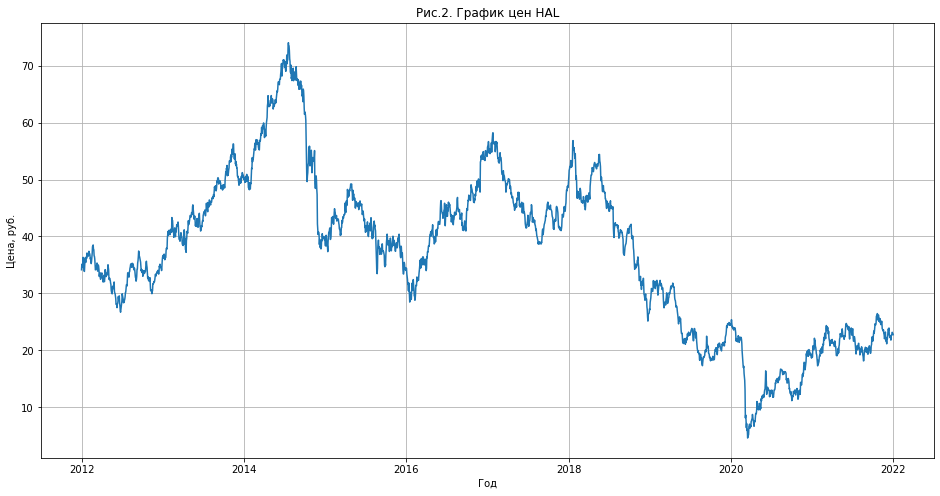
ORCL -4.73 -2.71 -3.37 -4.11 -3.52 -3.77 -4.68 -2.79 -6.11 -4.96

Таблица 4 Минимальные относительные скачки цен вверх (в %)

Судя по данным из таблиц 3 и 4 значимых скачков цен ни у одной из акций не наблюдается, однако стоит посмотреть на график “COP” и “HAL”, так как их относительные скачки выделяются по сравнению с другими тикерами.

Нарисуем график с помощью “КОД 3”.





Как видно из вышеприведенных графиков цен тикеры “COP” и “HAL” сильно заметных скачков не наблюдается и изменение цены не превышало даже 20 процентов, значит мы можем взять эти компании для анализа.

Предварительный анализ завершен. Компании, неподходящие по требованиям были исключены и теперь, можем приступить к рассмотрению теоретической и практической части.

# **3. Теоретическая справка**

## **3.1. Статистическая гипотеза**

**Статистическая гипотеза (Hj)** - некоторое утверждение о свойствах или характеристиках генеральной совокупности. Принять или опровергнуть статистическую гипотезу решается на основе определенного правила – статистического критерия.

Рассмотрим ошибки, которые может привести использование

статистического критерия:

1) Ошибка первого рода – правильная основная статистическая гипотеза H0 отвергается.

2) Ошибка второго рода - принимается основная неверная гипотеза H0, а альтернативная гипотеза H1 отвергается.

Обычно гипотеза H0 отклоняется, различия считаются статистически достоверными и принимается альтернативная гипотеза H1, если p < 0,05. Таким образом, чем p-value меньше, тем лучше, так как увеличивается сила отказа нулевой гипотезы.

## **3.2. Критерий Пирсона**

Критерий Пирсона, или как его по-другому называют Хи-квадрат, применяется чтобы проверить схожесть теоретического распределения и эмпирического. Рекомендуется применять данный критерий при объёме выборки больше, чем 100.

Чтобы использовать этот критерий нужно заранее разбить выборки на определенные равные интервалы.

Основной показатель критерия Пирсона – это статистика, которая определяется следующей формулой: , где  
 – количество чисел, которые попали в определенный интервал;  
 – обозначают объём случайной выборки;   
– это вероятность с который случайна величина попадет в определенный интервал, считается как , где в свою очередь  
F(x) – закон распределения;  
k – это количество степеней свободы, формула, по который рассчитывается: , где s – это кол-во групп выборки, а r –кол-во параметров в предполагаемом распределении. Так как в нашем случае распределение нормальное и два оцениваемых параметра это дисперсия и математическое ожидание, следовательно .

Если Хи-квадрат(наблюдаемый) меньше, чем Хи-квадрат (критической точки), то принимаем нулевую гипотезу. Наоборот, если Хи-квадрат (наблюдаемый) больше, чем Хи-квадрат (критической точки), то отклоняем нулевую гипотезу и принимаем альтернативную.

## **3.3. Критерий Колмогорова**

Критерий Колмогорова, или как его по-другому называют Колмогорова-Смирнова нужен для проверки гипотезы об отношении теоретического и эмпирического распределения одному определённому закону распределения.

Пусть задана выборка независимых случайных одинаково распределённых случайных величин – Xn, тогда Fn(x) будет теоретической функцией распределения, а F̂n(x) является эмпирической.

Статистика критерия задана следующим выражением:

D = max| F̂n(x) - Fn(x) |

Данный параметр характеризует силу расхождения между исследуемым распределением и ожидаемым.

Исходя из теоремы Колмогорова:

где

Далее нужно рассчитать показатель u, который равен . Для того, чтобы H0, была отвергнута нужно, чтобы данная величина была больше, чем критический , и следовательно гипотеза H1 будет принята. Наоборот, если критический уровень окажется больше вычисленного, то H0 принимается.

## **3.4. Индикатор RSI**

RSI (индекс относительной сильный) – это осциллирующий индикатор, который показывает, когда исследуемый актив находится в зоне перепроданности или перекупленности. Данный индикатор ходит в диапазоне от 0 до 100. Если RSI находится ниже 30, то говорят, что актив в зоне перепроданности, если в диапазоне от 30 до 70, то актив в своей норме, если выше 70, то в зоне перекупленности.

Один из популярных методов торговли по данному осцилятору заключается в поиске расхождений между ценой на графике и показателем RSI, то есть если цена исследуемого актива растет, образует локальный максимум, корректируется, а затем образует максимум выше предыдущего, в то время как индикатор на образовывает новый максимум, то принято считать, что скоро будет разворот рынка и цена пойдет на понижение. Аналогично в другую сторону.

Рассмотрим техническую часть осцилятора. Формула по которой считается RSI приведена ниже:

, где RS - относительная сила.

Для расчета относительной силы выбираются все свечи в выбранном промежутке времени, цена закрытия которых, выше чем цена закрытия предыдущей свечи и определяется среднее значение прироста цена с помощью экспоненциального среднего скользящего (EMA). Аналогично, для свечей, цена закрытия которых ниже, чем цена закрытия предыдущей свечи. Таким образом RS считается как отношение этих двух величин:

Единственный настраиваемый параметр для RSI – это временной промежуток. К примеру у нас дневные свечи, и по стандарту , то есть 14 дней.

**4. Проверка гипотез на модельных данных**

Начнем блок практической части с проверки на нормальное распределение модельных данных. Мы должны это сделать, чтобы убедиться, что написанный код работает правильно.

В программе методом Монте-Карло происходит вычисление статистики основного критерия, Хи-квадрат, N = 10000 раз при верной нулевой гипотезе при объёме выборки n = 252. Число интервалов k будем искать по формуле Старджесса: . Формируется таблица из 999 квантилей для распределения статистики

Для реализации данной задачи используем “КОД 4”.

Ниже будет представлена таблица из 9 квантилей (0.1, 0.2, … , 0.9):

|  |  |  |
| --- | --- | --- |
| 1 | quantile | value |
| 2 | 0.1 | 1.8156 |
| 3 | 0.2 | 2.5815 |
| 4 | 0.3 | 3.2910 |
| 5 | 0.4 | 3.9781 |
| 6 | 0.5 | 4.7025 |
| 7 | 0.6 | 5.4979 |
| 8 | 0.7 | 6.5094 |
| 9 | 0.8 | 7.8905 |
| 10 | 0.9 | 10.2366 |

Таблица 5. Квантили модельных данных

В работе представлены только 9 квантилей, все квантили от 0.001 до 0.999 будут представлены в отдельном файле.

Далее с помощью “КОД 5” построим гистограммы распределения P-значения на модельных данных.

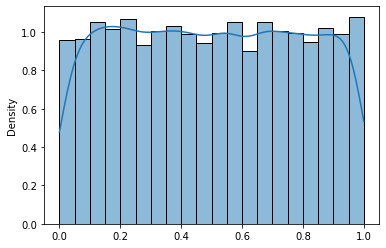


Рисунок 3. Гистограмма p-value по критерию Пирсона,   
на модельных данных

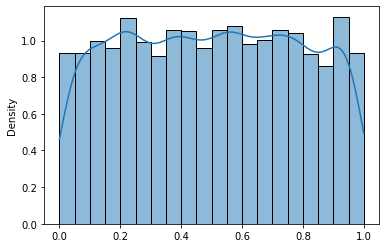


Рисунок 4. Гистограмма p-value по критерию Колмагорова,   
на модельных данных

Как видно из рисунков 3 и 4, равномерность подтверждается. Также была рассчитана проверка равномерности встроенной функцией Python по критерию Колмагорова и p-value данного исследования составил 0.446. Так как 0.446 больше, чем 0.05 (критическое p-значение), то мы можем утверждать, что нулевая гипотеза подтверждается.

# **5. Альтернативные гипотезы**

Для проверки критерия Пирсона будут использованы распределения Стьюдента со следующими степенями свободы: t(3), t(30), t(300). Распределение Стьюдента похоже на нормальное и стремится к нему при высоких степенях свободы. Понаблюдаем за этим, используя “КОД 7”.

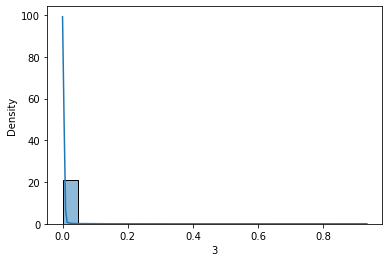


Рисунок 5. Гистограмма p-значения. С.с. 3

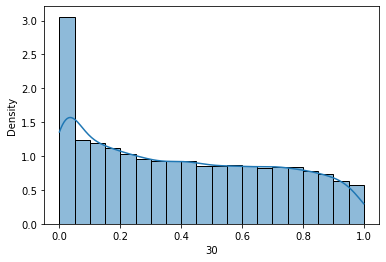


Рисунок 6. Гистограмма p-значения. С.с 30

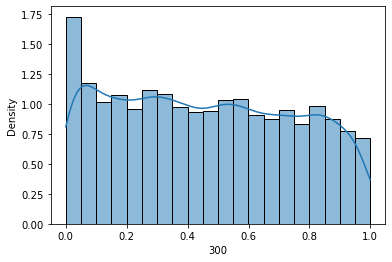


Рисунок 7. Гистограмма p-значения. С.с. 300

Действительно, как видно из графиков, распределение Стьюдента стремится к нормальному при возрастании числа степеней свободы.

С помощью “КОД 6”, вычислим мощность критерия при альтернативных распределениях, с учетом того, что p-value меньше 0.05.

|  |  |  |
| --- | --- | --- |
| 3 | 30 | 300 |
| 0.994 | 0.156 | 0.087 |

Таблица 6. Взаимосвязь числа степеней свободы и мощности критерия

Что и следовало ожидать, при маленькой степени свободы мощность критерия стремится к 1, что говорит о маленькой вероятности ошибки второго рода, а при возрастании числа степеней распределение становится похоже на нормальное и мощность критерия уменьшается. Следовательно, получить ошибку второго рода при больших числах степеней свободы возрастает.

# **6. Проверка гипотезы на реальных данных**

Приступим к проверке гипотезы на основе 14 компаний из индекса S&P 100 на основе критериев согласия Пирсона и Колмогорова.

Для начала, с помощью “КОД 8”, посчитаем p-значение логарифмической доходности для каждого тикета по каждому году. Данные представлены ниже.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 |
| AAPL | 0.0 | 0.832 | 0.002 | 0.0 | 0.0 | 0.0 | 0.001 | 0.276 | 0.011 | 0.016 |
| COP | 0.58 | 0.002 | 0.013 | 0.004 | 0.0 | 0.985 | 0.0 | 0.029 | 0.0 | 0.047 |
| EBAY | 0.625 | 0.063 | 0.18 | 0.001 | 0.009 | 0.047 | 0.239 | 0.001 | 0.001 | 0.004 |
| FDX | 0.276 | 0.009 | 0.015 | 0.0 | 0.0 | 0.0 | 0.0 | 0.413 | 0.0 | 0.067 |
| GM | 0.016 | 0.073 | 0.01 | 0.00 | 0.001 | 0.003 | 0.06 | 0.009 | 0.0 | 0.064 |
| HAL | 0.192 | 0.038 | 0.0 | 0.434 | 0.0 | 0.0 | 0.005 | 0.323 | 0.0 | 0.525 |
| HON | 0.029 | 0.661 | 0.051 | 0.0 | 0.003 | 0.0 | 0.0 | 0.0 | 0.0 | 0.297 |
| HPQ | 0.0 | 0.0 | 0.0 | 0.0 | 0.05 | 0.0 | 0.0 | 0.0 | 0.0 | 0.052 |
| IBM | 0.001 | 0.003 | 0.024 | 0.0 | 0.0 | 0.222 | 0.002 | 0.0 | 0.0 | 0.0 |
| INTC | 0.982 | 0.394 | 0.0 | 0.0 | 0.006 | 0.0 | 0.098 | 0.484 | 0.0 | 0.0 |
| KO | 0.281 | 0.025 | 0.083 | 0.01 | 0.0 | 0.0 | 0.0 | 0.727 | 0.0 | 0.005 |
| MA | 0.409 | 0.292 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| MCD | 0.014 | 0.229 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.011 | 0.0 | 0.313 |
| ORCL | 0.0 | 0.13 | 0.011 | 0.0 | 0.0 | 0.0 | 0.0 | 0.166 | 0.0 | 0.0 |

Таблица 7. P-value для реальных данных без учета rsi

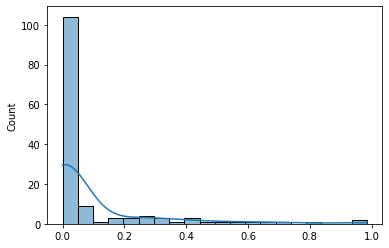


Рисунок 8. Гистограмма p-value для реальных данных без учета rsi

Гистограмма явно даёт понять, что распределение p-значения неравномерно и близко к нулю. Следовательно, гипотеза о нормальном распределении логарифмической доходности отклоняется.

С помощью “КОД 9” посмотрим на долю верных гипотез в зависимости критического p-значения. Результат представлен в таблице 8

|  |  |  |
| --- | --- | --- |
| 1% | 5% | 10% |
| 0.364 | 0.257 | 0.193 |

Таблица 8. Доля верных гипотез

Теперь проверим гипотезу с учетом показателя RSI. Будем считать, что n = 14, то есть на каждой итерации цикла будем брать текущую свечу и 13 предыдущих. Сделаем это с помощью “КОД 10”. Данные буду представлены ниже в таблицах.

|  |  |  |  |
| --- | --- | --- | --- |
|  | RSI < 40 | 30 < RSI < 70 | RSI > 60 |
| AAPL | 0.267 | 0.0 | 0.922 |
| COP | - | 0.0 | - |
| EBAY | 0.558 | 0.0 | 0.129 |
| FDX | 0.113 | 0.0 | - |
| GM | - | 0.0 | - |
| HAL | - | 0.0 | - |
| HON | 0.006 | 0.0 | 0.028 |
| HPQ | - | 0.0 | - |
| IBM | - | 0.0 | - |
| INTC | - | 0.0 | - |
| KO | - | 0.0 | - |
| MA | 0.0 | 0.0 | 0.745 |
| MCD | - | 0.0 | - |
| ORCL | - | 0.0 | - |

Таблица 9. P-значение критерия Пирсона  
 для каждой компании за весь период с учетом значения RSI

|  |  |  |  |
| --- | --- | --- | --- |
|  | RSI < 40 | 30 < RSI < 70 | RSI > 60 |
| AAPL | 0.0 | 0.0 | 0.0 |
| COP | - | 0.0 | - |
| EBAY | 0.0 | 0.0 | 0.0 |
| FDX | 0.0 | 0.0 | 0.185 |
| GM | 0.34 | 0.0 | - |
| HAL | 0.058 | 0.0 | - |
| HON | 0.0 | 0.0 | 0.002 |
| HPQ | 0.193 | 0.0 | 0.968 |
| IBM | - | 0.0 | - |
| INTC | - | 0.0 | 0.343 |
| KO | - | 0.0 | - |
| MA | 0.0 | 0.0 | 0.0 |
| MCD | 0.5 | 0.0 | - |
| ORCL | 0.334 | 0.0 | 0.33 |

Таблица 10. P-значение критерия Колмогорова  
 для каждой компании за весь период с учетом значения RSI

Из вышеперечисленного, можно сделать вывод, что нулевая гипотеза о нормальном распределении логарифмической доходности с учетом RSI отвергается потому что, в большинстве случаев P-значения равны 0 или близки к 0.

# **7. Заключение**

В представленной курсовой работе проверялась гипотеза о нормальном распределении логарифмической доходности 14 акций индекса S&P 100 при условии, что наканун е значение индикатора RSI находилось в определенном диапазоне. По результатам работы подтверждаются результаты курсовых работ прошлых лет, а именно отклонение нулевой гипотезы о нормально распределении.

# **8. Литература**

1. Wikipedia URL: [https://ru.wikipedia.org/wiki/](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D0%B5%D0%BA%D1%81_%D0%BE%D1%82%D0%BD%D0%BE%D1%81%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D0%BB%D1%8B)
2. Yahoo Finance URL: <https://finance.yahoo.com/>
3. Состав индекса S&P100. URL: <https://www.finanz.ru/indeksi/sostav/s&p_100>
4. TradingView URL: <https://www.tradingview.com/>
5. Бабайцев В.А., Браилов А.В., Солодовников А.С. Математика в экономике. Теория вероятностей: Курс лекций. М.: Финансовая академия, 2002

# **9. Приложения**

**Приложение 1**

Код программ для курсовой работы написано на языке Python

Характеристики компьютера:

Процессор: AMD Ryzen 5 3600 6-Core Processor 3.95 GHz

Оперативная память: 16 гб

**Приложение 2**

from statistics import mean

import pandas as pd

import numpy as np

import seaborn as sns

from math import sqrt, log, log2, isnan

import scipy.stats as sts

import matplotlib.pyplot as plt

**Код 1. Таблица 2. Кол-во торговых дней.**

dataset = ['AAPL','BCO.HM','COP','EBAY','FDX','GM','HAL','HON','HPQ','IBM','INTC','KO','MA','MCD','ORCL']

SandP100 = pd.DataFrame(index = dataset)

for x in range(2012,2022):

years=[]

for i in range(len(dataset)):

df = pd.read\_csv('D:/curs/'+dataset[i]+'.csv', delimiter=',')

year = (df['Date']>=str(x) + '-01-01') & (df['Date'] < str(x+1) + '-01-01')

years.append(len(df[year]))

SandP100[str(x)] = years

print(SandP100)

**Код 2. Таблица 3 и 4. Максимальные скачки цен вверх и вниз.**

dataset = ['AAPL','COP','EBAY','FDX','GM','HAL','HON','HPQ','IBM','INTC','KO','MA','MCD','ORCL']

SandPmin = pd.DataFrame(index = dataset)

SandPmax = pd.DataFrame(index = dataset)

for yea in range(2012,2022):

tiket = pd.read\_csv('D:/curs/AAPL.csv', delimiter=',')

year = (tiket['Date']>=str(yea) + '-01-01') & (tiket['Date'] < str(yea+1) + '-01-01')

valmin = 0

valmax = 0

for tik in range(len(tiket['Date'])):

if year[tik]:

val = (tiket['Close'][tik] - tiket['Open'][tik]) / tiket['Open'][tik] \* 100

if round(val,5) > valmax:

valmax = round(val,2)

if round(val,5) < valmin:

valmin = round(val,2)

SandPmax[yea] = [valmax,0,0,0,0,0,0,0,0,0,0,0,0,0]

SandPmin[yea] = [valmin,0,0,0,0,0,0,0,0,0,0,0,0,0]

for y in range(len(dataset)):

tiket = pd.read\_csv('D:/curs/'+dataset[y]+'.csv', delimiter=',')

for yea in range(2012,2022):

valmin = 0

valmax = 0

year = (tiket['Date']>=str(yea) + '-01-01') & (tiket['Date'] < str(yea+1) + '-01-01')

for tik in range(len(tiket['Date'])):

if year[tik]:

val = (tiket['Close'][tik] - tiket['Open'][tik]) / tiket['Open'][tik] \* 100

if round(val,5) > valmax:

valmax = round(val,2)

if round(val,5) < valmin:

valmin = round(val,2)

SandPmax[yea][dataset[y]] = valmax

SandPmin[yea][dataset[y]] = valmin

print(SandPmax)

print(SandPmin)

**Код 3. Рисунок 1 и 2. График цен акций.**

import matplotlib.pyplot as plt

import numpy as np

from numpy import linspace

import pandas as pd

#первый график

df = pd.read\_csv('D:/curs/COP.csv', delimiter=',')

newdf = pd.DataFrame()

newdf['date'] = df['Date']

newdf['close'] = df['Close']

year = (df['Date']>='2012-01-01') & (df['Date'] < '2022-01-01')

y = newdf['close']

x = linspace(2012, 2022, len(y))

fig, ax = plt.subplots(figsize=(16,8))

plt.grid(True) # линия сетки

plt.plot(x,y)

plt.ylabel('Цена, руб.')

plt.xlabel('Год')

plt.title('Рис.1. График цен COP')

plt.show()

#второй график

df = pd.read\_csv('D:/curs/HAL.csv', delimiter=',')

newdf = pd.DataFrame()

newdf['date'] = df['Date']

newdf['close'] = df['Close']

year = (df['Date']>='2012-01-01') & (df['Date'] < '2022-01-01')

y = newdf['close']

x = linspace(2012, 2022, len(y))

fig, ax = plt.subplots(figsize=(16,8))

plt.grid(True) # линия сетки

plt.plot(x,y)

plt.ylabel('Цена, руб.')

plt.xlabel('Год')

plt.title('Рис.2. График цен HAL')

plt.show()

**Код 4. Таблица 5. 9 и 999 квантилей.**

# функция,возвращающая мат. ожидаение, дисперсия, массив элементов в интервалах и список центральных точек отрезков разбиения

def ozenka(massive):

massive = sorted(massive)

n = len(massive)

k = int(math.log2(n)) + 1

step = (max(massive)-min(massive))/k

centre = [min(massive)]

razb\_otrez = []

razb\_otrez.append([x for x in massive if x < centre[-1]+step/2])

for i in range(2, k):

centre.append(centre[0]+(i-1)\*step)

razb\_otrez.append([x for x in massive if centre[-1]-0.5\*step < x < centre[-1]+0.5\*step])

centre.append(centre[0]+(k-1)\*step)

razb\_otrez.append([x for x in massive if x > centre[-1]-0.5\*step])

math\_wa = sum(massive)/n # мат ожидание

summa = 0

for i in range(n):

summa += (massive[i] - math\_wa)\*\*2

dispar = summa/n #дисперсия

# список частот (количество элементов в интервалах)

kolvo = [len(x) for x in razb\_otrez]

return math\_wa, dispar, kolvo, centre

def Pirson(n):

viborka = sts.norm(0, 1).rvs(n)

m = int(math.log2(n)) + 1

step = max(viborka)-min(viborka)

step = step/m

asdfgh = ozenka(viborka)

M = asdfgh [0]

Disp = asdfgh[1]

kolvo = asdfgh[2]

centre = asdfgh[3]

Exp = sts.norm(M, math.sqrt(Disp))

Prob = [Exp.cdf(centre[0]+0.5\*step)]

for i in range(1, len(centre)-1): # функция cdf вычисляет вероятность попадания случайной величины в исселдуемый промежуток

Prob.append(Exp.cdf(centre[i]+0.5\*step) - Exp.cdf(centre[i]-0.5\*step))

Prob.append(1-Exp.cdf(centre[-1]-0.5\*step))

# высчитываем статистику критерия Пирсона

chi = []

for i in range(len(Prob)):

chi.append(((kolvo[i]-n\*Prob[i])\*\*2)/(n\*Prob[i])) # формула

Xi\_square = sum(chi)

# вычисляем p-значение

p\_value = (sts.chi2(len(kolvo)-3).sf(Xi\_square)) # функция chi2. вычитаем 3 из-за количества степеней свободы

# возвращаем статистику критерия Пирсона и p-value

return Xi\_square, p\_value

N = 10000

# создаём пустые таблицы для 9 и 999 квантилей

df\_999 = pd.DataFrame(index = [i for i in range(1,1000)])

df\_9 = pd.DataFrame(index = [i for i in range(1,10)])

# создаём списки значений статистики и p-value

xi = []

p\_value = []

# вычисляем 10000 значений статистики по критерию пирсона

for statist in range(N):

current\_pirs = Pirson(252)

xi.append(current\_pirs[0])

p\_value.append(current\_pirs[1])

# подсчет 9 и 999 квантилей

xi2\_q9 = np.quantile(xi, np.arange(0.1, 1, 0.1))

xi2\_q999 = np.quantile(xi, np.arange(0.001, 1, 0.001))

# запись в таблицы квантилей и их значений

df\_999['quantile'] = [round(i, 3) for i in list(np.arange(0.001, 1, 0.001))]

df\_9['quantile'] = [round(i, 3) for i in list(np.arange(0.1, 1, 0.1))]

df\_999['value'] = np.round(xi2\_q999, 4)

df\_9['value'] = np.round(xi2\_q9, 4)

print(df\_9)

print(df\_999)

**Код 5. Рисунок 3 и 4. Гистограмма p-значений критерий Пирсона и Колмогорова на модельных данных.**

n = 252

N = 10000

xi2 = []

chi\_square = []

for p in range(10000):

chi\_square.append(Pirson(n)[0])

chi\_square\_q999 = np.quantile(chi\_square, np.arange(0.001, 1, 0.001))

p\_value\_pirs\_list = []

for pp in range(10000):

lenn = 0

xx = Pirson(n)[0]

for p in range(len(chi\_square\_q999)):

if chi\_square\_q999[p] > xx:

lenn += 1

p\_value\_pirs\_list.append(lenn/len(chi\_square\_q999))

p\_value\_ks\_list = [] #cоздаем cписок p-значений критерия Колмогорова - Смирнова

n = 250

for p in range(10000):

list\_norm = sts.norm(0, 1).rvs(n)

p\_value\_ks\_list.append(sts.kstest(list\_norm,'norm')[1])

sns.histplot(p\_value\_ks\_list, bins = 20, stat = 'density', kde = True)

sns.histplot(p\_value\_pirs, bins = 20, stat = 'density', kde = True)

# cделаем проверку, что распределения p-значений близки друг к другу и выведем P-value двух выборок

p\_value = sts.ks\_2samp(p\_value\_ks\_list,p\_value\_pirs\_list)

print("P-value =",round(p\_value[1],3))

**Код 6. Таблица 6. Мощность критерия для альтернативных гипотез.**

N = 1000

n = 252

df = pd.DataFrame()

p\_value\_pirs = []

for i in range(1000):

massive = sts.t(3).rvs(n)

k = 1 + int(math.log2(n))

step = max(massive)-min(massive)

step = step / k

asdfgh = ozenka(massive)

M = asdfgh[0]

Disp = asdfgh[1]

kolvo = asdfgh[2]

centre = asdfgh[3]

Exp = sts.norm(M, math.sqrt(Disp))

Prob = [Exp.cdf(centre[0]+0.5\*step)]

for i in range(1, len(centre)-1): # функция cdf вычисляет вероятность попадания случайной величины в исселдуемый промежуток

Prob.append(Exp.cdf(centre[i]+0.5\*step) - Exp.cdf(centre[i]-0.5\*step))

Prob.append(1-Exp.cdf(centre[-1]-0.5\*step))

T = []

for i in range(len(Prob)):

if Prob[i] == 0:

continue

T.append(((kolvo[i]-n\*Prob[i])\*\*2)/(n\*Prob[i]))

xi\_2 = sum(T)

p\_value.append(sts.chi2(len(kolvo)-3).sf(xi\_2))

PV = [v for v in p\_value if v < 0.05 or math.isnan(v)]

df['3'] = [len(PV)/1000]

#аналогично сделаем для степеней свободы равных 30 и 300

print(df)

**Код 7. Рисунок 5, 6, 7. Гистограмма p-значения критерия Пирсона при различных степенях свободы (3, 30, 30 0).**

n = 252

N = 10000

df = pd.DataFrame()

for lenn in (3,30,300):

p\_value = []

for j in range(N):

massive = sts.t(lenn).rvs(n) # генерируем выборку распределения Стьюдента нужного объёма

k = int(math.log2(n)) + 1 # формула Стерджесса

step = (max(massive)-min(massive))/k

function = ozenka(massive)

M, Disp, centre, kolvo = function[0], function[1], function[3], function[2]

Exp = sts.norm(M, math.sqrt(Disp))

Prob = [Exp.cdf(centre[0]+0.5\*step)]

for i in range(1, len(centre)-1): # функция cdf вычисляет вероятность попадания случайной величины в исселдуемый промежуток

Prob.append(Exp.cdf(centre[i]+0.5\*step) - Exp.cdf(centre[i]-0.5\*step))

Prob.append(1-Exp.cdf(centre[-1]-0.5\*step))

T = []

for u in range(len(Prob)):

if Prob[u] == 0:

continue

T.append(((kolvo[u]-n\*Prob[u])\*\*2)/(n\*Prob[u]))

xi2 = sum(T)

p\_value.append(sts.chi2(len(kolvo)-3).sf(xi2))

lenn = str(lenn)

df[lenn] = p\_value

sns.histplot(df['3'], bins = 20, stat = 'density', kde = True)

sns.histplot(df['30'], bins = 20, stat = 'density', kde = True)

sns.histplot(df['300'], bins = 20, stat = 'density', kde = True)

**Код 8. Таблица 7. Русинок 8. p-значение критерия Пирсона логарифмической доходности на реальных данных.**

dataset = ['AAPL','COP','EBAY','FDX','GM','HAL','HON','HPQ','IBM','INTC','KO','MA','MCD','ORCL']

SandPpirs = pd.DataFrame(index = dataset)

for i in range(2012,2022):

SandPpirs[str(i)] = [1,0,0,0,0,0,0,0,0,0,0,0,0,0]

for i in range(2012,2022):

SandPpirs[str(i)] = SandPpirs[str(i)].astype(float)

p\_value\_for\_hist = []

for y in range(len(dataset)):

values = []

df = pd.read\_csv('D:/curs/'+dataset[y]+'.csv', delimiter=',')

for x in range(2012,2022):

viborka = []

year = (df['Date']>=str(x) + '-01-01') & (df['Date'] < str(x+1) + '-01-01')

for i in range(len(df['Date'])):

if year[i]:

viborka.append(np.log(df['Close'][i]/df['Open'][i]))

#подсчет n

year = (df['Date']>=str(x) + '-01-01') & (df['Date'] < str(x+1) + '-01-01')

n = len(df[year])

k = 1 + int(math.log2(n))

step = max(viborka)-min(viborka)

step = step / k

asdfgh = ozenka(viborka)

M = asdfgh[0]

Disp = asdfgh[1]

kolvo = asdfgh[2]

centre = asdfgh[3]

Exp = sts.norm(M, math.sqrt(Disp))

Prob = [Exp.cdf(centre[0]+0.5\*step)]

for i in range(1, len(centre)-1): # функция cdf вычисляет вероятность попадания случайной величины в исселдуемый промежуток

Prob.append(Exp.cdf(centre[i]+0.5\*step) - Exp.cdf(centre[i]-0.5\*step))

Prob.append(1-Exp.cdf(centre[-1]-0.5\*step))

T = []

for i in range(len(Prob)):

if Prob[i] == 0:

continue

T.append(((kolvo[i]-n\*Prob[i])\*\*2)/(n\*Prob[i]))

xi\_2 = sum(T)

p\_value = (sts.chi2(len(kolvo)-3).sf(xi\_2))

p\_value\_for\_hist.append(p\_value) # для гистограммы массив из p-значений

SandPpirs[str(x)][dataset[y]] = round(p\_value,3)

print(SandPpirs)

sns.histplot(p\_value\_for\_hist, bins = 20, stat = 'count', kde = True)

**Код 9. Таблица 8. Доля верных гипотез на реальных данных.**

p\_val\_1 = 0

p\_val\_5 = 0

p\_val\_10 = 0

p\_val = pd.DataFrame(columns = ['1%','5%','10%'])

for i in range(2012,2022):

for j in range(len(dataset)):

if SandPpirs[str(i)][dataset[j]] > 0.01:

p\_val\_1 += 1

if SandPpirs[str(i)][dataset[j]] > 0.05:

p\_val\_5 += 1

if SandPpirs[str(i)][dataset[j]] > 0.1:

p\_val\_10 += 1

percents = [round(p\_val\_1/len(dataset)/(2022-2012),3),round(p\_val\_5/len(dataset)/(10),3),round(p\_val\_10/len(dataset)/10,3)]

p\_val.loc[0] = percents

print(p\_val)

**Код 10. Таблица 9 и 10. Нахождение p-значений на реальных данных с учетом rsi.**

dataset = ['AAPL','COP','EBAY','FDX','GM','HAL','HON','HPQ','IBM','INTC','KO','MA','MCD','ORCL']

SandP\_RSI\_all\_ema = pd.DataFrame(index = dataset)

# массив изначальных EMA взятых с сайта TradingView

EMA = [14.3, 54.6, 12.8, 83.3, 20.3, 33.8, 51.3, 11.8, 177.2, 24.3, 34.5, 36.9, 98.9, 27]

for x in range(len(dataset)):

df = pd.read\_csv('D:/curs/'+dataset[x]+'.csv', delimiter=',')

rsi\_below\_30 = []

rsi\_between\_30\_70 = []

rsi\_higher\_70 = []

for i in range(15,len(df['Date'])):

mas\_rsi = []

mas\_rsi\_up = []

mas\_rsi\_down = []

for j in range(i-15, i):

mas\_rsi.append(df['Close'][j])

#заполняем массив для средних

for j in range(1,len(mas\_rsi)):

if mas\_rsi[j] > mas\_rsi[j-1]:

mas\_rsi\_up.append(mas\_rsi[j])

if mas\_rsi[j] < mas\_rsi[j-1]:

mas\_rsi\_down.append(mas\_rsi[j])

#проверим, что в массивах хотябы 1 элемент

if len(mas\_rsi\_down) == 0:

mas\_rsi\_down.append(df['Close'][i-14])

if len(mas\_rsi\_up) == 0:

mas\_rsi\_up.append(df['Close'][i-14])

#посчитаем значения для формулы

ema\_first\_up = EMA[x]

ema\_first\_down = EMA[x]

alpha\_up = 2/(1+len(mas\_rsi\_up)) #сглаживающий фактор

alpha\_down = 2/(1+len(mas\_rsi\_down)) #сглаживающий фактор

ema\_up = (df['Close'][i] \* alpha\_up) + (ema\_first\_up \* (1-alpha\_up))

ema\_down = (df['Close'][i] \* alpha\_down) + (ema\_first\_down \* (1-alpha\_down))

rs = ema\_up/ema\_down

ema\_first\_up = ema\_up

ema\_first\_down = ema\_down

#формула

rsi = 100-(100/(1+rs))

#добавление в массив цен закрытий

if rsi < 40:

rsi\_below\_30.append(np.log(df['Close'][i]/df['Open'][i]))

if rsi < 70 and rsi > 30:

rsi\_between\_30\_70.append(np.log(df['Close'][i]/df['Open'][i]))

if rsi > 60:

rsi\_higher\_70.append(np.log(df['Close'][i]/df['Open'][i]))

#массив для трех случаев

rsi\_massive = [rsi\_below\_30, rsi\_between\_30\_70, rsi\_higher\_70]

p\_value\_massive = []

p\_val\_kolm = []

#посчитаем p-значение для каждой выборки

for u in range(3):

n = len(rsi\_massive[u])

if n == 0:

p\_value\_massive.append(None)

p\_val\_kolm.append(None)

continue

k = int(math.log2(n)) + 1

step = max(rsi\_massive[u])-min(rsi\_massive[u])

step = step / k

asdfgh = ozenka(viborka)

M = asdfgh[0]

Disp = asdfgh[1]

kolvo = asdfgh[2]

centre = asdfgh[3]

Exp = sts.norm(M, math.sqrt(Disp))

Prob = [Exp.cdf(centre[0]+0.5\*step)]

for i in range(1, len(centre)-1):

Prob.append(Exp.cdf(centre[i]+0.5\*step) - Exp.cdf(centre[i]-0.5\*step))

Prob.append(1-Exp.cdf(centre[-1]-0.5\*step))

T = []

for i in range(len(Prob)):

if Prob[i] == 0:

continue

T.append(((kolvo[i]-n\*Prob[i])\*\*2)/(n\*Prob[i]))

xi\_2 = sum(T)

p\_value = (sts.chi2(len(kolvo)-3).sf(xi\_2))

p\_value = round(p\_value,3)

p\_value\_massive.append(p\_value)

#критерий Колмогорова

p\_val\_kolm.append(round(sts.kstest(rsi\_massive[u],'norm')[1],3))

#print(p\_value\_massive)

print(p\_val\_kolm)