Федеральное государственное образовательное бюджетное учреждение
высшего профессионального образования
«Финансовый университет при Правительстве Российской Федерации»

Департамент анализа данных, принятия решений и финансовых технологий

Курсовая работа

на тему

«Проверка гипотезы о равенстве дисперсий логарифмической доходности индекса фондового рынка и входящих в его состав акций»

Вид данных для исследования:

«**Котировки акций компаний, входящих в индекс ММВБ потребительского сектора**»

Выполнил:

студент группы ПМ20-4,

Есаков В.А.

Научный руководитель:

доцент, к.э.н.

Игудесман К.Б.

Москва
2022

План работы

I. Введение

В качестве задачи для исследования в курсовой работе я проведу аналитическую проверку правильности следующей гипотезы: дисперсия логарифмической доходности фондового рынка равна дисперсиям входящих в его состав акций. Для подтверждения справедливости гипотезы будет применяться простой и распространённый способ сравнения дисперсий – критерий Фишера.

Предметом исследования я выбрал данные котировок акций с фондовой биржи из одного индекса. В данном случае, индекс финансового сектора (MOEXFN), являющийся частью Московской биржи (ММВБ). Рассматриваемый в работе временной отрезок – с 1 января 2016 года по 30 декабря 2021 года.

Индекс ММВБ, экспертами указывается, как один из важнейших показателей экономики, так как он взвешен по рыночной капитализации. Является ценовым композитным фондовым индексом и включает в себя пакеты акций крупнейших российских эмитентов. Показателем их значимости является, главным образом, принадлежность к ликвидным и активно расширяющимся сферам жизни : финансовый сектор, сельское хозяйство, промышленность, транспортные сети, здравоохранение.

Выбранный для анализа индекс (MOEXFN – финансовый сектор) состоит из банковских компаний и других крупных организаций, заведующих финансами. Они предоставляют различные услуги по управлению, хранению деньгами, их инвестициями или займами. В данных на апрель 2022 года, частью индекса являются: Сбербанк (один из крупнейших банков России), имеет обычный и привилегированный пакет акций, QIWI (российская компания, специализирующаяся на виртуальных платёжных сервисах), ВТБ, Тинькофф, банк Санкт-Петербурга, Московский кредитный банк (крупные российские банки, предоставляющие различные услуги по хранению активов, депозитов, получению займов, инвестициям в

различные ценные бумаги), RENI (компания, предоставляющая услуги страхования по любым имущественным, здравоохранительным и другим вопросам), SFI (публичное акционерное агентство, занимающееся инвестиционными управлением), индекс Московской биржи (Московская биржа занимается проведением операций по ценным бумагам, задаёт основной инвестиционный курс для всего российского рынка, является общим финансовым регулятором, поэтому является неотъемлемой частью финансового сектора экономики).

Отличительной особенностью данной работы с технической точки зрения будет проверка гипотезы с помощью языка программирования Python (версии 3.0 и новее). Это позволит действительно разумно использовать ресурсы для анализа большого объёма данных. Дополнительно будет использоваться среда разработки Jupiter Notebook, поддерживающая множество гибких способов визуализации данных в интерактивном формате, в частности : графики, гистограммы, формулы, табличные структуры.

Выбор язык программирования Python позволит использовать вычислительные мощности компьютера и современных программ для анализа данных, что является актуальным средством для исследования данной гипотезы. Выбор финансового сектора обусловлен его ключевым влиянием на всю экономику и авторитет всего государства в целом, который будет актуален всегда при существующей модели функционирования рынка.

В качестве итогов проведения данной работы получится установить наличие взаимосвязи между изменениями логарифмической доходности фондового рынка и изменениями логарифмической доходности входящих в состав индекса акций, а значит и их биржевых котировок, как следствие.

I. Основные теоретически положения

1) Математическая статистика

Предмет высшего образования – теория вероятностей и математическая статистика – включает в себя этот раздел математики и представляет собой совокупность множества более узких дисциплин, таких как финансовая, социальная, экономическая или любая другая статистическая аналитика сферы жизни общества. Основная задача дисциплины – обоснование теоретической базы всех этих конкретных дисциплин.

Создание единой методики и алгоритма исследования задачи путём фундаментальных и прикладных утверждений определяет цель математической статистики.

2) Статистическая гипотеза

Такой гипотезой является любого рода утверждение, закрепляющее за ситуацией определённые свойства : параметры внутреннего распределения, о соотношениях между случайными величинами и так далее. При этом если достоверно известно, что предположение о характере генерального распределения рассчитано до конечного числа параметров, то такую гипотезу можно считать параметрической.

Введём обозначения - H_0 (основная) и H_1(дополнительная) – две статистические гипотезы, которые являются взаимоисключающими. Далее всегда будем считать, что у нас есть базисное утверждение о справедливости одной из гипотез. Соответственно, если гипотеза H_0 подтвердилась, то H_1 автоматически была опровергнута и наоборот.

Статистикой критерия называют правило, при котором отрицание гипотезы H_0 и наличие некоторой выборки $(x_1, \dots, x_n)$ принадлежащей непустой области К, и таким же образом в обратную сторону гипотеза H_0 подтверждается, а выборка $(x_1, \dots, x_n)$ не входит в некоторую область К. В этом случае К принято называть областью допустимых значений (альтернативное название – область принятия гипотезы), при обратной

ситуации (когда в какой-то зоне отрицается принятие гипотезы H_0) её называют критической областью, как правило задаваемую через неравенства:

$$K = \{(x_1, \dots, x_n) \in R^n : t > c\} \quad (2.1)$$

или

$$K = \{(x_1, \dots, x_n) \in R^n : t < c\} \quad (2.2)$$

или

$$K = \{(x_1, \dots, x_n) \in R^n : t < c_1\} \cup \{(x_1, \dots, x_n) \in R^n : t > c_2\}, \quad (2.3)$$

где $c, c_1, c_2 (c_2 > c_1) = const$, $t = t(x_1, \dots, x_n)$ - статистики критерия

[6]

3) Первый и второй род ошибок

Два различных вида ошибок могут возникнуть при статистических вычислениях. I род – гипотеза H_0 отвергается, хотя в действительности является верной. Ошибкой же II-ого рода является ситуация, в которой отрицается гипотеза H_1 , являющаяся истинной.

Ошибка первого рода является мерой значимости критерия (обозначается как α), а ошибка второго рода задаёт формулу для мощности критерия. Если принять за β вероятность ошибки, то искомое значение будет равно 1 – β.

Отклонение или принятие основной гипотезы опирается на оценку уровня значимости критерия. Так проверяется статистическая гипотеза для фиксированного значения уровня значимости. Если его изменить, значения придётся пересчитывать, в том числе критическую оценку [1]

4) P-значения

Данная характеристика влияет на принятие основной гипотезы для любого уровня значимости, что позволяет не делать лишних операций по вычислению критических значений.

$PV(\overrightarrow{x}) \geq \alpha$, где $\vec{x}$ является фиксированным выбором из случайного набора данных, а – уровень значимости, для которых принята гипотеза H_0

И напротив, $PV(\overrightarrow{x}) < \alpha$, для всех а, при которых гипотеза отвергается.
PV(x) является P-значением статистического критерия.

Теперь отдельно разберём случай, при котором $PV(\overrightarrow{x}) = \alpha$.

Пусть $c(\alpha)$ - уравнение произвольной убывающей функции

$$K_\alpha = \{t(\vec{x}) > c(\alpha)\}, \quad (4.1)$$

$$PV(\overrightarrow{x}) = \mathrm{c}(t(\vec{x}))^{-1}) \quad (4.2)$$

Также из этого равенства следует другое $t(\vec{x}) = \mathrm{c}(\alpha)$, утверждающее, что основная гипотеза принимается. Так получается вывод общепринятой формулы, которую чаще всего применяют при расчёте p-значений:

$$PV(\overrightarrow{x}) = P_{H_0}(t(\vec{X}) > t(\vec{x}) \quad (4.3)$$

Действительно, при любом уровне значимости α

$$P_{H_0}(t(\vec{X}) > c(\alpha) = \alpha. \quad (4.4)$$

Аналогично $K_\alpha = \{t(\vec{x}) < c(\alpha)\}, \quad (4.5)$

где $c(\alpha)$ – непрерывная возрастающая функция, Р-значение удовлетворяет отношению

$$PV(\overrightarrow{x}) = P_{H_0}(t(\vec{X}) < t(\vec{x}) \quad (4.6)$$

[1]

По таблице p-значений, зная число степеней свободы эксперимента в соответствующей строке находим большее значение, чем значений хи-квадрата. Уравнение для хи-квадрата следующее: $x^2 = \Sigma((o-e)^2/e)$, где «о» — это наблюдаемое значение, а «е» — это ожидаемое значение. Суммируем результаты данного уравнения для всех возможных результатов и после этого по данной таблице определяем соответствующее p-значение в заголовке столбца.

	P										
DF	**0.995**	**0.975**	**0.20**	**0.10**	**0.05**	**0.025**	**0.02**	**0.01**	**0.005**	**0.002**	**0.001**
1	0.0000393	0.000982	1.642	2.706	3.841	5.024	5.412	6.635	7.879	9.550	10.828
2	0.0100	0.0506	3.219	4.605	5.991	7.378	7.824	9.210	10.597	12.429	13.816
3	0.0717	0.216	4.642	6.251	7.815	9.348	9.837	11.345	12.838	14.796	16.266
4	0.207	0.484	5.989	7.779	9.488	11.143	11.668	13.277	14.860	16.924	18.467
5	0.412	0.831	7.289	9.236	11.070	12.833	13.388	15.086	16.750	18.907	20.515
6	0.676	1.237	8.558	10.645	12.592	14.449	15.033	16.812	18.548	20.791	22.458
7	0.989	1.690	9.803	12.017	14.067	16.013	16.622	18.475	20.278	22.601	24.322
8	1.344	2.180	11.030	13.362	15.507	17.535	18.168	20.090	21.955	24.352	26.124
9	1.735	2.700	12.242	14.684	16.919	19.023	19.679	21.666	23.589	26.056	27.877
10	2.156	3.247	13.442	15.987	18.307	20.483	21.161	23.209	25.188	27.722	29.588
11	2.603	3.816	14.631	17.275	19.675	21.920	22.618	24.725	26.757	29.354	31.264
12	3.074	4.404	15.812	18.549	21.026	23.337	24.054	26.217	28.300	30.957	32.909
13	3.565	5.009	16.985	19.812	22.362	24.736	25.472	27.688	29.819	32.535	34.528
14	4.075	5.629	18.151	21.064	23.685	26.119	26.873	29.141	31.319	34.091	36.123
15	4.601	6.262	19.311	22.307	24.996	27.488	28.259	30.578	32.801	35.628	37.697
16	5.142	6.908	20.465	23.542	26.296	28.845	29.633	32.000	34.267	37.146	39.252
17	5.697	7.564	21.615	24.769	27.587	30.191	30.995	33.409	35.718	38.648	40.790
18	6.265	8.231	22.760	25.989	28.869	31.526	32.346	34.805	37.156	40.136	42.312
19	6.844	8.907	23.900	27.204	30.144	32.852	33.687	36.191	38.582	41.610	43.820
20	7.434	9.591	25.038	28.412	31.410	34.170	35.020	37.566	39.997	43.072	45.315

5) Логарифмическая доходность

Альтернативный способ измерения доходности. Может использоваться вместо процентной и имеет существенное преимущество в том, насколько большие данные можно сравнивать, используя его, при этом в итоге получая корректный результат.

$\ln \frac{P_k}{P_{k-1}} = \ln P_k - \ln P_{k-1}$ (так как по свойству логарифм частного равен разности логарифмов)

k – текущий временной промежуток, P_k – цена акции в момент времени k, P_{k-1} – цена акции за предшествующий период.

Помимо сказанной выше, логарифмическая доходность имеет другие ощутимые преимущества :

В первую очередь, имея низкую частоту выплат при такой оценке легко объединять доходы с имеющими высокую частоту. В итоге ежемесячная доходность будет складываться из ежедневных значений. Такой подход позволяет упростить вычисления по сравнению с процентной оценкой. Кроме того, при таком методе можно вычислить начальное значение стоимости ценной бумаги, которая сначала выросла в цене на x%, а после снизилась на равную величину процентов x. [3]

6) Критерий Колмогорова

Один из основных критериев, определяющих оценку достоверности гипотезы. С помощью него можно проверить, сходно ли утверждение с каким-то из классических законов распределения.

Так и считается наибольшая по модулю разница между фактическим значением заданной эмпирической функцией Fn(x) и теоретически ожидаемым значением функции распределения F(x).

Введём обозначение $m(x,\vec{x})$ – количество составляющих вектора $\vec{x} = (x_1, \dots, x_n)$, где $x_i < x$ для любого $x \in R^n$. В случае случайного вектора, формула не изменится, но принимать она будет значения дискретной случайной величины от 0 до n . Эмпирическая функция распределения, полученная из выборки X объёма ni сопоставленная с некоторой функцией F(x), будет определяться по следующей формуле:

$$\hat{F} = \hat{F}(x,\vec{x}) = \frac{m(x,\vec{x})}{n}. \qquad \hat{F} = \hat{F}(x,\vec{X}) = \frac{m(x,\vec{X})}{n}.$$

Вторая формула применяется для оценки функции F(x) по случайной выборке X.

Примечание: $\hat{F}(x,\vec{X})$ – функция случайного процесса, поскольку x принимает различные случайные значения, и в то же время, как $\hat{F}(x,\vec{x})$ является числовой функцией.

Формулу расстояния между 2 функциями можно задать следующим соотношением :

$$d = \sup_x \left|\hat{F}(x) - F(x)\right|. \quad (6.3)$$

Как и раньше, в зависимости от аргумента, d может быть числом (при функции $\hat{F}(x,\vec{x})$, а может быть случайной величиной, принимающей значения от 0 до 1, поскольку будет зависеть от $\vec{X}$ при наличии функции $\hat{F}(x,\vec{X})$.

Таким образом, опираясь на теоремы Колмогорова, существует предел функции F(x), если она непрерывна и выбрано неотрицательно значение u:

$$\lim_{n\to\infty} P\left(\sqrt{n}d(\vec{X}) < u\right) = K(u), \quad (6.4)$$

где

$$K(u) = \sum_{k=-\infty}^{\infty}(-1)^k e^{-2k^2u^2} \quad (6.5)$$

По утверждению этой теоремы также устанавливается согласованность с критической областью $\sqrt{n}d(\vec{x}) > u_a$, где u_a - корень уравнения K(u) = 1-a, уровень значимости задаётся через предел стремления к α при бесконечно возрастающем n. То есть можно утверждать, что появляется понятие асимптотического уровня значимости, которым является α. При таких условиях, а также при n > 20, имеет смысл применять данный критерий, получивший наименование Колмогорова. Если n < 20, настоящий уровень значимости будет существенно отличаться от номинального значения.

Если принять F(x), как некоторую теоретическую функцию, а $\hat{F}(x)$, как фактическую эмпирическую функцию распределения, то для вычисления расстояния между ними используется данная формула:

$$d(\vec{x}) = \max_{1\le i\le n}\left\{\left|\frac{i}{n} - F(x_{(i)})\right|, \left|\frac{i-1}{n} - F(x_{(i)})\right|\right\}, \qquad (6.6)$$

где $x_{(i)}$ - i-й член вариационного ряда

$x_{(1)} \le x_{(2)} \le \ldots \le x_{(n-1)} \le x_{(n)}$ [1].

Таблица. “Граничные значения для статистического критерия Смирнова-Колмогорова ”

α	0,15	0,10	0,05	0,03	0,01
$D_{nH}\,\alpha$	0,775	0,819	0,895	0,995	1,035

7) Критерий Фишера

Использование критерия Фишера обычно, когда требуется проверить соотношение двух дисперсий выборок нормального распределения. Для этого необходимо вычислить дисперсии случайных процессов, после чего по значению соотношения можно будет понять, выполняется критерий Фишера или нет.

По центральной предельной теореме, принято считать, что закон нормального распределения задаёт функции распределения котировок доходностей акций компаний. Иными словами, распределение суммарной величины всех логарифмов близко к нормальному, при условии, что количество дней достаточно велико. Чтобы убедиться в этом, смоделируем ситуацию, воспользовавшись критерием Колмогорова

В качестве начальных данных смоделируем по функциям нормального распределения две выборки случайных значений :

$$X_1, \dots\dots, X_m \sim N(\mu_x, \sigma_x^2),$$

$$Y_1, \dots\dots, Y_n \sim N(\mu_y, \sigma_y^2).$$

Допустим, что значения параметров $\mu_x, \sigma_x^2, \mu_y, \sigma_y^2$ нам известны. Тогда основной гипотезой будет являться утверждение, что $\sigma_x^2 = \sigma_y^2$ (H_0). В качестве дополнительной гипотезы выберем один из 3 вариантов:

1) $H_1 : \sigma_x^2 > \sigma_y^2$;
2) $H_1 : \sigma_x^2 < \sigma_y^2$;
3) $H_1 : \sigma_x^2 \neq \sigma_y^2$.

Теорема о построении критериев для проверки гипотезы с известным уравнением значимости α:

Если верна H_0, то

$$\frac{s_x^2}{s_y^2} \sim F(m-1, n-1),$$

где $s_x^2 = \sum_{i=1}^{m}(X_i - \bar{X})^2$, $s_y^2 = \sum_{i=1}^{n}(Y_i - \bar{Y})^2$, а F(m-1,n-1) – распределение Фишера с m-1 и n-1 степенями свободы [2].

Значение критерия Фишера для двух выборок данных будет вычисляться по формуле:

$$F = \frac{\max(var1, var2)^2}{\min(var1, var2)^2}$$

Где var1 – дисперсия первой выборки, var2 – дисперсия второй выборки

II. Практическая работа

1) Предварительная обработка и аналитика данных

Для начала работы необходимо подготовить данные. Источником послужит веб ресурс, содержащий глобальную финансовую аналитику, а также вся статистика котировок акций компаний [7] за прошлые годы. Данные о том, акции каких компаний входят в этот индекс, а также о других секторах рынка можно использовать с сайта Московской биржи [8], а также воспользоваться дополнительно данными Московского финансового дома [9] для дополнительной проверки статистики.

Сопоставим тикеры акций, входящих в состав Финансового сектора ММВБ, с названиями компаний в таблице:

Таблица 1.
Состав тикеров индекса MOEXFN, включающего в себя акции финансового сектора Мосбиржи

Тикер акции	Название компании
SBER	ПАО «Сбербанк России»
SBER_p	ПАО «Сбербанк России» (привилегированные акции)
CBOM	ПАО «Московский кредитный банк»
MOEX	ПАО «Московская биржа»
RENI	ПАО «Группа Ренессанс Страхование»
QIWIDR	АО «Киви»
BSPB	ПАО «Банк Санкт-Петербург»
SFIN	ПАО «ЭсЭфАй»
TCSGDR	TCS Group Holding PLC
VTBR	ПАО «Банк ВТБ»

Чтобы вычислить количество дней, в которые проводились торги акциями этой компании, используем столбец ‘Дата’, записанную в формате ddMMyyyy. Для этого сгруппируем имеющиеся данные этого столбца по годам и приведём их к таблице в удобном виде. Записи строк будут перечислять данные по одному тикеру, а столбцы будут содержать информацию по одному году. Всё сохраняем, как файл с расширением csv с разделителем между записями «;». Результаты данного преобразования сохранены и их можно посмотреть в таблице 2.

Таблица 2. Суммарное количество торговых дней для всех компаний.

	A	B	C	D	E	F	G	H
1		Ticker	2016	2017	2018	2019	2020	2021
2	0	BSPB	252	252	255	252	250	255
3	1	CBOM	252	252	255	252	250	255
4	2	QIWIDR	252	252	254	252	250	255
5	3	RENI	0	0	0	0	0	48
6	4	SBER	252	252	254	252	250	255
7	5	SBER_p	252	252	254	252	250	255
8	6	SFIN	243	252	254	252	250	255
9	7	TCSGDR	0	0	0	44	250	255
10	8	VTBR	252	252	254	252	250	255
11	9	MOEX	252	252	254	252	250	255
12								

Как можно заметить, среди данных компаний есть котировки акций, которые начали продаваться позже 2016 года. Так у RENI статистика начинается только в конце 2021 года, также и у TCSGDR только с конца 2019 года. Их данные не будут полезны для анализа общего положения индекса фондового рынка, поэтому далее в статистике они рассматриваться не будут.

Сформируем таблицу, в которой будут присутствовать только компании, которые стабильно продавались большую часть года в рассматриваемый временной промежуток – с 2016 по 2021 годы. Результат просеивания данных можно наблюдать в таблице 3.

Таблица 3. Суммарное количество торговых дней для отредактированного списка компаний

	A	B	C	D	E	F	G	H
1		Ticker	2016	2017	2018	2019	2020	2021
2	0	BSPB	252	252	255	252	250	255
3	1	CBOM	252	252	255	252	250	255
4	2	QIWIDR	252	252	254	252	250	255
5	3	SBER	252	252	254	252	250	255
6	4	SBER_p	252	252	254	252	250	255
7	5	SFIN	243	252	254	252	250	255
8	6	VTBR	252	252	254	252	250	255
9	7	MOEX	252	252	254	252	250	255

Список, который можно увидеть выше, содержит 8 компаний, каждая из которых на протяжении последних 6 лет имела более 229 торговых, поэтому на основании этих данных можно продолжить анализ гипотезы и преобразование данных к нужному виду.

Теперь выделим для каждой акции максимальный по возрастанию и по убыванию относительный скачок цены. Для этого используем данные, которые содержаться в столбце “Цена” – стоимость акции, сформированная в момент закрытия. После применения функции, рассчитывающей рост значения, относительно предыдущего, мы получаем 2 таблицы, которые сохранены также в формате csv файла. Чтобы цифры выглядели более понятными, откроем файл в Microsoft Excel и применим к данным условное форматирование.

В таблице 4 приведены максимальные относительные скачки цен вниз. Вариация идёт от меньшего по модулю значения (Зелёный) к большему (Красный)

Таблица 4. Максимальный относительный рост цены вверх

	A	B	C	D	E	F	G	H	I
1		Ticker	2016	2017	2018	2019	2020	2021	
2	0	BSPB	0,072	0,06	0,087	0,052	0,056	0,06	
3	1	CBOM	0,035	0,029	0,03	0,061	0,091	0,042	
4	2	QIWIDR	0,094	0,094	0,103	0,242	0,11	0,1	
5	3	SBER	0,064	0,063	0,08	0,03	0,129	0,057	
6	4	SBER_p	0,055	0,071	0,076	0,028	0,084	0,053	
7	5	SFIN	0,03	0,128	0,139	0,116	0,12	0,049	
8	6	VTBR	0,09	0,08	0,058	0,101	0,086	0,072	
9	7	MOEX	0,052	0,039	0,05	0,043	0,092	0,042	
10									

Далее рассмотрим обратную ситуацию. Посчитаем максимальные относительные скачки цен вниз. Результаты занесём в таблицу 5, где также применим условное форматирование Excel таблиц. Вариация идёт от меньшего по модулю значения (Зелёный) к большему (Красный)

Таблица 5. Максимальный относительный рост цены вниз

	A	B	C	D	E	F	G	H	I
1		Ticker	2016	2017	2018	2019	2020	2021	
2	0	BSPB	-0,046	-0,084	-0,077	-0,057	-0,127	-0,068	
3	1	CBOM	-0,044	-0,025	-0,031	-0,04	-0,072	-0,028	
4	2	QIWIDR	-0,093	-0,069	-0,13	-0,07	-0,212	-0,068	
5	3	SBER	-0,062	-0,039	-0,17	-0,053	-0,096	-0,056	
6	4	SBER_p	-0,053	-0,047	-0,134	-0,061	-0,092	-0,046	
7	5	SFIN	-0,018	-0,079	-0,261	-0,077	-0,052	-0,035	
8	6	VTBR	-0,039	-0,062	-0,09	-0,04	-0,144	-0,065	
9	7	MOEX	-0,073	-0,056	-0,056	-0,058	-0,085	-0,046	
10									
11									

В итоге теперь мы располагаем всей статистикой значительных изменений цен выбранных акций .По данным таблиц можно выделить 2 котировки, стабильность которых стоит проверить : SFIN и QIWIDR. Для этого откроем файлы соответствующих акций и по данным в столбце “Цена” построим график изменения цены с течением времени.

Рисунок 1. График изменения цены акций компании SFIN.

Наибольшие скачки цен вниз были заметны в 2018 году (-26,1 %), однако данные случаи не выбиваются за общие границы значений выборки, а также не превышают 40 % в относительном изменении цены. Поэтому данные этих акций можно использовать для дальнейшего анализа.

Рисунок 2. График изменения цены акций компании QIWIDR.

Максимальный рост цены акции зафиксирован в 2019 году (24,2 %). Постоянные рост и падение цены акции в течение дня по своей величине не

имеют слишком большое значение, поэтому данную выборку также можно считать репрезентативной.

Таким образом, после предварительной обработки данных, у нас остались котировки 8 акций, которые подходят для дальнейшего анализа и исследования гипотезы, то есть имеют большое число (> 229) дней, в которые их акции участвовали в торгах на бирже и не имеют существенный разброс в стоимости. Выборка реальных данных, на которых будет проверяться исследуемая гипотеза, подготовлена и приведена к нужному виду.

2) Проверка гипотезы на сгенерированных данных

Проверим нашу гипотезу о равенстве дисперсий логарифмических доходностей фондового рынка и входящих в его состав акций для данных, которые будут сгенерированы по определённому образцу, в соответствии с критерием Колмогорова. Для этого используем метод Монте-Карло, создадим 2 выборки из 252 элементов каждая, после чего на их основе выясним: верна ли гипотеза, и правильно ли написана программа для вычисления критерия Фишера и P-значения.

Для наглядности выведем гистограмму P-значений, отдельно продемонстрировав каждое из распределений на графике. Так как мы проводим несколько экспериментов, и каждый раз получаем различные p-значения, стоит проверить, как часто встречаются определённые значения:

Рисунок 3. Первое распределение сгенерированных данных. Р-значения.

Рисунок 4. Второе распределение сгенерированных данных. Р-значения.

После соответствующих вычислений в среде разработки, получаем Р-значения, равные 0.41963114811849644 и 0.36498239642797903 для первой и второй выборки соответственно. Это показатель того, что Р-значения распределены по равномерному закону для таких модельных данных, так как р-значение по своей величине больше общепринятого критерия значимости (5%)

Теперь проверим эти данные критерием Фишера. Полученное значение: 1.0321616259272595. Результат свидетельствует о том, что дисперсии практически равные, а значит вывод о том, что гипотеза о равенстве дисперсий логарифмических доходностей фондового рынка и входящих в его состав акций полностью выполняется в идеальных случаях, когда используются модельные данные одинакового образца.

3) Проверка правильности гипотезы на реальной статистике

После проверки, что программа работает корректно, все данные обрабатываются в нужном формате, и котировки проверены и приведены к нужному виду, можно приступить к моделированию алгоритма на реальных данных. Итак, используем критерий Фишера, чтобы проверить верна ли гипотеза о равенстве дисперсий логарифмических доходностей фондового рынка и входящих в его состав акций на примере акций индекса MOEXFN –

индекса Московской биржи, включающий в себя финансовый сектор. Для вычислений, как и при предварительном анализе котировок, нам потребуются столбцы с данными “Дата” (день совершения сделок на бирже) и “Цена” (стоимость закрытия последней сделки с этой акцией в определённый день). Для того, чтобы проверить гипотезу, нам необходима логарифмическая доходность, данными о которой мы пока не располагаем в наших таблицах с данными. Вычислим эти значения и сохраним в списке с другими данными. Для этого используем уже существующие функции языка Python.

После этого необходимо вычислить P-значения (I раздел. Пункт 4) функции на основе реальных данных. Для этого мы можем использовать написанную ранее функцию, подготовить выборку данных за 6 лет, удовлетворяющую критерию Колмогорова. Итоговые значения вынесем для наглядности в таблицу.

Таблица 6. P-значения для настоящих данных за 6 лет в течение каждого года

Таблица. P-значения по критерию Колмогорова

10]:

	Тикер	2016	2017	2018	2019	2020	2021
0	BSPB	0.0	0.0	0.0	0.0	0.0	0.0
1	CBOM	0.0	0.0	0.0	0.0	0.0	0.0
2	QIWIDR	0.0	0.0	0.0	0.0	0.0	0.0
3	SBER	0.0	0.0	0.0	0.0	0.0	0.0
4	SBER_p	0.0	0.0	0.0	0.0	0.0	0.0
5	SFIN	0.0	0.0	0.0	0.0	0.0	0.0
6	VTBR	0.0	0.0	0.0	0.0	0.0	0.0
7	MOEX	0.0	0.0	0.0	0.0	0.0	0.0
8	MOEXFN	0.0	0.0	0.0	0.0	0.0	0.0

Также, предоставим гистограмму, отражающую полученные данные для наглядного отображения.

Рисунок 5. Распределение P-значений

Из-за того, что P-значения приближены к 0, нет оснований для выводов о том, какое распределение задаёт эти функции цены реальных акций компаний, в то числе исключается и равномерное распределение. Следовательно, для каждой из котировок акций компаний вычисляется отдельно значение дисперсии, чтобы потом провести все финальные расчёты.

Таблица 7. Данные о дисперсии логарифмических доходностей за 6 лет по каждому году

Таблица. Дисперсии логарифмической доходности по годам

	Тикер	2016	2017	2018	2019	2020	2021
0	BSPB	0.017888	0.009437	0.006949	0.003200	0.015068	0.024605
1	CBOM	0.002226	0.000771	0.000937	0.001772	0.002109	0.001813
2	QIWIDR	0.014999	0.031903	0.007078	0.028793	0.035955	0.015811
3	SBER	0.033116	0.017967	0.016176	0.003656	0.015230	0.007723
4	SBER_p	0.035845	0.028103	0.012094	0.004961	0.012988	0.007813
5	SFIN	0.006580	0.002167	0.012295	0.004028	0.004777	0.003341
6	VTBR	0.001562	0.009853	0.020372	0.009122	0.015274	0.015281
7	MOEX	0.011590	0.005767	0.013191	0.003866	0.025829	0.003400
8	MOEXFN	0.008887	0.003512	0.009493	0.002571	0.019172	0.014882

Так как мы не можем использовать прямое сравнение дисперсий при анализе данных, поэтому переходим сразу же к подсчёту значений критерия

Фишера для выбранного набора акций. Сравниваться каждая из них будет со значениями индекса Мосбиржи Финансового сектора с помощью уже написанной ранее функции.

Значение критерия Фишера для двух выборок данных будет вычисляться по формуле:

$$F = \frac{\max(var1, var2)^2}{\min(var1, var2)^2}$$

Где var1 – дисперсия первой выборки, var2 – дисперсия второй выборки

Таблица 8. Значения критериев Фишера для каждой акции

	A	B	C
1		Tiker	Fishre's criterion
2	0	BSPB	3.178346397793912
3	1	CBOM	2.678001624966286
4	2	QIWIDR	1.3497189616514969
5	3	SBER	2.5014456919440096
6	4	SBER_p	6.717130340083961
7	5	SFIN	1.9100465069461328
8	6	VTBR	1.4646741559184644
9	7	MOEX	1.3747144093076238
10			

Полученные значения для данного эксперимента свидетельствуют о том, что выбранная для проверки гипотеза выполняется для индекса Финансового сектора Московской биржи. Значения критерия Фишера отражают соотношение близкое к нормальному.

Наибольшее значение дисперсии логарифмической доходности у привилегированных акций Сбербанка стало причиной наибольшего значения критерия Фишера при проверке вычислений в данных. Далеко не все реальные данные могут проявляться таким же образом, так как ситуация в разных секторах экономики принципиально отличается. Финансовый сектор очень сильно централизован и следует общей политике государства, более того, он во многом зависим от её состояния, вследствие чего напрашивается вывод о

том, что предложенная для анализа гипотеза выполняется на этом наборе реальных данных.

4) Другие гипотезы и оценка мощности критерия

В качестве альтернативных вариантов могли подойти следующие распределения логарифмической доходности:

1. распределение по модулю закона Стьюдента с тремя степенями свободы;
2. логнормальное распределение.

Вычислим мощность критерия Фишера, вычисляя 1000 раз P-значения при уровне значимости $\alpha = 0.05$ для каждого альтернативного распределения. Примеры генерации и оценки таких экспериментов приведены в коде, прилагающимся к курсовой.

Вид распределения	20	125	252
Распределение по модулю закона Стьюдента с тремя степенями свободы	0.6406	0.9874	0.9998
Логнормальное распределение со стандартным отклонением равным 1/4	0.6603	0.9893	0.9989

Таблица 9. Мощность критерия Фишера для различных вычислений

На основе этих данных можно сделать вывод, что для логнормального распределения и распределения Стьюдента с тремя степенями свободы мощность критерия Фишера возрастает в соответствии с ростом размера выборки данных. Это означает, что при увеличении размера выборки уменьшается вероятность допустить ошибку второго рода.

IV. Заключение

При написании данной курсовой работы была поставлена цель – проверить гипотезу о равенстве дисперсий логарифмических доходностей индекса фондового рынка и входящих в его состав акций, с использованием методов сравнения и анализа, такие как критерий Фишера и критерий Колмогорова. За основу брались данные индекса Финансового сектора

Московской биржи (MOEXFN), а проверка проводилась как на случайно сгенерированных модельных данных, так и на реальной выборке котировок акций компаний.

Результатом анализа выполнимости выбранной гипотезы, мною были получены следующие результаты : гипотеза подтвердилась на модельных данных, сгенерированных согласно определённому принципу, и также подтвердилась на выбранном наборе реальных данных, предварительно прошедших просеивание и анализ репрезентативности их, как достоверных.

По данным анализа лишь одного сектора бизнеса не следует обобщать вывод о справедливости гипотезы, однако можно сказать, что она выполнима для финансового сектора, который представляет интересы общего рынка, а также основных его регуляторов.

Искажения в результатах работы и входе анализа данных могли быть вызваны различными недостатками в критериях, выбранных для проверки гипотезы. Так критерий Фишера очень чувствителен к отклонениям от нормального распределения в исследуемой выборке, а также к выборкам разного размера, которые гораздо чаще встречаются в реальных данных, в отличие от настроенных моделей.

Однако, фактическое выполнение критериев на модельных данных в диапазоне от 0 до 1 подтверждено мною в ходе проведения работы, а это означает, что проведённые вычисления не являются ошибочными и выбранные критерии обеспечивают достоверную проверку.

V. Список использованных источников

1. – М.: Финансы и статистика, 2013

2. Браилов А.В. Лекции по математической статистике. – М.: Финакадемия, 2008.

3. Браилов А.В. Лекции по теории вероятности. – М.: Финакадемия, 2008.

4. Глебов Криволапов Практикум по математической статистике. Проверка гипотез с использованием Excel, MatCale, R и Python. М.: Прометей, 2019.

5. Теория вероятностей и математическая статистика (для бакалавров): учебное пособие / Кацко И.А. ред. – Москва: КноРус, 2019. – 389с. – URL: https://www.book.ru/book/930219 (Дата обращения: 25.04.2021)

6. Смирнова З.М., Крейнина М.В. ПРОВЕРКА СТАТИСТИЧЕСКИХ ГИПОТЕЗ.

7. https://ru.investing.com/equities/

8. https://mfd.ru/

9. https://smart-lab.ru/q/index_stocks/MOEXFN/

VI. Приложения

Приложение 1. Технические характеристики персонального ноутбука

```
Технические характеристики компьютера:

Процессор Intel Core i7-9750H CPU

Тактовая частота 2.59 GHz 4.5 GHz

Частота системной шины 2400 МГц

Объём кэша второго уровня 1 536 Кб

Время работы программы:
--- 9.982163429260254 seconds ---
```

Приложение 2. Список файлов

Имя файла	Содержание
Таблица 1	Состав тикеров индекса MOEXFN, включающего в себя акции финансового сектора Мосбиржи
Таблица 2	Суммарное количество торговых дней для всех компаний.
Таблица 3	Суммарное количество торговых дней для отредактированного списка компаний
Таблица 4	Максимальный относительный рост цены вверх
Таблица 5	Максимальный относительный рост цены вниз
Таблица 6	P-значения для настоящих данных за 6 лет в течение каждого года
Таблица 7	Данные о дисперсии логарифмических доходностей за 6 лет по каждому году
Таблица 8	Значения критериев Фишера для каждой акции
Таблица 9	Мощность критерия Фишера для различных вычислений
Рисунок 1	График изменения цены акций компании SFIN.
Рисунок 2	График изменения цены акций компании QIWIDR.
Рисунок 3	Первое распределение сгенерированных данных. P-значения.
Рисунок 4	Второе распределение сгенерированных данных. P-значения.
Рисунок 5	Распределение P-значений

Приложение 3. Программный код на языке python, реализованный в среде Jupyter notebook

Стартовый импорт библиотек

In [329]:

```
import time
```

In [330]:

```
start_time = time.time()  #Переменная для подсчёта времени работы программы
```

In [331]:

```
#Импорт задействованных библиотек(технические)
import numpy as np
import pandas as pd
import csv
#Библиотеки для анализа различных статистических показателей
import matplotlib.pyplot as plt
import matplotlib
import statistics as st
import scipy.stats as stats
```

In [332]:

```
#Задаём стиль графиков
matplotlib.style.use('ggplot')
```

Подготовка данных для анализа

In [333]:

```
#Подсчёт общего числа рабочих дней в компаниях
ticker_names = ['BSPB','CBOM', 'QIWIDR', 'RENI', 'SBER', 'SBER_p', 'SFIN', 'TCSGDR', '\
all_days_amount = pd.DataFrame()
all_days_amount['Ticker']  =  ticker_names
years = [i for i in range(2016, 2022)]
for i in years:
    work_days_amount = []
    for cur_ticker in ticker_names:
        file = pd.read_csv(cur_ticker + '.csv', sep = ';')
        dlit = (file['Дата'] % 10000 >= i)&(file['Дата'] % 10000 < (i+1))
        amount = len(file[dlit])
        work_days_amount.append(amount)
    all_days_amount[str(i)] = work_days_amount

all_days_amount.to_csv('Общее количество торговых дней в году для всех компаний.csv', s
all_days_amount
```

Out[333]:

	Ticker	2016	2017	2018	2019	2020	2021
0	BSPB	252	252	255	252	250	255
1	CBOM	252	252	255	252	250	255
2	QIWIDR	252	252	254	252	250	255
3	RENI	0	0	0	0	0	48
4	SBER	252	252	254	252	250	255
5	SBER_p	252	252	254	252	250	255
6	SFIN	243	252	254	252	250	255
7	TCSGDR	0	0	0	44	250	255
8	VTBR	252	252	254	252	250	255
9	MOEX	252	252	254	252	250	255

In [334]:

```
#Подсчёт общего числа рабочих дней в компаниях, у которых сохраняется стабильное число
ticker_names = ['BSPB', 'CBOM', 'QIWIDR', 'SBER', 'SBER_p', 'SFIN', 'VTBR', 'MOEX']
all_days_amount = pd.DataFrame()
all_days_amount['Ticker']  =  ticker_names
years = [i for i in range(2016, 2022)]
for i in years:
    work_days_amount = []
    for cur_ticker in ticker_names:
        file = pd.read_csv(cur_ticker + '.csv', sep = ';')
        dlit = (file['Дата'] % 10000 >= i)&(file['Дата'] % 10000 < (i+1))
        amount = len(file[dlit])
        work_days_amount.append(amount)
    all_days_amount[str(i)] = work_days_amount

all_days_amount.to_csv('Общее количество торговых дней в году для выбранных компаний.cs
all_days_amount
```

Out[334]:

	Ticker	2016	2017	2018	2019	2020	2021
0	BSPB	252	252	255	252	250	255
1	CBOM	252	252	255	252	250	255
2	QIWIDR	252	252	254	252	250	255
3	SBER	252	252	254	252	250	255
4	SBER_p	252	252	254	252	250	255
5	SFIN	243	252	254	252	250	255
6	VTBR	252	252	254	252	250	255
7	MOEX	252	252	254	252	250	255

In [335]:

```
#Подсчёт относительных скачков цен вниз
ticker_names = ['BSPB', 'CBOM', 'QIWIDR', 'SBER', 'SBER_p', 'SFIN', 'VTBR', 'MOEX']

years = [i for i in range(2016, 2022)]

lower_cost = pd.DataFrame() #Создаём Data Frame дял сохранения данных о котировках по а
lower_cost['Тикер'] = ticker_names

for i in years:
    decreas_vol = []
    for cur_name in ticker_names:
        company_data = pd.read_csv(cur_name + '.csv', sep = ';')#Считываем данные котир
        company_data['Delta'] = company_data['Цена'].pct_change().round(3)#Считаем разн
        dlit = (company_data['Дата'] % 10000 >= i)&(company_data['Дата'] % 10000 < (i+1
        cur_year = company_data[dlit]['Delta'].min()
        decreas_vol.append(cur_year)
    lower_cost[str(i)] = decreas_vol

lower_cost.to_csv('Относительные изменения цен вниз.csv', sep = ';', decimal=',')
lower_cost

```

Out[335]:

	Тикер	2016	2017	2018	2019	2020	2021
0	BSPB	-0.046	-0.084	-0.077	-0.057	-0.127	-0.068
1	CBOM	-0.044	-0.025	-0.031	-0.040	-0.072	-0.028
2	QIWIDR	-0.093	-0.069	-0.130	-0.070	-0.212	-0.068
3	SBER	-0.062	-0.039	-0.170	-0.053	-0.096	-0.056
4	SBER_p	-0.053	-0.047	-0.134	-0.061	-0.092	-0.046
5	SFIN	-0.018	-0.079	-0.261	-0.077	-0.052	-0.035
6	VTBR	-0.039	-0.062	-0.090	-0.040	-0.144	-0.065
7	MOEX	-0.073	-0.056	-0.056	-0.058	-0.085	-0.046

In [336]:

```
#Подсчёт относительных скачков цен вверх
ticker_names = ['BSPB', 'CBOM', 'QIWIDR', 'SBER', 'SBER_p', 'SFIN', 'VTBR', 'MOEX']

years = [i for i in range(2016, 2022)]

upper_cost = pd.DataFrame() #Создаём Data Frame дял сохранения данных о котировках по а
upper_cost['Тикер'] = ticker_names

for i in years:
    incr_vol = []
    for cur_name in ticker_names:
        company_data = pd.read_csv(cur_name + '.csv', sep = ';')#Считываем данные котир
        company_data['Delta'] = company_data['Цена'].pct_change().round(3)#Считаем разн
        dlit = (company_data['Дата'] % 10000 >= i)&(company_data['Дата'] % 10000 < (i+1
        cur_year = company_data[dlit]['Delta'].max()
        incr_vol.append(cur_year)
    upper_cost[str(i)] = incr_vol

upper_cost.to_csv('Относительные изменения цен вниз.csv', sep = ';', decimal=',')
upper_cost

```

Out[336]:

	Тикер	2016	2017	2018	2019	2020	2021
0	BSPB	0.072	0.060	0.087	0.052	0.056	0.060
1	CBOM	0.035	0.029	0.030	0.061	0.091	0.042
2	QIWIDR	0.094	0.094	0.103	0.242	0.110	0.100
3	SBER	0.064	0.063	0.080	0.030	0.129	0.057
4	SBER_p	0.055	0.071	0.076	0.028	0.084	0.053
5	SFIN	0.030	0.128	0.139	0.116	0.120	0.049
6	VTBR	0.090	0.080	0.058	0.101	0.086	0.072
7	MOEX	0.052	0.039	0.050	0.043	0.092	0.042

Создание модели данных и проверка гипотезы на них

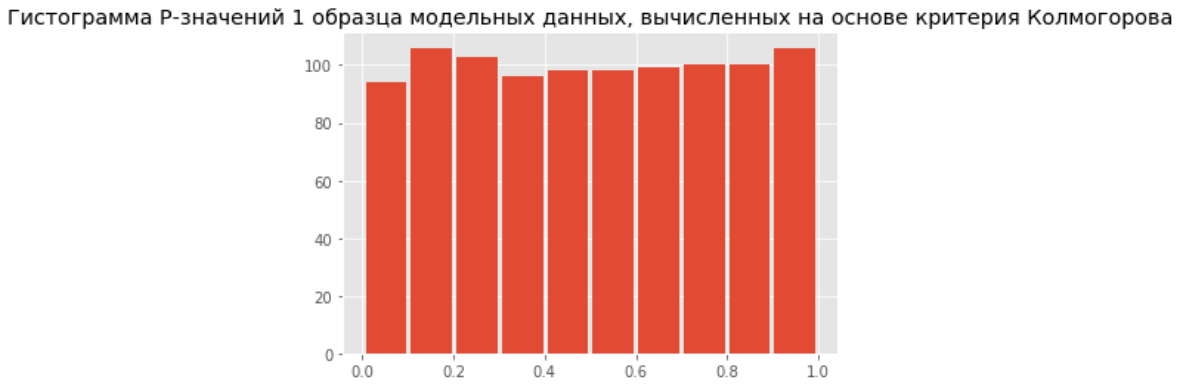
In [337]:

```
#Функция проверки критерия Колмогорова
def kolm_krit(n):
    data = np.random.normal(loc = 0, scale = 1, size = n) #Задаём параметры генерации н
    result = stats.kstest(data,'norm')
    return data, round(result[1],3)

#Размер выборки - среднее число торговых дней в год
n = 252
```

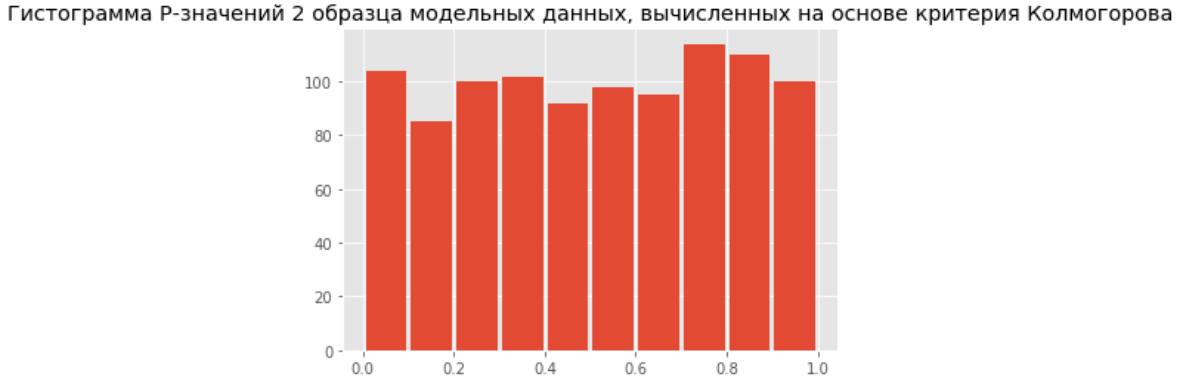
In [339]:

```
#построение гистограммы P-значений, вычисленных с помощью критерия Колмогорова. (1 выбо

P_values_1 = [] # Пустой массив для сохранения P-значений выборки
tetst_data_1 = kolm_krit(n)[0] # Генерация данных на основании критерия Колмогорова
print(stats.kstest(tetst_data_1, 'norm'))

for i in range(10 ** 3):
    P_values_1.append(kolm_krit(n)[1])#Получаем P-значения с помощью написанного модуля

#Отображение данных (из предыдущей ячейки) в формате гистограммы
plt.hist(P_values_1, rwidth = 0.9)
plt.title('Гистограмма P-значений 1 образца модельных данных, вычисленных на основе кри
plt.show()
#построение гистограммы P-значений, вычисленных с помощью критерия Колмогорова. (2 выбо

P_values_2 = [] # Пустой массив для сохранения P-значений выборки
tetst_data_2 = kolm_krit(n)[0] # Генерация данных на основании критерия Колмогорова
print(stats.kstest(tetst_data_2, 'norm'))

for i in range(10 ** 3):
    P_values_2.append(kolm_krit(n)[1])#Получаем P-значения с помощью написанного модуля

#Отображение данных (из предыдущей ячейки) в формате гистограммы
plt.hist(P_values_2,rwidth = 0.9) #Создаём макет гистограммы и передаём ей данные и пар
plt.title('Гистограмма P-значений 2 образца модельных данных, вычисленных на основе кри
plt.show()
```

```
KstestResult(statistic=0.05484159073843625, pvalue=0.41963114811849644)
```




```
KstestResult(statistic=0.05733722721489448, pvalue=0.36498239642797903)
```



In [353]:

```
# Проверим, имеют ли сгенерированных 2 набора данных общий вид
compar1 = stats.ks_2samp(data_1, data_2)
print(compar1[1])
```

```
0.8916529782426248
```

In [354]:

```
#Функция проверки критерия Фишера
def fisher_crit(var1, var2):
    return ((max(var1, var2))**2/(min(var1, var2))**2)
```

In [355]:

```
#Вычислим дисперсию 2 выборок данных
var_data_1 = np.var(data_1)
var_data_2 = np.var(data_2)
print("Дисперсия первой выборки = " + str(var_data_1), '\nДисперсия второй выборки = '

res_fisher = fisher_crit(var_data_1, var_data_2)
print("Критерий Фишера = ", res_fisher)
```

```
Дисперсия первой выборки = 0.9312889089253301
Дисперсия второй выборки = 0.9166648458853652
Критерий Фишера =  1.0321616259272595
```

In [343]:

```
#Проверка мощности критерия
m = 10000 #количество экспериментов
res1 = []
res2 = []
for i in range(m):
    data_11 = (stats.t.rvs(df = 3, loc = 0, scale = 1,size = 252))
    data_12 = (stats.t.rvs(df = 3, loc = 0, scale = 1,size = 252))
    data_21 = (stats.lognorm.rvs(s = 1/4 ,loc = 0, scale = 1,size = 252))
    data_22 = (stats.lognorm.rvs(s = 1/4 ,loc = 0, scale = 1,size = 252))
    var_data_11 = np.var(data_11)
    var_data_12 = np.var(data_12)
    var_data_21 = np.var(data_21)
    var_data_22 = np.var(data_22)
    res1.append(fisher_crit(var_data_11, var_data_12))
    res2.append(fisher_crit(var_data_21, var_data_22))

pow1 = 0
pow2 = 0
## Вычисление мощности критерия
for i in res1:
    if i < 0.05:
        pow1 += 1
print(pow1 / m)

for i in res2:
    if i < 0.05:
        pow2 += 1
print(pow2 / m)

```

```
0.0
0.0
```

Проверка гипотезы на подготовленных реальных данных фондового рынка

In [344]:

```
#импорт таблиц

def to_table(ticker): # Функция, выполняющая считывание файла(входной параметр ticker)
    company = pd.read_csv(ticker + '.csv', sep = ';') #Открываем файл с котировками аку
    company['Дата'] = company['Дата'].astype('object')
    for i in range(len(company['Дата'])):
        company['Дата'][i] = str(company['Дата'][i])
        if len(company['Дата'][i]) < 8:
            company['Дата'][i] = '0' + company['Дата'][i]
    company['Дата'] = pd.to_datetime(company['Дата'], format = '%d%m%Y') #преобразовани
    return company

def logTransf(ticker, num):
    LD = []
    all_years = ['2016', '2017', '2018', '2019', '2020', '2021']
    company = to_table(ticker)
    company['Логарифм доходности'] = np.log(company['Цена'])

    all_days_table = pd.read_csv('Общее количество торговых дней в году для выбранных к
    all_days_table = all_days_table.drop(columns = all_days_table.iloc[:, range(1)]) #У

    for year in all_years:
        n = (all_days_table[str(year)][num]) #Количество дней в текущем проверяемом год

        first_index = company['Логарифм доходности'][company['Дата'] <= str(year)+'-12-
        #Незабываем сохранить данные по текущему году, чтобы потом целиком скинуть всю
        LD.append((company['Логарифм доходности'][company['Дата'] <= year + '-12-31'])[

    return LD


ticker_names = ['BSPB', 'CBOM', 'QIWIDR', 'SBER', 'SBER_p', 'SFIN', 'VTBR', 'MOEX']
PV = []
for cur_ticker in ticker_names:
    num = ticker_names.index(cur_ticker)
    log = logTransf(cur_ticker, num)

```

In [345]:

```
def calc_log_max(file, year = 0):
    #Функция для расчёта логарифмической доходности.
    # Параметры - название файла (file) и год, за который требуется посчитать доходност
    #при значении 0 считает за все годы

    company = pd.read_csv(file, sep=';') #считывание файла csv по определённому тикеру
    company['Дата'] = [int(str(company['Дата'][i])[4:]) for i in range(len(company['Дат

    #Далее через встроенную функцию логарифма, нормируем значения выборки, используя то

    if year > 0: #Расчёт логарифмической доходности при заданном параметре год
        log_max_res = np.log(company['Цена'][company['Дата'] <= int(year)][company['Дат
    else:
        log_max_res = np.log(company['Цена'])
    return [*stats.kstest(log_max_res, 'norm'), np.var(log_max_res)] #возвращаем 3 пара

```

In [346]:

```
#Список всех рассматриваемых тикеров компаний
all_suitable_tickers = ['BSPB', 'CBOM', 'QIWIDR', 'SBER', 'SBER_p', 'SFIN', 'VTBR', 'MC

#Временной промежуток, на котором у нас есть данные: с 2016 по 2021
years = [i for i in range(2016,2022)]

#Создание дата фрейма для дальнейших вычислений
table_log_max = pd.DataFrame()

#Столбец с наименованиями тикеров
table_log_max['Тикер'] = all_suitable_tickers
```

In [347]:

```
#Вычисляем, какой был наибольший скачок стоимостей акций при группировке по годам
all_data = [] #Те же данные в формате массива, чтобы их можно было показать на гистогра
for year in years:
    log_cur_max = [] #Список для хранения значения по всем тикерам в течение одного год
    for ticker in all_suitable_tickers:
        #Добавление в список значение макс.логарифмических доходностей
        #Подсчёт идёт по каждому тикеру за все дни в течение одного года, после чего чи
        log_cur_max.append(round(calc_log_max(ticker+'.csv', year)[1], 10))
        all_data.append(round(log_cur_max[-1], 10))

    #После вычисления всех данных по тикерам в течение одного года (log_number_max)
    #Сохраняем в таблицу (table_log_max) все данные для общей статистики
    table_log_max[str(year)] = log_cur_max #группировка в данной таблице идёт по годам

#Заголовок перед таблицей с данными (выделяем жирным шрифтом)
print('\033[1m' + 'Таблица. P-значения по критерию Колмогорова')

table_log_max
```

Таблица. P-значения по критерию Колмогорова

Out[347]:

	Тикер	2016	2017	2018	2019	2020	2021
0	BSPB	0.0	0.0	0.0	0.0	0.0	0.0
1	CBOM	0.0	0.0	0.0	0.0	0.0	0.0
2	QIWIDR	0.0	0.0	0.0	0.0	0.0	0.0
3	SBER	0.0	0.0	0.0	0.0	0.0	0.0
4	SBER_p	0.0	0.0	0.0	0.0	0.0	0.0
5	SFIN	0.0	0.0	0.0	0.0	0.0	0.0
6	VTBR	0.0	0.0	0.0	0.0	0.0	0.0
7	MOEX	0.0	0.0	0.0	0.0	0.0	0.0
8	MOEXFN	0.0	0.0	0.0	0.0	0.0	0.0

In [348]:

```
#Отображение данных (из предыдущей ячейки) в формате гистограммы
plt.hist(all_data,rwidth = 0.9) #Создаём макет гистограммы и передаём ей данные и парам
plt.title('Гистограмма P-значений реальных данных, вычисленных с помощью критерия Колмо
plt.show()
```

In [349]:

```
#Расчёт максимального скачка цены (по годам)
for year in years:
    log_cur_max = [] #Список для хранения значения по всем тикерам в течение одного год
    for ticker in all_suitable_tickers:
        #Добавление в список значение макс.логарифмических доходностей
        #Подсчёт идёт по каждому тикеру за все дни в течение одного года, после чего чи
        log_cur_max.append(round(calc_log_max(ticker + '.csv', year)[2], 10))

    #После вычисления всех данных по тикерам в течение одного года (log_number_max)
    #Сохраняем в таблицу (table_log_max) все данные для общей статистики
    table_log_max[str(year)] = log_cur_max

#Заголовок перед таблицей с данными (выделяем жирным шрифтом)
print('\033[1m' + 'Таблица. Дисперсии логарифмической доходности по годам')

table_log_max
```

Таблица. Дисперсии логарифмической доходности по годам

Out[349]:

	Тикер	2016	2017	2018	2019	2020	2021
0	BSPB	0.017888	0.009437	0.006949	0.003200	0.015068	0.024605
1	CBOM	0.002226	0.000771	0.000937	0.001772	0.002109	0.001813
2	QIWIDR	0.014999	0.031903	0.007078	0.028793	0.035955	0.015811
3	SBER	0.033116	0.017967	0.016176	0.003656	0.015230	0.007723
4	SBER_p	0.035845	0.028103	0.012094	0.004961	0.012988	0.007813
5	SFIN	0.006580	0.002167	0.012295	0.004028	0.004777	0.003341
6	VTBR	0.001562	0.009853	0.020372	0.009122	0.015274	0.015281
7	MOEX	0.011590	0.005767	0.013191	0.003866	0.025829	0.003400
8	MOEXFN	0.008887	0.003512	0.009493	0.002571	0.019172	0.014882

In [350]:

```
#Расчёт значений дисперсии по каждому тикеру за всё время
list_var = []

for ticker in all_suitable_tickers:
    #Добавление в список значение макс.логарифмических доходностей
    #Подсчёт идёт по каждому тикеру за все дни измеряемого промежутка, после чего число
    list_var.append(round(calc_log_max(ticker + '.csv')[2], 10))

    #После вычисления всех данных по тикеру
    #Сохраняем в список (table_log_max) все данные для общей статистики

#Заголовок перед таблицей с данными (выделяем жирным шрифтом)
print('Таблица. Дисперсии логарифмической доходности\n')

for i in range(len(all_suitable_tickers)):
    print(all_suitable_tickers[i], ':', list_var[i]) #Выводим дисперсию вместе с назван
```

```
Таблица. Дисперсии логарифмической доходности

BSPB : 0.0300476109
CBOM : 0.0327344571
QIWIDR : 0.0461093379
SBER : 0.0847239314
SBER_p : 0.1388361137
SFIN : 0.0387604303
VTBR : 0.0648307536
MOEX : 0.0456882284
MOEXFN : 0.0535686326
```

In [351]:

```
#Создаю Дата фрейм для сохранения значений критерия Фишера
result = pd.DataFrame()
result['Tiker'] = all_suitable_tickers[:-1]  #Переносим список всех тикеров кроме инде
result["Fishre's criterion"] = all_suitable_tickers[:-1] #Столбец для значений критери

for i in list_var[:-1]:
    y = fisher_crit(i, list_var[-1]) #По уже написанной функции считаем значение критер
    result["Fishre's criterion"][list_var.index(i)] = y  #Сохраняем в строке соответст

result.to_csv('Значения критерия Фишера для всех выбранных акций.csv', sep = ';')

print('\033[1m' + 'Таблица. Значения критерия Фишера')
result
```

Таблица. Значения критерия Фишера

Out[351]:

	Tiker	Fishre's criterion
0	BSPB	3.17835
1	CBOM	2.678
2	QIWIDR	1.34972
3	SBER	2.50145
4	SBER_p	6.71713
5	SFIN	1.91005
6	VTBR	1.46467
7	MOEX	1.37471

In [352]:

```
finish_time = time.time()
#Общее время работы программы, включая все операции анализа, импорта библиотек и генера
print("--- %s seconds ---" % (finish_time - start_time))
```

```
--- 10.82125997543335 seconds ---
```

In [ ]:

```

```