

ПЕРМСКИЙ ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО
АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет экономики, менеджмента и бизнес-информатики

Мамаев Вадим Николаевич

**КОЛИЧЕСТВЕННЫЙ АНАЛИЗ ДЕЯТЕЛЬНОСТИ РОССИЙСКИХ ПАЕВЫХ
ИНВЕСТИЦИОННЫХ ФОНДОВ**

Выпускная квалификационная работа - МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
по направлению подготовки 38.04.08 Финансы и кредит
образовательная программа «Финансы»

Рецензент
К.э.н.
П.Ю. Малышев

Руководитель
К.э.н.
П.А. Паршаков

Консультант
И.Ю. Захаров

Пермь 2018

Оглавление

Введение.....	3
Обзор литературы.....	6
1. Данные.....	9
1.1. Сбор данных	9
1.2. Подготовка данных	21
2. Моделирование.....	30
2.1. Построение бенчмарка на основе индексных фондов	30
2.2. Построение инвестиционного портфеля функцией в R.....	35
2.3. Инвестиционный портфель на основе акций из Индекса ММВБ 10.....	49
2.4. Инвестиционный портфель на основе фьючерса на Индекс МосБиржи	53
3. Бутстрап модель	61
3.1. Описание бутстрап подхода.....	61
3.2. Алгоритм бутстрап подхода	63
3.3. Применение бутстрап подхода	66
3.4. Инвестиционная стратегия на основе бутстрап подхода.....	74
Заключение	89
Список литературы	92
Приложение	94

Введение

«Паевой инвестиционный фонд – обособленный имущественный комплекс, состоящий из имущества, переданного в доверительное управление управляющей компании учредителем (учредителями) доверительного управления с условием объединения этого имущества с имуществом иных учредителей доверительного управления, и из имущества, полученного в процессе такого управления, доля в праве собственности на которое удостоверяется ценной бумагой, выдаваемой управляющей компанией.» (Федеральный закон № 156-ФЗ от 29 ноября 2001 года «Об инвестиционных фондах») [4]

У граждан есть несколько инструментов для того, чтобы сохранить и приумножить свои сбережения, по-другому – инвестировать. Есть несколько способов, первый из них и самый простой – это вклад в банке с фиксированной процентной ставкой, при этом вклад застрахован Агентством по страхованию вкладов на сумму до 1,4 миллиона рублей. Недостаток данного способа – низкий процент доходности. Другой способ инвестирования может быть открытие индивидуального инвестиционного счёта (ИИС), который является обычным брокерским счётом, но с минимальным сроком на 3 года, при этом доступны две льготы, одна из них – это налоговый вычет в размере 13% от инвестированной суммы, другая: освобождение от уплаты налога на прибыль в размере 13%, по прибыли по данному счёту. В данном случае гражданин должен сам выбирать объекты инвестирования: акции, облигации и другие производные финансовые инструменты, доступные на фондовом рынке. При данном способе есть возможность получить более высокую доходность, но можно получить убыток. Также гражданину, не являясь квалифицированным инвестором, будет сложно выбрать правильные объекты инвестирования.

На практике паевой инвестиционный фонд (ПИФ) является инструментом коллективного инвестирования, при этом управляющая компания принимает

решение об объектах инвестирования. При выборе данного инструмента инвестирования выбор объектов инвестирования осуществляет квалифицированный инвестор. Вкладчик может получить более высокую доходность, чем вклад в банке, но риск меньше, чем инвестировать напрямую на фондовом рынке.

По данным Банка России на конец 2017 года стоимость чистых активов (СЧА) паевых инвестиционных фондов составляет 2 579,9 млрд. рублей [3]. Основную долю составляют закрытые фонды, 94,2% от совокупного СЧА, и объектом инвестирования этих фондов как правило является недвижимость. Для граждан лучше подходят открытые паевые инвестиционные фонды (ОПИФ), доля которых составляет 5,1% от совокупного СЧА фондов или 130,8 млрд. рублей, оставшаяся часть приходится на интервальные инвестиционные фонды. Крупнейшие сегменты открытых фондов – фонды облигаций и акций. Доля фондов облигаций составляет 51,5%, средневзвешенная доходность составила более 5,3%, при этом индекс корпоративных облигаций (MICEXCBITR) за тот же год вырос на 10,8%. Доля сегмента фондов акций составляет 27,5% от общего числа ОПИФ, средневзвешенная доходность равняется 18%, за год Индекс МосБиржи (IMOEX) вырос на 26,8%.

Доходность ПИФа зависит от того, какие объекты инвестирования выберет управляющий, на основании своего профессионального опыта, а значит, следует оценивать фонды с точки зрения навыков управляющих. В работах Kosowski et al (2006) и Fama, French, (2010) авторы предложили рассматривать эффективность фонда как отделение удачи от навыка управляющего. Данный случай демонстрируется следующим примером, если фонд получает доходность 10% в год, то можно говорить, что прибыль была получена благодаря профессиональным навыкам управляющего. Но если ещё 100 фондов получили такую же доходность, значит, что на рынке сложилась такая ситуация, что данную доходность было легко заработать, и в данном случае это будет называться удачей.

Д. М. Муравьев (2006) и П.А. Паршаков (2014) на основе данных работ анализировали российский рынок ПИФов. В работах данных авторов также поднимался вопрос о выборе бенчмарка для анализа российского фондового рынка, так как он относится к категории развивающихся. Что в свою очередь может свидетельствовать о том, что на данном рынке легко получить случайную доходность и как следствие можно легко «обыграть» бенчмарк – Индекс МосБиржи.

Цель данной работы заключается в определении паевых инвестиционных фондов, которые могут получать доходность относительно других участников рынка не случайно, а при помощи навыков управляющего фондом. Для достижения поставленной цели необходимо решить задачу автоматического получения данных по фондам, а в частности: историческую стоимость пая, величину стоимости чистых активов и информацию об издержках фонда. Далее необходимо определить бенчмарк, который бы в большей мере характеризовал ту доходность, которую можно получить на российском фондовом рынке, и произвести его расчёты. В результате нужно определить: способны ли управляющие российских паевых фондов обыграть данный бенчмарк и это будет говорить о том, что у данных фондов есть способность к получению безрисковой доходности на российском фондовом рынке.

Данная работа с одной стороны позволит определить есть ли такие фонды, которые способны «обыграть рынок». С другой стороны, в работе будет предложен подход, который может быть применён для тестирования торговых стратегий на фондовом рынке.

Обзор литературы

Впервые задача отделения навыка управляющего от удачи рассматривалась в работе Kosowski et al (2006) [6]. В данной работе был предложен бутстрап метод, который устойчив к гетероскедастичности временного ряда, автокорреляции в доходностях фондов. Также данный метод позволяет учесть ненормальность распределения доходностей фондов, которая получается, как результат неравномерного принятия рисков (heterogenous risk-taking) и ненормального распределения альфы отдельных фондов. В данной работе используются данные по взаимным фондам США за период с 1975 по 2002 год. Авторы приходят к выводу, что малое количество менеджеров фондов могут получать доходность за счёт навыков по сравнению с другими участниками рынка. В частности, в работе было обнаружено, что значительное меньшинство менеджеров выбирает акции достаточно хорошо, чтобы покрыть их расходы. Более того, превосходные альфы этих менеджеров сохраняются в течение периода времени.

Коэффициент альфа – параметр, с помощью которого можно точно оценить эффективность управления капиталом при учете рисков составляющей. Коэффициент альфа отображает, насколько результаты работы фонда на рынке зависят от навыков управляющего, а не от ситуации на фондовом рынке. Математически коэффициент альфа позволяет вычислить отношение параметра доходности интересующего фонда (актива) с потенциальной прибылью от эталонного портфеля, основой которого является индекс. С помощью коэффициента альфа инвестор точно определяет уровень риска, который необходим для получения интересующей его доходности. Коэффициент альфа показывает, насколько качественно управляющий работает с имеющимися в распоряжении активами и можно ли ему доверять.

Недостатком коэффициента альфа является то, что он не даст точный ответ, действительно ли талантлив управляющий фондом или же ему просто

везет. Часто бывает, что управляющие с положительной альфой сегодня уже через время теряют свои позиции и уходят в «минус». Именно данный недостаток решает бутстрап подход. Бутстрап модель предназначена для того, чтобы сравнить фонды по коэффициенту альфа. Альфа коэффициент для фонда получается путём регрессии из модели CAPM. Далее строится симулированная доходность фонда, где неизменными показателями остаются бета коэффициент и избыточная доходность бенчмарка. В данном подходе остатки для симулированной доходности одного фонда перемешиваются с заменой только для этого фонда. Далее извлекается симулированный коэффициент альфа и по всем фондам берётся квантиль 95% лучших коэффициентов. В конце реальные коэффициенты альфа фондов сравниваются с полученной выборкой симулированных альфа, и на основе этого отбираются лучшие фонды.

В работе Cuthbertson et al (2010) авторы также использовали бутстрап подход для оценки взаимных фондов США и Великобритании [5]. Они пришли к выводу, что существует около 0-5% лучших паевых инвестиционных фондов с действительно положительной характеристикой альфа и около 20% фондов, которые имеют поистине плохие альфа коэффициенты, и 75% имеют эффективность на уровне коэффициента альфа равного нулю. Также авторы продемонстрировали, что результаты для паевых фондов облигаций аналогичны результатам для фондов акций. Основной причиной низких показателей авторы выделили издержки на работу фонда и затраты на перебалансировку портфеля.

В следующей работе Fama, French (2010) использовали данные по взаимным фондам США с 1984 по 2006 год и пришли к выводу, если есть менеджеры фондов с достаточным навыком для создания ориентированных на результат ожидаемых доходностей, которые покрывают затраты, то они скрыты в совокупных результатах из-за менеджеров с недостаточными навыками [7]. Отличие бутстрап подхода в данной статье, от модели в Kosowski et al (2006) в том, что остатки для симулированной доходности перемешиваются с заменой по всем фондам сразу.

Применение бутстрап подхода для российских паевых инвестиционных фондов впервые было описано в работе Д.М. Муравьев (2006) [1]. Автор пришёл к выводу, что в России существует большое число фондов, управляющие которых обладают навыками к неслучайному опережению Индекса ММВБ. Также автор отмечает, что задача отделения случайности от навыка не до конца решена, так как нужно учитывать становые особенности развивающегося рынка. Автор ставит под сомнение валидность Индекса ММВБ в качестве бенчмарка для оценки способности управляющего, так как данный индекс можно легко обыграть.

Другое исследование по российским инвестиционным фондам проводил П.А. Паршаков (2014) [2]. В работе было рассмотрено 383 фонда за период с 2001 по 2013 год, а также был использован более подходящий для российского рынка бенчмарк MSCI Russia (MSCI Russia Standard Index) для акций, который рассчитывается аналитической компанией Morgan Stanley Capital International Inc. А для фондов облигаций – индекс MICEX CORP BOND TR (Индекс корпоративных облигаций ММВБ). В работе также рассматривается влияние частоты данных на показатели навыка управляющего. Автор приходит к выводу, что только 13% из всех фондов акций получили доходность, которая объясняется навыками управляющего.

1. Данные

1.1. Сбор данных

Для исследования в данной работе необходимы данные по не отраслевым и индексным паевым инвестиционным фондам, Индексу МосБиржи, Индексу ММВБ 10, безрисковой доходности, исторической стоимости акций на Московской Бирже, фьючерсу на Индекс МосБиржи. Получение данных будет осуществляться автоматически с использованием программы на языке R с информационного портала Investfunds (<http://pif.investfunds.ru/>) [8].

В самом начале в программе нужно подключить библиотеки, которые понадобятся в дальнейшем: RSelenium позволяет автоматизировать действия в веб-браузере, wdman является легковесным браузером [9], xts необходим для формата данных, который работает с временным массивом данных, при помощи gdata можно скачивать файлы, в которых хранятся нужные данные.

```
require(RSelenium) # инструмент для автоматизации
require(wdman)     # браузер
require(xts)       # формат временных данных
require(gdata)     # скачивание файлов
```

Рис. 1. Подключение библиотек в программе R

Далее следует запуск сервера и браузера для открытия информационного портала – сайта. В качестве браузера используется PhantomJS, который написан на JavaScript и быстро работает вследствие отсутствия графического интерфейса для пользователя, как например у обычных веб-браузеров: Internet Explorer, Google Chrome, Mozilla Firefox и другие. Все манипуляции выполняются путём отправки соответствующих команд, а сам браузер запущен как отдельный процесс в фоновом режиме. Команды от пользователя обрабатывает Selenium, распознаёт их, передаёт браузеру, затем браузер возвращает результат Selenium, который уже возвращается обратно пользователю [10]. Но в самом начале нужно прописать для Selenium параметры для взаимодействия с браузером, где необходимо указать адрес

сервера – в данном случае это локальный компьютер пользователя и поэтому задаётся значение “localhost”, далее номер порта, через который будет происходить отправка и получение данных в сети интернет, а также название запущенного браузера. После задания параметров происходит запуск сервера и его подключение к браузеру.

```
pjs <- phantomjs() # запуск сервера - браузера
pjs$output()
rem <- remoteDriver(remoteServerAddr = "localhost"
                    , port = 4567L
                    , browserName = "phantomjs"

) # описание подключения для Selenium
rem$open() # подключение к браузеру
rem$getStatus()
```

Рис. 2. Запуск сервера и браузера

После вышеперечисленных действий происходит взаимодействие с информационным порталом по получению списка нужных паевых инвестиционных фондов. Создаётся два массива `names` и `links` для хранения наименований фондов и соответствующей ссылки на данный фонд на портале Investfunds. Для открытия соответствующей страницы на сайте с фондами, которые удовлетворяют определенным критериям, используется ссылка, в которой заданы соответствующие параметры. Параметры указываются после символа “?” – это стандартный способ, используемый в URL адресах. Параметр `page=funds` указывает на паевые инвестиционные фонды; `type[]` – определяет тип фонда, который может быть открытым, закрытым и интервальным, в данном случае видны символы, поскольку на сайте указана кириллическая кодировка, которую не воспринимают стандартные веб-браузеры. Параметр `value` указывает на объект инвестиций, и значение в данном случае равно 1 – акции, как получить данное значение продемонстрировано на рисунке Рис. 4. Параметр `c_val[1]` уточняет список фондов, которые удовлетворяют значению `value`, поэтому в скобках у данного параметра указано значение 1. Значение `c_val[1]=8` определяет не отраслевые паевые инвестиционные фонды. Как определить данный параметр показано на рисунке Рис. 5. В случае, если будут

переданы значения `value=2` и `c_val[2]=23`, тогда Investfunds в результате покажет индексные паевые инвестиционные фонды, которые придерживаются стратегии следования Индексу МосБиржи. Параметр `prage` определяет номер страницы в результате сформированного списка фондов на сайте. Всего на одной странице демонстрируется 50 фондов.

```
links=names=c() # массив ссылок и названий
for (i in 0:1){ # 2 страницы

link=paste0("http://pif.investfunds.ru/funds/index.phtml?page=f
unds&type[]=%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9&c_
val[1]=8&value=1&npage=",i) # адрес страницы
  rem$navigate(link) # открытие страницы
  rem$getCurrentUrl() # текущий адрес
  elem <- rem$findElements(using = 'xpath',
"//tr/td[1]/a[@class='link']") # выбор элементов
  names <- append(names,unlist(lapply(elem,
function(x){x$getElementText()}))) # запись названий ПИФов
  links <- append(links,unlist(lapply(elem,
function(x){x$getElementAttribute("href")}))) # запись ссылок
ПИФов
}
```

Рис. 3. Получение списка паевых фондов с заданными параметрами

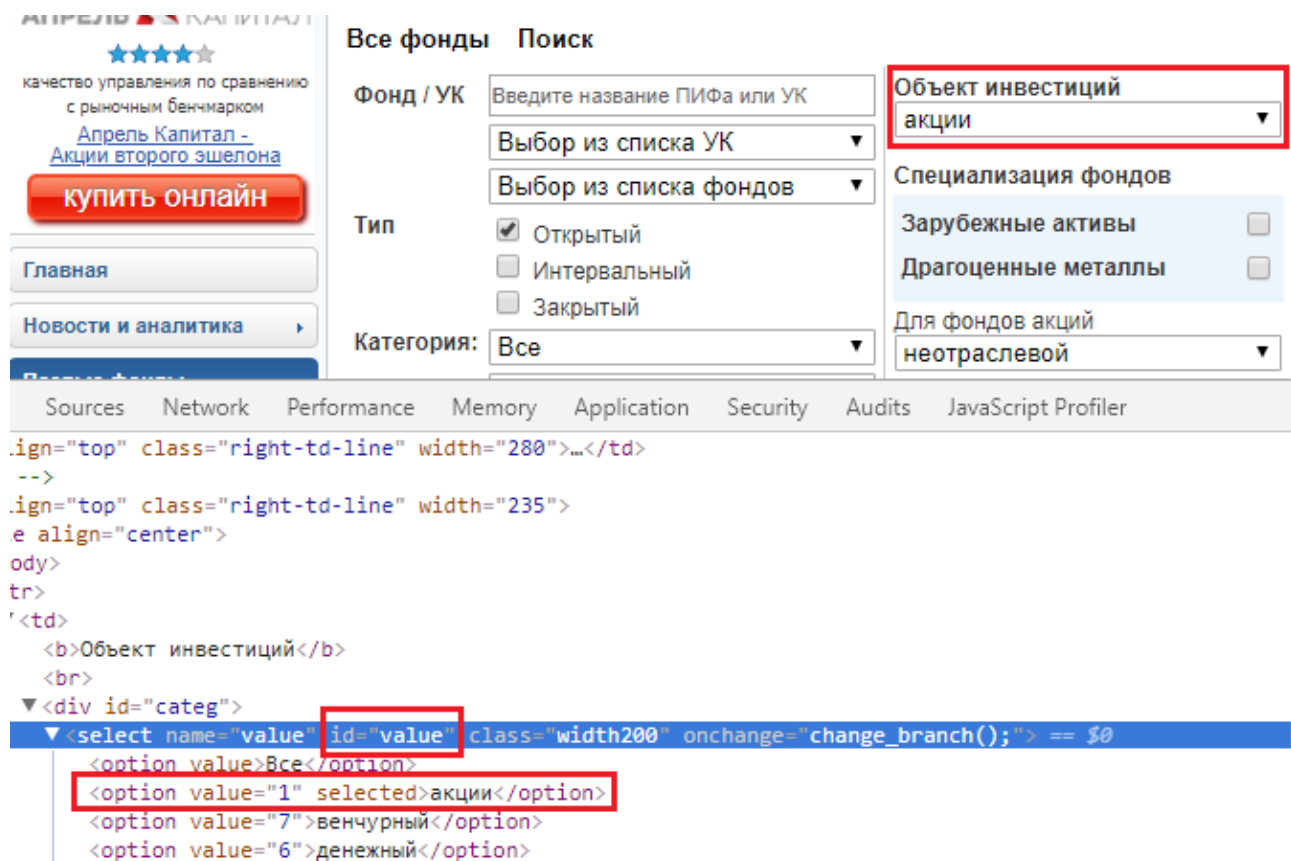


Рис. 4. Определение параметра объекта инвестиций фонда

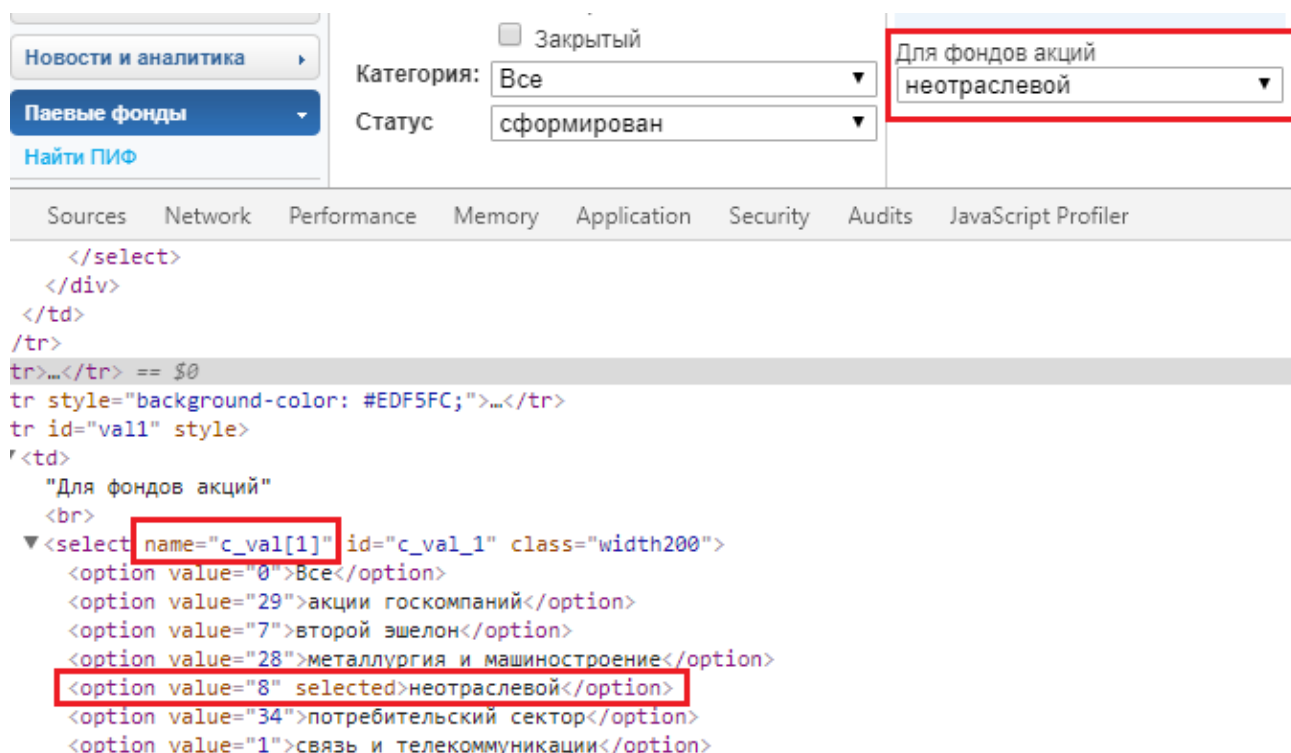


Рис. 5. Уточнение выбранного параметра фонда

В конечном итоге Investfunds находит 57 открытых не отраслевых паевых инвестиционных фондов акций и 13 индексных паевых инвестиционных фондов Индекса МосБиржи.

Таблица 1

Индексные паевые инвестиционные фонды

№	Наименование
1	АК БАРС Индексный
2	Аллтек - Индекс ММВБ
3	Альфа–Капитал Индекс ММВБ
4	БКС Индекс ММВБ
5	ВТБ - Индекс ММВБ
6	ДОХОДЪ. Российские акции. Первый эшелон
7	Ингосстрах Индекс МосБиржи
8	Метрополь Посейдон
9	Открытие – Индекс ММВБ
10	Райффайзен – Индекс ММВБ голубых фишек
11	РГС – Индекс ММВБ
12	Солид – Индекс ММВБ
13	СТОИК-Биржевые активы

Таблица 2

Неотраслевые паевые инвестиционные фонды

1	Агидель - акции	31	Меркури - Акции
2	АК БАРС - Акции	32	Метрополь Золотое Руно
3	Алёнка – Капитал	33	Мировые рынки
4	Альфа - Капитал Бренды	34	Мономах-Перспектива
5	Альфа-Капитал Ликвидные акции	35	Национальные акции плюс
6	Альфа–Капитал Ресурсы	36	Открытие - Иностранные акции
7	Апрель Капитал – Акции	37	Открытие – Акции
8	Апрель Капитал – Акции несырьевых компаний	38	Промсвязь Акции
9	Апрель Капитал – Акции сырьевых компаний	39	Райффайзен - Акции
10	Арсатера – фонд акций	40	Регион Фонд Акции
11	АТОН - ИНФРАСТРУКТУРА	41	РОНИН фонд акций
12	Атон - Петр Столыпин	42	РСХБ – Фонд Акции
13	АТОН - ФОНД 2025	43	РСХБ — Лучшие отрасли
14	Атон – Фонд международных	44	Сбербанк – Природные ресурсы

	рынков		
15	Базовые отрасли - акции	45	Сбербанк – Фонд активного управления
16	ВТБ – Фонд Акции	46	Сбербанк – Фонд акций Добрыня Никитич
17	ВТБ – Фонд Акции инфраструктурных компаний	47	Система Капитал - Мобильный. Акции. Валютный
18	ВТБ – Фонд Глобальных дивидендов	48	Система Капитал – Мобильный. Акции
19	ВТБ — БРИК	49	Солид - Глобус
20	Газпромбанк – Акции	50	Солид-Инвест
21	Глобал Капитал – Акции	51	СТОИК
22	Дмитрий Донской	52	СТОИК–Акции
23	ДОХОДЪ. Альтернативные инвестиции	53	ТКБ Инвестмент Партнерс – Премиум. Фонд акций
24	ИДжи Акции	54	ТКБ Инвестмент Партнерс – Фонд сбалансированный глобальный
25	ИнвестКапитал – фонд акций	55	ТФГ – Российские акции
26	КапиталЪ – Мировая индустрия спорта	56	УРАЛСИБ Акции роста
27	КапиталЪ – Перспективные вложения	57	УРАЛСИБ Первый
28	Консервативные ценные бумаги		
29	Лидер – Акции		
30	МДМ – мир акций		

По ссылкам на найденные паевые инвестиционные фонды будут собраны данные на каждый фонд. Общая информация по фондам, такая как наименование паевого инвестиционного фонда, ссылка на данный фонд на портале Investfunds, его внутренний номер на данном портале, а также размер вознаграждения менеджеру фонда и депозитарию, и прочие расходы, записываются в одну таблицу, а исторические данные по стоимости пая и стоимости чистых активов соответствующего фонда сохраняются в две отдельные таблицы формата временных данных. Две таблицы хранятся в отдельной структуре данных – list, затем все эти данные по каждому фонду собираются в одну переменную с типом list.

```

all=data.frame( # таблица с описанием и общей информацией
ПИФов
  Name=as.character(), # название ПИФа
  Link=as.character(), # ссылка на ПИФ
  Number=as.character(), # внутренний номер
  FeeMan=as.numeric(), # вознаграждение менеджера
  FeeDep=as.numeric(), # вознаграждение депозитария
  FeeOth=as.numeric(), # прочие расходы
  stringsAsFactors=FALSE
)
allD=list() # данные по цене и СЧА для ПИФов

```

Рис. 6. Описание таблицы с характеристиками фондов

Исторические данные о стоимости пая и величины стоимости чистых активов определенного фонда на портале Investfunds хранятся в файле формата Excel с расширением xls. Ниже представлен фрагмент программного кода, который в цикле проходя по каждому паевому инвестиционному фонду из выборки, полученной ранее, сохраняет информацию о фонде в таблицу. Файл Excel для каждого фонда скачивается путём отправки http запроса напрямую к серверу портала Investfunds, обходя графический интерфейс сайта. В результате в ответ возвращается файл, который сохраняется на жестком диске пользователя. Затем данный файл считывается в переменную, где далее из данной переменной информация по стоимости пая и чистых активов фонда сохраняется в две таблицы, включая временные данные.

```

for(i in 1:length(names)){ # цикл по каждому ПИФу
  all[i,"Name"]<-names[i] # название ПИФа
  all[i,"Link"]<-links[i] # ссылка на ПИФ
  all[i,"Number"]<-regmatches(links[i],      regexpr("[0-9]+",
links[i], perl=TRUE)) # внутренний номер ПИФа на сайте
  rem$navigate(links[i]) # открытие страницы ПИФа

  elem<-rem$findElement(using = 'xpath', "//a[@class='popup-
link arr-open-title']") # поиск кнопки на сайте для отображения
информации
  elem$clickElement() # нажатие кнопки
  elem      <-      rem$findElements(using      =      'xpath',
"//td[@style='border-top:0px;text-align:center']")      #      список
данных по вознаграждению
  z=unlist(lapply(elem,      function(x){x$getElementText()}))      #
текстовое представление данных по вознаграждению
  all[i,c("FeeMan","FeeDep","FeeOth")]=as.numeric(regmatches(z,
regexpr("[0-9.]+", z, perl=TRUE)))/100 # сохранения в числовом
виде

download.file(paste0("http://pif.investfunds.ru/funds/export_to
_excel.php?f2[0]=",all[i,"Number"],"&export=2&export_type=xls&s
tart_day&finish_day=08&finish_month=02&finish_year=2018&rnd=503
0"),destfile="pif.xls",mode="wb",quiet=TRUE) # скачивание файла
с данными по цене и СЧА ПИФа
x<-
read.xls(xls=paste0(dir,"/pif.xls"),perl="C:/Perl64/bin/perl.ex
e") # чтение файла в переменную - таблицу
x<-x[which(x[,1]!="&x[,1]!="00.00.0000"),] # очистка пустых
строк и неправильных дат
x<-x[2:length(x[,1]),1:3] # очистка от названий
x[,2]=as.numeric(gsub("      ", "",as.character(x[,2])))      #
преобразование цены из текстового в числовой формат
x[,3]=as.numeric(gsub("      ", "",as.character(x[,3])))      #
преобразование СЧА из текстового в числовой формат
x1<-as.xts(x[,2],order.by=as.Date(x[,1],      "%d.%m.%Y"))      #
преобразование цены в xts формат
x2<-as.xts(x[,3],order.by=as.Date(x[,1],      "%d.%m.%Y"))      #
преобразование СЧА в xts формат
allD[[i]]=list(x1,x2) # сохранение в лист двух xts фреймов
print(rem$getCurrentUrl()) # печать текущего адреса страницы
}
names(allD)=all[, "Number"] # название в листах соответствуйте
внутреннему номеру ПИФов

```

Рис. 7. Загрузка данных по каждому паевому фонду

Сам `http` запрос начинается с ссылки на портал `"http://pif.investfunds.ru/funds/export_to_excel.php"`. Далее задаются параметры

для экспорта, например, в `f2[0]` указывается внутренний номер паевого инвестиционного фонда на портале, который был сохранён ранее, `export_type=xls` означает формат файла с расширением `xls`. Другие параметры `finish_day`, `finish_month`, `finish_year` указывают на дату, по которую необходимо включить данные в выгрузку, соответственно день, месяц и год. Если не передавать значения `start_day`, `start_month` и `start_year`, которые информируют о начальной дате, то в файле временной ряд данных будет начинаться с начала создания запрашиваемого фонда. После отправки запроса на сервер, в ответ приходит файл с данными, который сохраняется в данном случае под именем `"rif.xls"`. Для импорта данного файла в переменную используется функция `read.xls`, которой требуется интерпретатор Perl установленный на компьютере пользователя [12].

Две таблицы для каждого фонда группируются в формат данных `list`, и затем как один объект добавляется к другим фондам, данные которых так же были сгруппированы. В результате получается одна переменная формата `list`, в которой находится множество элементов такого же формата `list`, и в одном элементе находятся две таблицы с данными, которые соответствуют одному паевому инвестиционному фонду. Названия элементов соответствуют внутреннему номеру фонда на портале Investfunds, чтобы в дальнейшем можно было легко идентифицировать фонд.

Поскольку в качестве одного из бенчмарков будет выступать портфель акций, соответствующий Индексу ММВБ 10, то необходимо собрать данные о исторической стоимости акций на московской фондовой бирже. Одним из источников информации о цене акций за определенный момент времени является сервис по экспорту архивных данных от компании ПАО «РБК», расположенный по URL адресу: <http://export.rbc.ru/expdocs/free.micex.0.shtml>. Данный сервис позволяет получить данные в формате `csv`, где поля разделяются запятой (можно указать при экспорте). Формат записи данных в файле приведён на рисунке Рис. 9, где указывается сокращенное наименование акций эмитента на фондовой бирже, дата, цена закрытия, самая низкая и

высокая цена для данной акции в течение торгового дня, цена закрытия, а также объём торгов, в размере количества акций, и среднее значение цены акции.

Рис. 8. Сервис экспорта данных о стоимости акций от ПАО «РБК»

```
TICKER, DATE, OPEN, HIGH, LOW, CLOSE, VOL, WAPRICE
ABRD, 2017-01-03, 93.5, 99.5, 93, 99.5, 400, 96.5
AFKS, 2017-01-03, 23.215, 23.84, 22.945, 22.945, 5035400, 23.34
AFLT, 2017-01-03, 153.98, 157.95, 151.33, 151.68, 3548200, 154.21
...
```

Рис. 9. Структура формата данных о стоимости акций

Поскольку данные содержат в себе информацию о всех торгуемых акциях за выбранный период, поэтому они имеют большой объём, и файл выгрузки с данными формируется сервером продолжительное количество времени. А также, данные достаточно получить лишь один раз для дальнейшего использования. Вследствие чего, целесообразно экспортировать данные вручную, разбивая данные по периодам в один календарный год. В случае с автоматическим получением данных по паевым инвестиционным фондам, задавая разные параметры можно выбрать разные типы фондов и разную специализацию фонда, а также количество фондов может быть достаточно

велико. Как следствие ручной сбор данных по паевым инвестиционным фондам может быть затруднительным процессом.

Для более точного воспроизведения портфеля на основе индекса нужно иметь информацию о базе расчёта данного индекса в определенный момент времени. База расчёта представляет собой список акций, на основе которых рассчитывается индекс, а также размер доли каждой акции в индексе. Московская Биржа предоставляет архивную базу расчета Индекса ММВБ 10 по URL адресу: <https://www.moex.com/ru/index/MICEX10INDEX/constituents/>. Для упрощения воссоздания индекса, изменение базы расчета было преобразовано в формат, который представлен на рисунке Рис. 10. Это формат файла таблицы csv, где в первом поле указывается дата начала новой базы расчета, далее конечная дата, в следующих десяти полях следует сокращенное название десяти акций, которые учитываются в индексе в определенный ранее период времени.

```
...  
22.09.2017, 21.12.2017, SBER, GAZP, GMKN, LKOH, MOEX, ALRS, ROSN, VTBR, M  
GNT, AFLT  
22.12.2017, 15.03.2018, SBER, GAZP, MGNT, GMKN, MOEX, LKOH, VTBR, ROSN, A  
LRS, AFLT  
16.03.2018, 21.06.2018, SBER, GAZP, MGNT, GMKN, LKOH, VTBR, MOEX, ROSN, A  
LRS, AFLT  
...
```

Рис. 10. Формат данных архивной базы расчёта Индекса ММВБ 10

В качестве безрисковой доходности были получены данные по кривой бескупонной доходности однолетних облигация федерального займа с информационно портала Investing.com (<https://www.investing.com>) по следующей ссылке: <https://www.investing.com/rates-bonds/russia-1-year-bond-yield-historical-data>. Файл с данными имеет формат таблицы с расширением csv, где представлены поля Date – дата в английском формате, Price – цена закрытия, Open – цена открытия, High, Low – самая высокая и низкая цена в течение торгового дня соответственно, Change % - изменение значения цены по отношению к предыдущему торговому дню. Аналогичным образом были

получены данные по Индексу МосБиржи
 (<https://www.investing.com/indices/mcx>) и Индексу ММВБ 10
 (<https://www.investing.com/indices/mcx10>).

```
"Date", "Price", "Open", "High", "Low", "Change %"
"Apr 09, 2018", "6.540", "6.410", "6.650", "6.070", "4.47"
"Apr 06, 2018", "6.260", "6.170", "6.270", "6.080", "1.62"
"Apr 05, 2018", "6.160", "6.190", "6.190", "6.070", "0.82"
...
```

Рис. 11. Формат данных файла с портала Investing.com

Также информация о значениях кривой бескупонной доходности государственных облигаций доступна на сайте Центрального Банка Российской Федерации по URL адресу: http://www.cbr.ru/hd_base/zcyc_params/.

Далее необходимо получить данные о торгах по фьючерсу на Индекс МосБиржи. Такую информацию предоставляет Московская Биржа по URL адресу: <https://www.moex.com/ru/forts/contractbaseresults.aspx>. Биржа предоставляет данные в следующем формате: дата в европейском формате, код фьючерса с ближайшей датой исполнения, средневзвешенная цена за лот за один торговый день, расчётная цена, цена первой сделки в указанную дату, максимальная и минимальная цена фьючерса в течение торгового дня, а также значение цены последней сделки и изменение стоимости фьючерса по отношению к предыдущему торговому дню, выраженное в процентах. Помимо этого, биржа так же предоставляет информацию о торгуемых объёмах фьючерса в течение торгового дня, а именно: объём последней сделки и число сделок, выраженное в количестве контрактов, данные по объёму торгов и объёму открытых позиций представлены как в денежном выражении в валюте «рубли», так и по количеству контрактов. Особо стоит отметить данные о размере гарантийного обеспечения, поскольку при покупке фьючерса покупатель резервирует только сумму в размере гарантийного обеспечения. И в дальнейшем в зависимости от стоимости фьючерса и от волатильности базового актива будет изменяться величина гарантийного обеспечения, и как

следствие, данная информация важна при составлении портфеля, в основе которого фьючерс на Индекс МосБиржи.

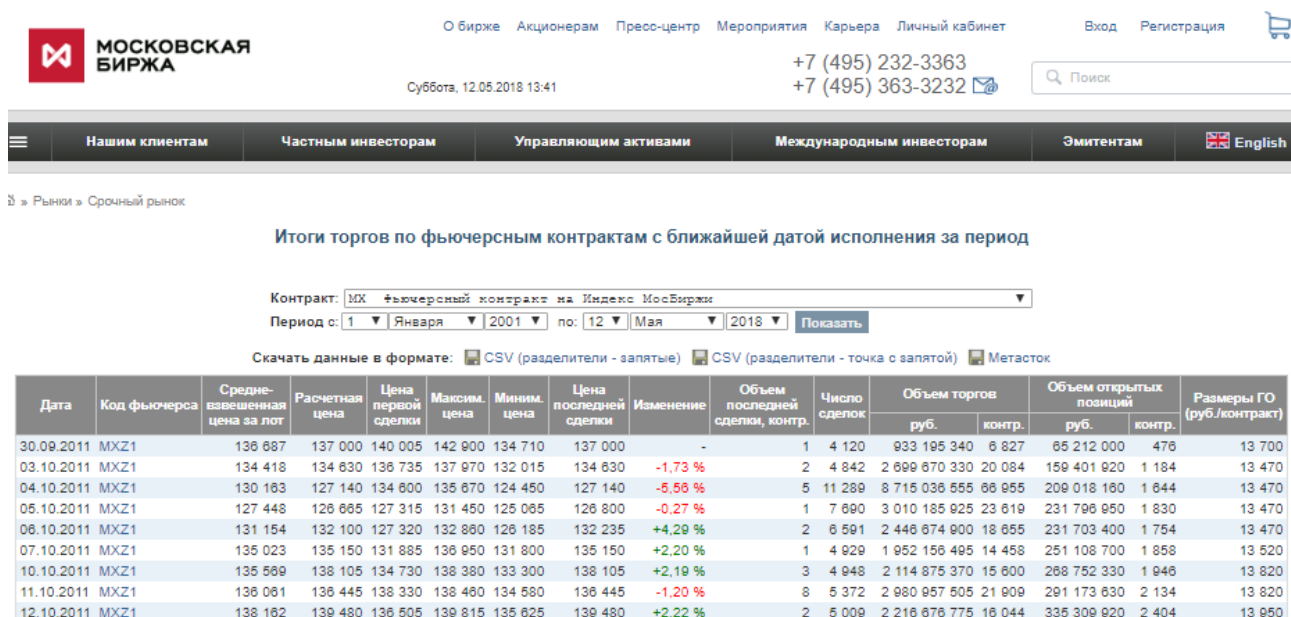


Рис. 12. Итоги торгов по фьючерсному контракту на Индекс МосБиржи

Также стоит упомянуть, что стоимость фьючерса рассчитывается как сто базовых активов, где базовый актив – это «Индекс МосБиржи». Следуя спецификации на данный инструмент, расчётная цена для фьючерса вычисляется, как среднее значение Индекса МосБиржи за период с 15:00 до 16:00 за текущую дату, умноженное на 100.

1.2. Подготовка данных

В начале, для создания портфеля, в основе которого лежат паи индексных паевых инвестиционных фондов, будут объединены данные по стоимости пая. В переменную `pif.price` будут объединены данные по стоимости соответствующих паев, а в переменную `pif.return` доходность данных паевых инвестиционных фондов за месяц. В цикле по всем индексным фондам в переменной, в которую ранее были сгруппированы все данные по исторической стоимости пая и стоимости чистых активов фондов, происходит получение данных по одному фонду. Далее функция `to.monthly` приводит дневные цены по стоимости пая к данным за месяц, где указывается конечная дата месяца, цена в начале и в конце данного месяца, также самая высокая и низкая цена в течение

месяца. Параметр `indexAt = 'lastof'` указывает на то, что конечная дата месяца будет указана как календарная дата, а не последняя дата, за которую есть данные по стоимости пая за данный месяц. Функция `Cl` из результата предыдущей функции выбирает дату и цену закрытия. Функция `monthlyReturn` вычисляет на основе результата предыдущей функции доходность за данный месяц. Все результаты сохраняются в таблицу с форматом `xts`, где используется временной формат данных, другими словами данные привязаны к соответствующим им датам [15]. Наименование столбцов в таблице соответствует номеру паевого инвестиционного фонда на портале `Investfunds` для дальнейшей идентификации фонда.

```
pif.price=xts()
pif.return=xts()
for(i in 1:length(allD)){
  if(i==1){
    # заполнение первой колонки таблицы
    pif.price=Cl(to.monthly(allD[[i]][[1]],indexAt = 'lastof'))
    pif.return=monthlyReturn(to.monthly(allD[[i]][[1]],indexAt
= 'lastof'))
  } else {
    # добавление следующих колонок в таблицу

    pif.price=cbind(pif.price,Cl(to.monthly(allD[[i]][[1]],indexAt
= 'lastof'))))

    pif.return=cbind(pif.return,monthlyReturn(to.monthly(allD[[i]][
[1]],indexAt = 'lastof'))))
  }
}
colnames(pif.price)=names(allD)
colnames(pif.return)=names(allD)
```

Рис. 13. Группировка данных о стоимости и доходности паевых фондов

Излишние данные, которые не влияют на процесс составления портфеля, например, в самом начале у некоторых фондов могут быть данные только за год, стираются.

```

for(i in 1:ncol(pif.return)){
  pif.return[1:tail(which(is.na(pif.return[,i])),n=1),i]=NA
  pif.price[1:tail(which(is.na(pif.price[,i])),n=1),i]=NA
}

```

Рис. 14. Стирание лишних значений из таблицы

Прежде чем приступить к преобразованию данных по кривой бескупонной доходности однолетних облигаций, следует указать среде выполнения R, что формат даты будет в английском формате. Далее данные из ранее полученного файла в формате csv считываются в переменную. Затем данные из переменной приводятся к временному формату данных, при этом следует указать текущий формат даты, например, “Apr 03, 2018”, через параметр `format = "%b %d, %Y"`, где %b – сокращенное название месяца, %d – день месяца в формате двузначного номера, %Y – номер года в четырехзначном формате.

```

# использовать английский формат даты
Sys.setlocale("LC_TIME", "English")
# чтение данных из файла
ru.ly=read.csv("~/RU1Y.csv",stringsAsFactors = FALSE)
# преобразование в формат данных xts
ru=xts(as.numeric(ru.ly[(1:(nrow(ru.ly)-2)),2]),order.by =
as.Date(ru.ly[(1:(nrow(ru.ly)-2)),1], format = "%b %d, %Y"))
# перевод доходности за год к доходности за месяц
ru.ly=xts((1 + C1(to.monthly(ru,indexAt =
'lastof'))/100)^(1/12)-1)

```

Рис. 15. Получение данных о доходности по безрисковой ставке

Поскольку данные по кривой бескупонной доходности облигаций представлены в годовом формате, то их нужно перевести в формат месячной доходности по следующей формуле:

$$r_m = (1 + r_Y)^{\frac{1}{12}} - 1, \quad (1)$$

где: r_m - значение доходности за месяц;

r_Y - значение доходности за год.

Аналогичным образом преобразовываются данные по Индексу МосБиржи и Индексу ММВБ 10. Формат даты в данном случае аналогичный, поскольку данные были получены из одного источника. Информация находится

в формате значения стоимости индекса за один день. Далее значения преобразуются к месячному формату с использованием функции `to.monthly`. После чего результат данной функции обрабатывается функцией `monthlyReturn`, которая возвращает месячную доходность соответствующего индекса. Переменные соответствующих индексов так же, как и для паевых инвестиционных фондов, имеют временной формат данных.

```
# чтение данных из файла
moex=read.csv("~/MOEX.csv",stringsAsFactors = FALSE)
# преобразование в формат данных xts
moex=xts(as.numeric(gsub(",","", moex[,2])),order.by =
as.Date(moex[,1], format = "%b %d, %Y"))
# вычисление доходности за месяц
moex=monthlyReturn(to.monthly(moex, indexAt = 'lastof'))

moex10=read.csv("~/MOEX10.csv",stringsAsFactors = FALSE)
moex10=xts(as.numeric(gsub(",","", moex10[,2])),order.by =
as.Date(moex10[,1], format = "%b %d, %Y"))
moex10=monthlyReturn(to.monthly(moex10, indexAt = 'lastof'))
```

Рис. 16. Получение данных о доходности по Индексу МосБиржи и Индексу ММВБ 10

Как было упомянуто выше, исторические данные по акциям были получены файлами, каждый из которых содержит данные за один год. В начале необходимо получить наименование файлов с данными из папки, куда они были сохранены. Затем в цикле каждый файл считывается в переменную, и из данной переменной вся информация добавляется в единую таблицу. Таким образом все данные по исторической стоимости акций на Московской Бирже были объединены в одну таблицу.

```
# получение наименования файлов
files=list.files("~/dat",full.names=TRUE)
# общая таблица с данными
data.equity=data.frame()
for(file in files){
  # чтение одного файла
  eq=read.csv(file,stringsAsFactors=FALSE)
  # добавление данных в одну таблицу
  data.equity=rbind(data.equity,eq)
}
```

Рис. 17. Получение списка файлов с данными о исторической стоимости акций

Поскольку данные по акциям необходимы для воспроизведения Индекса ММВБ 10, то далее в переменную записывается информация об изменениях базы расчёта данного индекса. До середины 2008 года в базе расчёта индекса встречается ISIN код акции, вместо распространённого в настоящее время краткого наименования биржевого инструмента – тикер. Поэтому была создана таблица перекодировки из ISIN кода в актуальный формат наименования или тикер, которые соответствуют одному эмитенту.

Таблица 3

Соответствие ISIN кода и тикера

№	ISIN код	Тикер	№	ISIN код	Тикер
1	RU0008926258	SNGS	10	RU0009029524	SNGSP
2	RU0008926621	EESR	11	RU0009029540	SBER
3	RU0008943394	RTKM	12	RU14MSNG3008	MSNG
4	RU0008958863	MSNG	13	RU0009046668	LKOH
5	RU0008960828	IRGZ	14	RU0009029557	SBERP
6	RU0009011126	MFGS	15	RU0009054449	YUKO
7	RU0009024277	LKOH	16	RU14GMKN0507	GMKN
8	RU0009029532	EESRP	17	RU14TATN3006	TATN
9	RU0009033591	TATN	18	RU14SIBN1003	SIBN

Данные о переходе от ISIN кода к тикеру считывается в переменную. Затем в исходной базе расчёта индекса происходит замена этих двух компонент.

```
# информация о базе расчета Индекс ММВБ 10
mic.10<-read.csv(paste0(dir,"/10micexdata.csv"),header =
FALSE,stringsAsFactors = FALSE)
# изменения наименования сокращенных названий акций
mic.10.replace=read.csv(paste0(dir,"/10micex.csv"),header =
FALSE,stringsAsFactors = FALSE)
# замена наименования сокращенных названий акций в базе расчета
for(i in 1:nrow(mic.10.replace)){
  mic.10[mic.10==mic.10.replace[i,1]]=mic.10.replace[i,2]
}
```

Рис. 18. Чтение и замена ISIN кода на тикер

Весь объём данных по акциям необходимо структурировать для их дальнейшей обработки. Для структурирования в первую очередь нужно выделить весь массив дат, в которые происходили торги по представленным акциям. В цикле по каждому периоду, в который изменялась база расчёта

индекса строится массив календарных дат. Далее по каждому элементу, входящему в состав индекса, из всего объёма данных выбирается его торговая история. Строится пересечение календарных и торговых дат, после чего данный диапазон добавляется к общему массиву дат за данный период. В результате должен получиться массив дат с уникальными значениями за определенный период, который затем будет добавлен в общий массив дат для всего объёма данных по акциям. Таким образом заранее будет известно количество строк в структурированной таблице, которая будет иметь также временной формат данных xts.

```
date.all=c() # массив со всеми датами
for(i in 18:nrow(mic.10)){
  # база расчета Индекса ММВБ 10 за определенный период
  tickers=unlist(mic.10[i,3:length(mic.10[i,])])
  # массив календарных дат для базы расчета
  dates=seq.Date(as.Date(mic.10[i,1],"d.%m.%Y"),
as.Date(mic.10[i,2],"d.%m.%Y"), by = "day")
  date.t=c() # массив дат для текущей базы расчета
  for(t in tickers){
    # данные по одной акции из базы расчёта
    y=data.equity[which(data.equity[,1]==t),]
    # пересечение календарных дат и торговых дней
    date.inter=intersect(dates,as.Date(y[,2]))
    # добавление пересечения дат в массив
    date.t=append(date.t,date.inter)
  }
  # добавление уникальных дат в общий массив
  date.all=append(date.all,unique(date.t))
}
```

Рис. 19. Получение массива всех торговых дат по акциям

Массив всех акций, которые когда-либо входили в Индекс ММВБ 10 представлен в виде сокращенных названий соответствующих инструментов. Данный массив используется в качестве названия столбцов в структурированной таблице.

```
mic.ticker=c() # массив сокращенных названий акций
mic.ticker=unique(unlist(sapply(c(18:nrow(mic.10)),function(i){
mic.ticker=append(mic.ticker,mic.10[i,3:12])})))
mic.ticker=mic.ticker[mic.ticker!=""]
```

Рис. 20. Массив сокращенных названий акций из Индекса ММВБ 10

Далее происходит создание основной структурированной таблицы и её заполнение. Таблица заполняется пустыми значениями, количество строк соответствует массиву дат, полученному ранее, количество столбцов соответствует массиву всех акций, которые когда-либо входили в индекс, так же полученному ранее. По каждой акции из полученного массива всех акций, которые когда-либо входили в индекс, выбирается из общего массива данных его торговая история. И цены закрытия записываются в структурированную таблицу по соответствующему совпадению дат и сокращенному названию акции. Таким образом заполняется вся таблица.

```
# структурированная таблица с данными
data.10=xts(matrix(NA, nrow = length(date.all), ncol =
length(mic.ticker), dimnames
list(date.all,mic.ticker)),order.by = as.Date(date.all))
for(t in colnames(data.10)){
  # данные по одной акции из базы расчёта
  y=data.equity[which(data.equity[,1]==t),]
  # цены закрытия по данной акции
  y=xts(Cl(y),order.by=as.Date(y[,2]))
  # заполнение данных в таблице
  data.10[index(y),t]=y[index(data.10)]
}
```

Рис. 21. Таблицы с данными об исторической стоимости акций

При подготовке данных по фьючерсам в начале происходит чтение данных из файла, полученного с сайта Московской Биржи. Далее составляется массив наименований всех фьючерсов, которые торговались с ближайшей датой исполнения. После чего создаётся общая таблица, где строки соответствуют торговым дням, а столбцы – фьючерсам, которые торговались.

```

# чтение данных из файла
future=read.csv("~/MX.csv", stringsAsFactors = F)
# наименование фьючерсов
future.name=unique(future[,2])
# таблица со значениями ГО по фьючерсам
future.data=as.data.frame(matrix(NA,nrow = nrow(future), ncol =
length(future.name)))
# название строк
row.names(future.data)=future[,1]
# название столбцов
colnames(future.data)=future.name
# преобразование во временной формат данных
future.data=as.xts(future.data, order.by =
as.Date(row.names(future.data), "%d.%m.%Y"))
# создание новых таблиц копированием
future.price=future.price.estim=future.action=future.data

```

Рис. 22. Подготовка таблиц для заполнения данными по фьючерсам

Путём копирования также создаются дополнительные таблицы. По каждому фьючерсу заполняются данные в таблицах. Информация по фьючерсу выбирается по его сокращенному наименованию, затем формат даты преобразуется из европейской в стандартный формат для R. Данные заполняются по совпадению дат и наименованию фьючерса. В таблицу `future.data` заносится информация о размере гарантийного обеспечения, в `future.action` – информация о том, торгуется ли фьючерс на определенную дату, в `future.price` – данные о значении средней взвешенной цены фьючерса, в `future.price.estim` – значения расчётной цены по фьючерсу. Также в конце временные переменные удаляются.

```

# заполнение данных по каждому фьючерсу
for(name in future.name){
  # временная переменная
  x=future[future[,2]==name,]
  # преобразование формата даты
  date=as.Date(x[,1], "%d.%m.%Y")
  # заполнение значений ГО
  future.data[date,name]=x[,16]
  # заполнение актуальности фьючерса
  future.action[date,name]=1
  # заполнение значений средней взвешенной цены
  future.price[date,name]=x[,3]
  # заполнение значений цены расчета
  future.price.estim[date,name]=x[,4]
  # удаление временных переменных
  rm(x,date)
}

```

Рис. 23. Заполнение таблиц данными по каждому фьючерсу

В данной главе описан процесс получения данных и их подготовка для дальнейшей их обработки, и получения конечного результата. Были собраны и подготовлены данные по открытым паевым инвестиционным фондам акций, индексным паевым инвестиционным фондам, Индексу МосБиржи, по бескупонной доходности однолетних государственных облигаций Российской Федерации, данные по Индексу ММВБ 10, исторические данные по стоимости акций на Московской Бирже, архивные данные по базе расчета Индекса ММВБ 10, а также данные по фьючерсам на Индекс ММВБ с ближайшей датой исполнения.

2. Моделирование

2.1. Построение бенчмарка на основе индексных фондов

Главным критерием создание бенчмарка на основе индексных паевых инвестиционных фондов должно стать юлизость значений при учете реальных издержек фондом самого Индекса МосБиржи. Чтобы создать портфель из индексных фондов, который бы максимально повторял Индекс МосБиржи, был разработан ниже предложенный алгоритм.

Ранее была создана общая таблица по доходности для всех индексных фондов. На основе этой таблицы будет создана другая, в которой будут помещены оценки для фондов. Данные оценки характеризуют то, насколько фонд максимально близко повторяет Индекс МосБиржи. Вначале задаётся период, на котором будет сравниваться фонд и сам индекс, в данном случае это 24 месяца. Данный период в цикле передвигается по всему временному отрезку данных, другими словами, получается плавающее окно.

В течение одного прохода цикла в промежуточную переменную `dta.m` записываются данные по индексным фондам длиной 24 месяца из общей таблицы. Создаётся пустой массив для сохранения оценок фондов. Далее начинается второй цикл по столбцам `dta.m`, поскольку каждый столбец данных соответствует фонду. Происходит проверка, что количество данных по фонду равно длине периода. Затем вычисляется пересечение дат данных и дат Индекса МосБиржи, для получения соответствующих данных последнего. В `diff.index` записывается отклонение фонда от индекса, которое вычисляется как разность доходности индексного фонда и бенчмарка по модулю. Чтобы сравнить насколько один фонд точнее следует индексу, чем другой, для каждого фонда за данный период в 24 месяца на основе его отклонения от бенчмарка формируется оценка как среднее значение отклонения, умноженное на среднеквадратичное отклонение разности фонда и бенчмарка. Такие оценки формируются для всех фондов в таблице.

После того, как оценки сформированы, на их основе в таблицу pif.act будут записаны веса фондов. Но сначала происходит проверка – есть ли не пустые оценки. Затем массиву оценок присваивается номер индексного фонда, чтобы их идентифицировать далее. Также из массива оценок удаляются те фонды, по которым было недостаточно данных для расчётов.

```
# период оценки
period = 24
# таблица оценки
pif.act=pif.return
# очистка значений
pif.act[]=NA
# движущийся период оценки в цикле
for(i in 1:(nrow(pif.return)-(period-1))){
  # выделение данных за период в переменную
  dta.m=pif.return[i:(i+(period-1)),]
  # массив с весами оценок фондов
  diff.best=c()
  for(j in 1:ncol(dta.m)){
    # данные по каждому фонду записаны в столбец
    pif=dta.m[,j]
    diff.best[j]=NA
    # проверка на то, что количество данных по фонду
    # соответствует длине периода
    if(length(na.trim(pif))==period){
      # вычисление пересечения дат с Индексом МосБиржи
      time=intersect(index(pif),index(моех))
      # вычисление разности между фондом и индексом
      diff.index=abs(Return.excess(pif,моех[as.Date(time)]))
      # вычисление показателя для сравнения с другими фондами
      diff.best[j]=mean(diff.index)*sd(diff.index)
    }
  }
  # проверка, если все данные пустые
  if(length(diff.best[is.na(diff.best)])==length(diff.best))
next
  # заполнение названия фондов к их оценке
  names(diff.best)=colnames(dta.m)
  # остаются только те фонды, по которым сформирована оценка
  diff.best=diff.best[!is.na(diff.best)]

  # запись в общую таблицу оценок фондов
  pif.act[tail(index(pif),n=1),names(sort(diff.best))]=c(1:length
(diff.best))
}
```

Рис. 24. Оценка фондов на их следование Индексу МосБиржи на движущемся периоде

В конце массив сформированных оценок для фондов сортируется от меньшего к большему и в конечную таблицу записываются веса для соответствующих фондов. Веса соответствуют порядковому номеру в массиве оценок.

	334	708	746	316	268	781	1195	937	936	1023	1259	389	751
2008-01-31	5	6	2	4	3	8	NA	NA	NA	NA	NA	7	1
2008-02-29	6	7	4	2	3	8	NA	NA	NA	NA	NA	5	1
2008-03-31	4	8	6	3	2	7	NA	NA	NA	NA	NA	5	1
2008-04-30	3	8	5	1	4	6	NA	NA	NA	NA	NA	7	2
2008-05-31	1	7	5	4	2	6	NA	NA	NA	NA	NA	8	3
2008-06-30	2	8	5	3	1	6	NA	NA	NA	NA	NA	7	4
2008-07-31	2	7	5	4	1	6	NA	9	10	NA	NA	8	3
2008-08-31	1	7	5	3	2	9	NA	8	6	NA	NA	10	4
2008-09-30	8	6	3	1	5	7	NA	9	4	10	NA	11	2
2008-10-31	4	5	7	1	3	11	NA	6	9	8	NA	10	2
2008-11-30	2	11	6	5	1	10	NA	3	8	7	NA	9	4
2008-12-31	2	10	5	4	1	9	NA	3	11	6	NA	8	7
2009-01-31	1	11	6	4	3	9	NA	2	10	5	NA	8	7
2009-02-28	1	12	11	5	2	9	NA	3	10	6	4	8	7
2009-03-31	1	13	12	5	2	10	8	4	11	6	3	9	7
2009-04-30	1	12	13	5	2	10	7	4	11	6	3	9	8
2009-05-31	3	13	12	5	2	10	7	4	11	6	1	9	8
2009-06-30	2	13	12	6	3	11	8	4	10	5	1	9	7
2009-07-31	2	12	13	6	4	11	8	3	10	5	1	9	7

Рис. 25. Пример сформированной таблицы оценок для индексных фондов

Для выбора индексного паевого инвестиционного фонда, который будет входить в портфель в определенный момент времени, создан алгоритм, приведённый ниже. Будет выбираться один фонд, который по наименьшей сумме весов за период в 12 месяцев максимально точно следует основному Индексу МосБиржи. Данный фонд будет объектом инвестиций на следующий период в 12 месяцев, по истечению которого снова будет выбираться фонд, который наиболее близко соответствовал индексу на протяжении этого периода в 12 месяцев.

Вначале создаётся копия таблицы, приведенной выше, на её основе будет создана таблица инвестиций. Далее задаётся период для инвестиций, и индикатор. Если данный индикатор делится на период без остатка, значит наступил момент вычисления фонда для инвестиций на следующий период.

В цикле по таблице выше во временную переменную выбираются данные по длине заданного периода. Далее идёт проверка, что данных достаточно для анализа. Если данных достаточно и наступило условие для выбора фонда инвестиций, то алгоритм продолжается дальше. Сначала суммируются по столбцам значения во временной переменной, каждый столбец соответствует одному индексному фонду. Затем значения суммы сортируются в порядке возрастания, а в массив записываются номера индексных фондов в соответствии с сортировкой. Далее вычисляется период инвестиции – начало и конец периода. Выбирается первое значение из массива, которое соответствует номеру фонда, показатели которого наиболее точно повторяли Индекс МосБиржи.

```

# создание копии таблицы
pif.act.copy=pif.act
# период для инвестиций
period=12
# стирание данных
pif.act[]=NA
# индикатор для начала
# вычисления фонда в который
# нужно инвестировать
j=11
for(i in 1:(nrow(pif.act.copy)-(period-1))){
  # выделение данных за период в переменную
  dta.m=pif.act.copy[i:(i+(period-1)),]
  # проверка, достаточно ли данных в периоде
  if(sum(dta.m[,na.rm = TRUE])>0){
    # увеличение периода на 1 месяц
    j=j+1
    # если количество месяцев кратно периоду
    # то нужно вычислять новый фонд для инвестиций
    # на следующий период
    if(j%%period==0){
      # вычисляется сумма по каждому фонду
      # затем данные записываются в массив
      best=names(sort(colSums(dta.m)))[1]
      # вычисление месяца начала инвестиций
      k=i+period
      # вычисление конечного месяца инвестиции
      k1=ifelse((k+(period-
1))>nrow(pif.act.copy),nrow(pif.act.copy),(k+(period-1)))
      # запись в таблицу информации о том,
      # в какой фонд будут сделаны инвестиции
      pif.act[index(pif.act.copy[k:k1,]),best[1]]=1
    }
  }
}
}

```

Рис. 26. Выбор индексных фондов для инвестиций

Затем в общей таблице за выбранный период и в столбце с номером фонда ставится индикатор, который в дальнейшем символизирует о том, что нужно инвестировать в данный фонд. В результате на основе данной таблицы будет строиться портфель из индексных паевых инвестиционных фондов, который будет выступать в качестве одного из бенчмарков.

	334	708	746	316	268	781	1195	937	936	1023	1259	389	751
2008-01-31	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-02-29	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-03-31	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-04-30	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-05-31	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-06-30	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-07-31	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-08-31	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-09-30	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-10-31	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-11-30	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2008-12-31	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2009-01-31	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2009-02-28	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2009-03-31	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
2009-04-30	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA
2009-05-31	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA
2009-06-30	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA
2009-07-31	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA

Рис. 27. Пример таблицы с индикаторами для инвестирования в индексные фонды

Построение портфеля на основе цен и индикаторов активов, которые должны входить в портфель, происходит в отдельной функции, которая будет описана в следующем параграфе.

2.2. Построение инвестиционного портфеля функцией в R

Данная функция вычисляет стоимость портфеля на основе данных о стоимости активов и индикаторов, которые указывают на то, какие именно активы входят в портфель в определённый момент времени. Функция принимает на вход следующие значения:

- а) таблица с данными о ценах фондов;
- б) таблица с индикаторами, которые указывают на то, какие активы в определённый момент входят в портфель;
- в) значение изначального капитала для инвестиций;
- г) массив с информацией о комиссиях, с двумя значениями: первое значение – комиссия на покупку, второе – комиссия на продажу;
- д) путь в файловом менеджере к файлу, где будут вестись записи о совершаемых операциях, при ведении портфеля.

Начало функции представлено на рисунке Рис. 28. Индикатор `diag` включает или отключает ведение записей – логов об операциях с активами в портфеле. Если передано значение пути к файлу, тогда записи будут вестись, по умолчанию значение пустое. Также в разные переменные записываются значения о комиссиях на продажу и покупку соответственно. Если логи будут вестись, то в файл в самое начало записывается текущие дата и время. Сам портфель представляет собой массив активов, и данный массив постоянно изменяется. Состояние портфеля – его стоимость записывается в таблицу `portfel.history`, которая имеет временной формат данных, другими словами, определенная запись в таблице соответствует стоимости портфеля на определенную дату.

```
portfel.equity=function(prices, action, bank, fee, log = ""){  
  # индикатор, нужно ли вести логи  
  diag = FALSE  
  if(log!="") diag=TRUE  
  # значение комиссии на покупку  
  fee.buy=fee[1]  
  # значение комиссии на продажу  
  fee.sell=fee[2]  
  # запись текущей даты в логи  
  if(diag) write(as.character(Sys.time()),log,,FALSE)  
  # портфель, как массив активов  
  portfel=c()  
  # история изменения стоимости портфеля  
  portfel.history=action[,1]  
  portfel.history[]=NA  
  ...  
}
```

Рис. 28. Начало функции построения инвестиционного портфеля

На рисунке Рис. 29 представлено продолжение функции, где начинается цикл, который является главным образующим звеном функции. Цикл проходит по строкам таблицы, в которой указано состояние портфеля (см. рисунок Рис. 27) в каждый момент времени, и к данному состоянию приводится портфель к концу цикла. В начале цикла записывается диагностическая информация в файл. Далее определяется был ли сплит – изменение количества акций, что отражено на цене в другой таблице, но в сумме стоимость актива не меняется. Сплит будет происходить если в строке (см. рисунок Рис. 27) будут значения

отличные от единицы. Также проверяется условие, что есть все цены по активам, из которых состоит портфель. Это необходимо для вычисления стоимости портфеля, что в свою очередь влияет на вычисление стоимости одной части портфеля, при стратегии равновзвешенного портфеля. Если стоимость портфеля невозможно вычислить, тогда с портфелем не будет происходить никаких манипуляций, и цикл перейдет к следующему шагу.

```
...
for(i in 1:nrow(action)){
  # запись даты
  if(diag) write(paste0("-----
",index(action)[i],"-----"),log,,TRUE)
  # если был сплит активов
  n.act=action[i,names(portfel)]
  if(length(n.act)>0) n.act=n.act[1,!is.na(n.act)]
  if(length(n.act)>0) n.act=n.act[1,n.act>1]
  if(length(n.act)>0)
portfel[colnames(n.act)]=portfel[colnames(n.act)]*n.act
  # если нет цен по всем активам в портфеле
  pr=prices[i,names(portfel)]
  if(length(pr)>0) pr=pr[1,!is.na(pr)]
  if(length(pr)!=length(portfel)){
    portfel.history[i]=equity
    if(diag) write(paste0("По данным активам торги
остановлены:
",paste(setdiff(names(portfel),colnames(pr)),collapse = "
")),log,,TRUE)
    next
  }
}
...
```

Рис. 29. Начало основного цикла по построению инвестиционного портфеля

Далее вычисляется стоимость портфеля путем умножения цены активов на количество данных активов в портфеле. В сам массив для портфеля записывается количество активов, и каждому элементу массива присваивается соответствующее наименование актива для дальнейшей идентификации. Затем в файл логов записывается информация о состоянии портфеля, состояние банка или денежных средств для инвестирования, а также о количестве активов в портфеле. Во временную переменную `buu` записывается новый состав портфеля, который будет достигнут к концу данной итерации цикла. Если портфель пустой и покупать никаких активов не нужно, тогда происходит

переход цикла к следующей итерации. Далее происходит вычисление веса одной части, который может изменяться в диапазоне от нуля до единицы. Также вычисляется стоимость одной части в зависимости от стоимости портфеля.

```
...
# стоимость портфеля

equity=sum(as.numeric(prices[i,names(portfel)]*portfel))+bank
# запись диагностической информации
if(diag) write(paste("Стоимость
портфеля:",equity),log,,TRUE)
if(diag) write(paste("Банк:",bank),log,,TRUE)
if(diag)
write(paste("Портфель:",paste(paste0(names(portfel),":"),portfe
l,collapse=",")),log,,TRUE)
if(diag)
write(paste("Количество:",length(portfel)),log,,TRUE)
# новый список активов для портфеля
buy=colnames(action)[which(!is.na(action[i,]))]
# запись стоимости портфеля в историю изменений
portfel.history[i]=equity
# если активы в портфеле такие же, в которые нужно
инвестировать
if(length(setdiff(buy,names(portfel)))==0) next

if(diag) write(paste("Новый портфель:",paste(buy,collapse =
" ")),log,,TRUE)
if(diag) write(paste("Количество:",length(buy)),log,,TRUE)

# если портфель пустой и покупать ничего не надо
if(length(portfel)==0 & length(buy)==0) next
# вес одной части в портфеле
part=0
# если покупать ничего не надо
if(length(buy)==0) {
  # будет ребалансировка
  if (length(portfel)!=0) part=1/length(portfel)
} else {
  # часть каждого актива (равновзвешенный)
  part=1/length(buy)
}
# стоимость одной части портфеля
part.money=equity*part
if(diag) write(paste("Вес 1 части:",part),log,,TRUE)
...
```

Рис. 30. Определения активов для инвестирования и состояния активов в портфеле

Далее происходит заполнение двух массивов, на основе которых будет происходить покупка и продажа активов. Первый массив активов на покупку вычисляется как различие между новым состоянием портфеля (переменной buy) и текущим состоянием портфеля. И массив на продажу – разница между текущим состоянием портфеля и новый портфелем активов.

```
...  
# новые активы, которые нужно купить в портфель: разница между  
новым списком активов для портфеля и текущим портфелем  
buy.new=setdiff(buy,names(portfel))  
  
# активы из портфеля, которые нужно продать: не входят в  
новый список активов для портфеля  
sell.new=setdiff(names(portfel), buy)  
...
```

Рис. 31. Вычисление активов для покупки и для продажи

Первое из четырёх действий с портфелем – это продажа активов. Если есть активы на продажу, то в цикле по каждому активу определяется его текущая цена, затем в промежуточную переменную записывается его количество в портфеле. Вычисляется размер комиссии на продажу, и цена корректируется в соответствии с комиссией. В конце банк – размер денежных средств пополняется на величину стоимости актива в портфеле за вычетом комиссии.

```

#### сначала продаём активы из портфеля, чтобы пополнить
количество денежных средств ####
    if(length(sell.new)>0) {
        if(diag) write(paste("#### Продажа
",paste(sell.new,collapse = " ")),log,,TRUE)
        for(p in sell.new){
            # текущая цены пифа
            price=as.numeric(prices[i,p])
            # если нет цены - переход к слдующему активу
            if(is.na(price)) next
            # вычисление комиссии
            fee=price * fee.sell
            # количество данного актива в портфеле
            n=as.numeric(portfel[p])
            if(diag) write(paste("ПИФ:",p,"Цена:",price,"Кол-
во:",n,"Итого:",n*price,"Комиссия:",n*fee),log,,TRUE)
            # изменение цены с учётом комиссии
            price=price - fee
            # продажа и пополнение количества денежных средств
            bank=bank + n*price
        }
        # исключение проданных пифов из портфеля
        portfel=portfel[! names(portfel) %in% sell.new]
    }

```

Рис. 32. Продажа активов из текущего портфеля

Проданные активы удаляются из портфеля. Данное действие стоит первым, так как необходимо пополнить количество свободных денежных средств перед покупкой других активов. В случае если все активы портфеля проданы, то происходит переход цикла на следующую итерацию.

```

...
    # сохранение информации о стоимости портфеля
    portfel.history[i]==sum(as.numeric(prices[i,names(portfel)]
*portfel))+bank
    # портфель стал пустым и покупать ничего не нужно
    if(length(portfel)==0 & length(buy.new)==0) next
...

```

Рис. 33. Сохранение стоимости и проверка состояния портфеля

Следующее действие – это ребалансировка оставшихся активов в портфеле. В данном действии будет происходить изменение количества актива в портфеле в зависимости от того, какую часть портфеля занимает его стоимость. Если стоимость актива больше величины, отведенной на одну часть

портфеля, тогда будет продано такое количество данного актива, чтобы удовлетворить условию равновзвешенности портфеля. Вначале берётся цена актива и его количество, после чего вычисляется стоимость актива, и далее считается – какую часть портфеля занимает данный актив. Если данное значение превышает то, которое необходимо, тогда происходит продажа части данного актива. Происходит вычисление денежной разницы между двумя значениями, и на основе этой разницы и текущей цены актива вычисляется количество, которое будет продано с округлением в большую сторону. Если продавать ничего не нужно, то цикл перейдет к следующему активу, иначе вычисляется комиссия на продажу, и цена актива корректируется в соответствии с комиссией. Величина актива в портфеле уменьшается на проданное количество, величина свободных денежных средств увеличивается по результату продажи. Если актива в портфеле больше не осталось, тогда актив удаляется из портфеля.

```

...
#### ребалансировка на продажу ####
if(diag) write("##### Допродажа #####",log,,TRUE)
for(p in names(portfel)){
  # текущая цены актива
  price=as.numeric(prices[i,p])
  # если нет цены - переход к следующему активу
  if(is.na(price)) next
  # количество пифа в портфеле
  n=as.numeric(portfel[p])
  # стоимость доли данного актива
  part.price=price*n
  # величина части в портфеле
  part.p=part.price/equity
  # если стоимости доли актива больше новой стоимости одной
части, то продажа избытка
  if(part.p > part){
    # разница для продажи
    diff.m=part.price - part.money
    # количество, которое будет продано (округление вверх)
    n.new= ceiling(diff.m / price)
    # если ничего не нужно продавать, то выход
    if(n.new==0) next
    # вычисление комиссии
    fee=price*fee.sell
    if(diag) write(paste("ПИФ:",p,"Цена:",price,"Кол-
во:",n.new,"Итого:",n.new*price,"Комиссия",n.new*fee),log,,TRUE
)
    # изменение цены с учётом комиссии
    price=price-fee
    # вычитание проданного количества пифов
    portfel[p]=portfel[p] - n.new
    # увеличение банка
    bank=bank + n.new*price
    # если был продан весь актив
    portfel[portfel[! portfel == 0]
  }
}
...

```

Рис. 34. Ребалансировка активов в портфеле, продажа

Третье действие – ребалансировка портфеля на покупку. Если в портфеле стоимость актива меньше, чем стоимость одной части портфеля, тогда необходимо докупить определенное количество данного актива для удовлетворения условию равновзвешенности портфеля. Данное действие похоже на предыдущее, комиссия на покупку в данном случае прибавляется к цене актива, а количество актива, доступное к покупке округляется в меньшую

сторону, и данное количество прибавляется к активу в портфеле, величина свободных средств уменьшается.

```
...
#### ребалансировка на покупку ####
if(diag) write("##### Допокупка #####",log,,TRUE)
for(p in names(portfel)){
  # текущая цены актива
  price=as.numeric(prices[i,p])
  # если нет цены - переход к следующему активу
  if(is.na(price)) next
  # количество актива в портфеле
  n=as.numeric(portfel[p])
  # стоимость актива в портфеле
  part.price=price*n
  # величина части в портфеле
  part.p=part.price/equity
  # если часть меньше, то нужно докупать
  if(part.p < part){
    # разница для покупки
    diff.m=part.money - part.price
    # вычисление комиссии
    fee=price*fee.buy
    # цена без комиссии
    price.old=price
    # цена с комиссией
    price=price + fee
    # количество, которое можно докупить
    n.new=floor(diff.m / price)
    # если ничего не надо докупать - переход к следующему
активу
    if(n.new==0) next
    if(diag) write(paste("ПИФ:",p,"Цена:",price.old,"Кол-
во:",n.new,"Итого:",n.new*price.old,"Комиссия:",n.new*fee),log,
,TRUE)
    # добавление купленного количества актива в портфель
    portfel[p]=portfel[p] + n.new
    # уменьшение свободных денежных средств
    bank=bank-n.new*price
  }
}
...
```

Рис. 35. Ребалансировка активов в портфеле, покупка

Последнее действие – покупка новых активов, которые отсутствуют в портфеле. Вначале происходит проверка – есть ли такие активы. Далее есть также проверка, что, если портфель будет состоять из одного актива, тогда одна

часть портфеля будет равняться самому портфелю и её стоимость будет равняться значению свободных денежных средств.

В цикле по каждому активу берётся его текущая цена, вычисляется комиссия, цена корректируется на комиссию, вычисляется количество данного актива с округлением в меньшую сторону до целого числа, которое можно приобрести в зависимости от цены, включающую комиссию.

```
...
#### покупка новых пифов ####
if(length(buy.new)>0) {
  if(diag) write(paste("#### Покупка
",paste(buy.new,collapse = " ")),log,,TRUE)
  # если портфель - это оди актив, то стоимость одной части
  # это все доступные денежные средства
  if(part==1) part.money=bank
  for(p in buy.new){
    # текущая цены актива
    price=as.numeric(prices[i,p])
    # если нет цены - переход к следующему активу
    if(is.na(price)) next
    # вычисление комиссии
    fee=price*fee.buy
    # цена без комиссии
    price.old=price
    # цена с комиссией
    price=price+fee
    # количество актива, которое можно купить
    n=floor(part.money / price)
    if(diag)
write(paste("Бюджет:",part.money,"ПИФ:",p,"Цена:",price.old,"Ко
л-во:",n,"Итого:",n*price.old,"Комиссия:",n*fee),log,,TRUE)
    # если ничего не купить, то выход
    if(n==0) next
    # добавление актива в портфель
    portfel=append(portfel,n)
    # уменьшение свободных денежных средств
    bank=bank - n*price
    # добавление названия актива в портфель
    names(portfel)[length(portfel)]=p
  }
}
...
```

Рис. 36. Покупка новых активов в портфель

В конце данный актив добавляется в портфель, величина доступных денежных средств уменьшается на полную стоимость актива, в портфеле присваивается соответствующее наименование для добавленного актива.

Итерация цикла заканчивается подсчётом стоимости портфеля, который в результате вышеописанных действий был приведён к новому состоянию активов и ребалансирован, как равновзвешенный портфель с учётом комиссий на покупку и продажу. Также в файл записывается диагностическая информация о последнем состоянии портфеля на данной итерации основного цикла.

```
#####  
  if(diag) write("#####",log,,TRUE)  
  # вычисление стоимости портфеля  
  
equity=sum(as.numeric(prices[i,names(portfel)]*portfel))+bank  
  if(diag) write(paste("Стоимость  
портфеля:",equity),log,,TRUE)  
  if(diag) write(paste("Банк:",bank),log,,TRUE)  
  if(diag)  
write(paste("Портфель:",paste(paste0(names(portfel),":"),portfe  
l,collapse=",")),log,,TRUE)  
  if(diag)  
write(paste("Количество:",length(portfel)),log,,TRUE)  
  # сохранение информации о стоимости портфеля  
  portfel.history[i]=equity  
}  
# в качестве результата возвращается  
# таблица данных о временной стоимости портфеля  
return(portfel.history)  
}
```

Рис. 37. Конец основного цикла, запись состояния портфеля

После окончания работы цикла в качестве результата функция возвращает таблицу с данными о временной стоимости портфеля, другими словами, в таблице отражено, как стоимость портфеля изменялась во времени.

Ниже на рисунке приведён пример вызова данной функции, где были переданы в качестве параметров таблица стоимости паев индексных фондов (сформирована в разделе подготовки данных), таблица индикаторов фондов, которые входят в портфель (см. Рис. 27), величина свободных денежных

средств и данные о комиссии: 0,5% на покупку и 1,5% на продажу. В терминах паевых фондов данные комиссии называются надбавка и вознаграждение фонду соответственно. После того, как результат получен, в цикле происходит поиск порядкового номера даты, с которой началось инвестирование в индексные паевые фонды, до этого момента портфель состоял только из свободных денежных средств. Вычисляется пересечение дат с датами Индекса МосБиржи для сравнения данных, и началом массива дат становится начало инвестирования. Данные объединяются в одну таблицу, с соответствующим наименованием столбцов. С использованием функции `charts.PerformanceSummary` из пакета `PerformanceAnalytics` строится график ниже (см. Рис. 41).

```
# вызов функции с параметрами
portfel.history=portfel.equity(pif.price, pif.act, 1000000,
c(0.005, 0.015))
# поиск номера даты, когда портфель
# начал формироваться
for(i in 1:nrow(pif.act)){
  if(sum(pif.act[i,],na.rm = TRUE)>0){
    break
  }
}
# пересечение дат с индексом
dt=intersect(index(portfel.history),index(moex))
dt=dt[i:length(dt)]
# создание общего массива
test=cbind(moex[as.Date(dt)],Return.calculate(portfel.history[a
s.Date(dt)]))
# наименование столбцов
colnames(test)=c("MOEX", "Portfel")
# построение графика
charts.PerformanceSummary(test)
# регрессия
summary(lm(test[,2]~test[,1]))
```

Рис. 38. Сравнение построенного индексного портфеля и Индекса МосБиржи

Для подтверждения того, насколько близко полученный портфель повторяет Индекс МосБиржи, можно построить линейную регрессию. Как видно по характеристикам Estimate значение альфа – Intercept близко к нулю, а коэффициент бета – `test[,1]` близок к единице, что говорит о том, что портфель

по параметру альфа не обыгрывает индекс и по параметру бета максимально следует изменениям индекса, о чем также говорит коэффициент значимости, близкий к нулю.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0004491   0.0011530   0.389    0.698
test[, 1]    0.9773629   0.0158550  61.644 <2e-16 ***
```

Рис. 39. Результат линейного регрессии индексного портфеля на Индекс МосБиржи

Следует отметить, что важной особенностью построение бенчмарка на основе инвестиционного портфеля является количество начальных денежных средств. Как видно из примера ниже, при величине инвестиций в 10000, коэффициент бета снизился с 0,98 до 0,8. Это означает что инвестиционный портфель в меньшей степени следует Индексу МосБиржи. Данное снижение коэффициента бета можно объяснить тем, что при недостаточном уровне инвестиций и разной стоимости паев инвестиционных фондов, не все паевые фонды добавляются в портфель.

```
# бенчмарк портфель из индексных паевых фондов
benchmark.pif=portfel.equity(pif.price, pif.act, 10000,
c(0.005, 0.015))

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0006416   0.0011913   0.539    0.591
test[, 1]    0.8061019   0.0163817  49.207 <2e-16 ***
```

Рис. 40. Инвестиционный портфель с другим значением начальных инвестиций

MOEX Performance

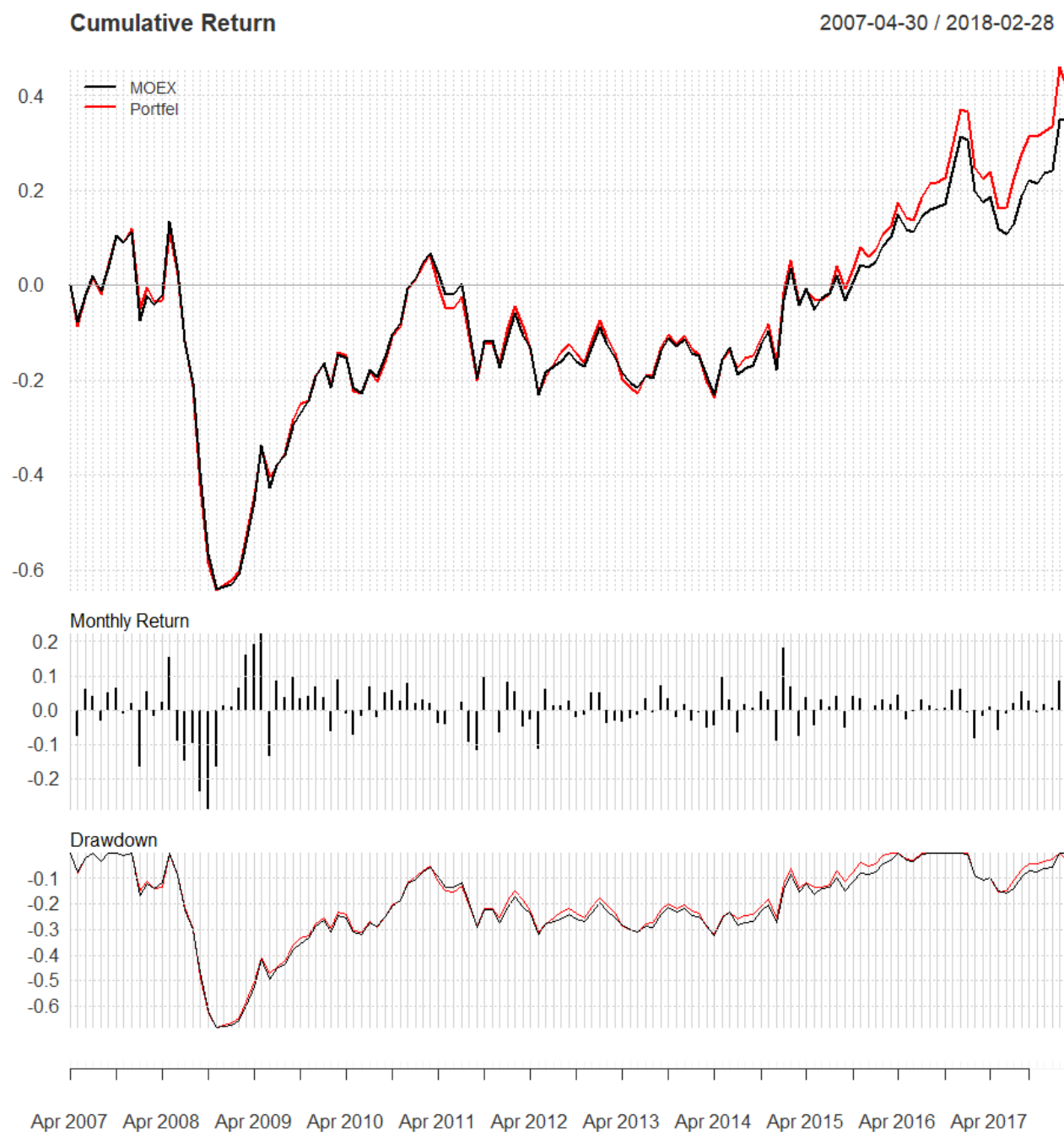


Рис. 41. Графики доходности индексного портфеля и Индекса МосБиржи

По графику также видно, что построенный портфель несмотря на комиссию максимально точно следует изменениям Индекса МосБиржи. Наиболее близкое следование индексу говорит о том, что в портфеле

отсутствуют дополнительные риски, иначе доходность портфеля и индекса различалась.

2.3. Инвестиционный портфель на основе акций из Индекса ММВБ 10

Для построения портфеля на основе акций из Индекса ММВБ 10 в соответствии с историей изменений базы расчёта данного индекса необходимо создать таблицу с индикаторами, которые указывают на состав портфеля в каждый момент времени. Таблица `data.act` составляется по подобию таблицы с ценами из раздела по подготовке данных по данному индексу. В качестве строк таблицы выступает массив торговых дат индекса, в качестве столбцов – все акции, которые когда-либо входили в индекс. Далее цикл проходит по таблице с данными о базе расчёта на заданные даты в соответствии с архивными значениями. По данным датам и полученной базе расчёта в основной таблице ставится индикатор, равный единице, о том, что данные активы в заданные даты составляли Индекс ММВБ 10.

```
# структурированная таблица с данными
data.act=xts(matrix(NA, nrow = length(date.all), ncol =
length(mic.ticker),dimnames =
list(date.all,mic.ticker)),order.by = as.Date(date.all))
for(i in 18:nrow(mic.10)){
  # база расчёта индекса
  tickers=unlist(mic.10[i,3:length(mic.10[i,])])
  # удаление пустых значений
  tickers=tickers[tickers!=""]
  # период действия базы расчёта
  dates=seq.Date(as.Date(mic.10[i,1],"%d.%m.%Y"),
as.Date(mic.10[i,2],"%d.%m.%Y"), by = "day")
  # заполнение индикаторов, о вхождении акций в портфель
  data.act[dates,tickers]=TRUE
}
# сплит акций
data.act["2007-07-18","SBER"]=1000
data.act["2007-07-18","SBERP"]=20
# заполнение индикаторов
# чтобы акции не были исключены из портфеля
data.act[c("2007-07-19","2007-07-20"),c("SBER","SBERP")]=TRUE
```

Рис. 42. Заполнение данных по акциям из базы расчёта Индекса ММВБ 10

В июле 2007 года происходил сплит акций ПАО «Сбербанк»: одна обыкновенная акция делилась из расчёта 1:1000, привилегированная акция в соотношении 1:20. Стоимость обыкновенной акции соответственно уменьшалась в 1000 раз, и стоимость привилегированной акции становилась меньше в 20 раз. Таким образом, суммарная стоимость актива в портфеле оставалась неизменной. Данный сплит соответствующим образом отражен в таблице, где вместо 1 будут значения 1000 и 20. Алгоритмом в процессе составления портфеля сплит будет учтён. Также на дальнейшие несколько дат ставится индикатор о том, чтобы акции ПАО «Сбербанк» оставались в портфеле, несмотря на то, что цена за 19 и 20 июля 2017 года не были установлены, так как происходило деление акций – сплит. Если индикаторы будут отсутствовать, то алгоритм будет трактовать это как исключение актива из портфеля.

Происходит вызов функции с передачей в качестве параметров: таблицы со стоимостью акций, таблицы индикаторов, о том, какие акции входят в портфель, величины свободных денежных средств и комиссии. Для акций параметр величины комиссии на продажу и покупку составил 0.01%. Далее полученный результат переводится в месячные данные, и сохраняются только цены на конец месяца. После чего вычисляется месячная доходность. Данный шаг отсутствовал при построении графика портфеля индексных паевых инвестиционных фондов по двум причинам. Первая причина, что изменение базы расчёта могло происходить в середине месяца, другими словами, база расчета могла меняться в определенную дату, поэтому нельзя сократить изначальные данные до месячных значений, поэтому использовались дневные. Вторая причина заключается в том, что функция возвращает таблицу, во временном формате данных, название строк которой, такое же, как и у таблицы с индикаторами акций для портфеля. Внутри функции по программному коду происходит копирование структуры данной таблицы.

После того, как результат получен, вычисляется пересечение дат портфеля и Индекса ММВБ 10 для построения совместного графика. Данные

объединяются в одну временную таблицу с соответствующим названием столбцов. И функцией `charts.PerformanceSummary` из пакета `PerformanceAnalytics` строится график. Далее строится линейная регрессия для численного сравнения портфеля с индексом.

```
# вызов функции с параметрами
portfel.history=portfel.equity(data.10, data.act, 1000000,
с(0.0001, 0.0001))
# перевод к месячной доходности
portfel.history=monthlyReturn(Cl(to.monthly(portfel.history,
indexAt = 'lastof'))))
# пересечение дат с индексом
dt=intersect(index(portfel.history),index(moex10))
# создание общего массива
test=cbind(moex10[as.Date(dt)],portfel.history[as.Date(dt)])
# наименование столбцов
colnames(test)=c("MOEX10","Portfel")
# построение графика
charts.PerformanceSummary(test)
# регрессия
summary(lm(test[,2]~test[,1]))
```

Рис. 43. Сравнение построенного портфеля с Индексом ММВБ 10

Данная линейная регрессия соответствует модели САРМ. Коэффициент альфа – Intercept близок к нулю, это означает, что портфель не обыгрывает и не проигрывает бенчмарку – Индексу ММВБ 10. Коэффициент бета – `test[, 1]` близок к единице, значит портфель максимально полно следует данному индексу. Результат линейной регрессии говорит о том, что построенный равновзвешенный портфель из акций, которые входили в индекс, максимально близко повторяет основной Индекс ММВБ 10, даже с учетом комиссий.

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.0001937	0.0010304	0.188	0.851
test[, 1]	0.9825513	0.0124612	78.849	<2e-16 ***

Рис. 44. Результат регрессии портфеля на Индекс ММВБ 10

Ниже продемонстрировано графическое сравнение кумулятивной доходности построенного портфеля и Индекса ММВБ 10. График равновзвешенного портфеля из акций, которые входили в индекс, максимально близко повторяет график Индекса ММВБ 10.

Первый график – это график кумулятивной доходности. Второй график показывает гистограмму месячной доходности Индекса ММВБ 10. Нижний график демонстрирует отрицательную доходность, как по построенному портфелю, так и по индексу.

MOEX10 Performance



Рис. 45. Графики доходности портфеля и Индекса ММВБ 10

Данный портфель можно использовать в качестве бенчмарка, так как он максимально полно следует Индексу ММВБ 10, что говорит, об отсутствии дополнительных рисков, которые бы повлияли на доходность, иначе это было бы отражено на графике.

2.4. Инвестиционный портфель на основе фьючерса на Индекс МосБиржи

Для составления портфеля на основе фьючерса на Индекс МосБиржи с ближайшей датой исполнения применяется другой алгоритм. Поскольку при торговле фьючерсами есть важные отличия от торговли активами на примере паевых фондов или акций. Первое отличие заключается в том, что фьючерс – это заключение контракта по текущей цене, и на момент заключения контракта резервируется сумма гарантийного обеспечения, которая нужна для того, чтобы обезопасить выполнение контракта. Прибыль получается в том случае, если будет заключен противоположный контракт с более высокой ценой, либо в последний день торговли фьючерсом его расчётная цена окажется выше цены при заключении контракта. Фьючерс может быть поставочным или расчётным, в случае поставочного инструмента по контракту поставляется товар, а в случае расчётного одна из сторон контракта перечисляет другой разницу между текущей ценой инструмента и ценой инструмента в контракте. Фьючерс на Индекс МосБиржи является расчётным. Размер гарантийного обеспечения меньше, чем цена контракта, что создаёт эффект плеча, когда при незначительном движении цены контракта можно получить как большую доходность, так и больший убыток. Это достигается за счёт того, что на одну и ту же сумму денежных средств можно инвестировать в больший объем контрактов, чем покупая их за реальную цену.

Другое отличие заключается в том, что каждый торговый день изменяется расчётная цена контракта. В результате чего, в конце каждого торгового дня между сторонами, заключившими контракт, возникает маргинальное

требование, которое необходимо перечислить одной из сторон. Фьючерсный контракт для обеих сторон это игра с нулевой суммой – одна из сторон гарантированно получит убыток. Маржинальное требование является убытком для одной стороны и прибылью для другой. Для той стороны, кто заключил длинную позицию по фьючерсу – купил фьючерс, если произойдёт повышение расчётной цены по отношению к расчётной цене предыдущего торгового дня, то маржинальное требование будет положительным, значит для данной стороны по фьючерсу будет прибыль.

Торговля фьючерсами сводится к изменению размера гарантийного обеспечения и величины маржинального требования, которые не зависят от клиента. В случае увеличения размера гарантийного обеспечения необходимо резервировать дополнительные денежные средства, и наоборот, в случае уменьшения – часть средств освобождается. Отрицательное маржинальное требование так же нужно удовлетворять в конце каждого торгового дня из свободных денежных средств, что означает, что у клиента должен быть запас денежных средств. Размер гарантийного обеспечения изменяется вследствие резкого изменения цены фьючерса в одну из сторон. Гарантийное обеспечение также зависит от волатильности базового актива, в данном случае Индекса МосБиржи.

Перед построением портфеля задаются изначальные параметры: величина свободных денежных средств для инвестирования, наименование фьючерса, где отсутствие наименования означает, что портфель пустой, а также таблица, в которой будет вестись история изменения стоимости портфеля.

```
# величина свободных денежных средств
portfel.bank=1000000
# наименование текущего фьючерса в портфеле
portfel.name=""
# история изменения стоимости портфеля
portfel.history=future.data[,1]
# стирание данных
portfel.history[]=NA
```

Рис. 46. Начальные параметры для портфеля на основе фьючерса

Ниже приведён сам алгоритм формирования портфеля, где основной цикл похож на цикл для паев фондов или акций, описанный выше. Цикл проходит по таблице с индикаторами, указывающими на то, какой фьючерс должен быть в портфеле. Далее в переменную `buy` записывается сокращенное наименование фьючерса с ближайшей датой исполнения. В случае если портфель не пустой и фьючерс в портфеле не совпадает с данным, тогда происходит освобождение гарантийного обеспечения, и данная сумма возвращается в свободные денежные средства.

```
for(i in 1:nrow(future.data)){
  # фьючерс в который нужно инвестировать на данной итерации
  buy=colnames(future.action)[which(!is.na(future.action[i,]))]
  # если фьючерс отличается от фьючерса в портфеле
  # и портфель не пустой то происходит освобождение ГО
  if(buy!=portfel.name & portfel.name!=""){
    # ГО возвращается в свободные средства
    portfel.bank=portfel.bank + portfel.warranty
    # удаление переменных
    rm(portfel.warranty,portfel.count,portfel.price)
  }
  ...
}
```

Рис. 47. Определение фьючерса для портфеля, освобождение ГО

Если наименование фьючерса не совпадает с фьючерсом в портфеле, значит далее произойдет инвестирование в новый фьючерс. Вначале записывается в переменную `price` цена фьючерса, в качестве которой выступает размер гарантийного обеспечения, которое будет зарезервировано, но по истечению срока контракта будет возвращено. Далее вычисляется количество фьючерсов, которое будет добавлено в портфель. В данном случае не будет использован эффект плеча, поскольку нет цели получить какую-либо доходность, поэтому в качестве цены подставляется полная стоимость фьючерса. В результате объём инвестиций в фьючерс полностью покрывает портфель, но будет лишь зарезервировано гарантийное обеспечение, оставшиеся свободные средства будут использованы в случае изменения гарантийного обеспечения и покрытия маржинального требования, которое может быть значительным. Затем свободные денежные средства уменьшаются

на размер гарантийного обеспечения, и данная инвестированная сумма сохраняется в другую переменную `portfel.warranty`. Также записывается наименование нового фьючерса в переменную `portfel.name`.

Стоит упомянуть, что, поскольку фьючерс – это контракт на базовый актив, то по контракту количество базового актива может быть больше одного. В данном случае фьючерс на Индекс МосБиржи включает в себя сто базовых активов. Или, другими словами, расчётная цена фьючерса вычисляется как среднее значения Индекса МосБиржи за период с 15:00 до 16:00 в последний день заключения контракта, умноженное на 100.

```
...
# если новый фьючерс и фьючерс в портфеле не совпадают
# то нужно инвестировать в данный фьючерс
# ГО было освобождено на предыдущем шаге
if(buy!=portfel.name){
    # текущая стоимость фьючерса - это ГО
    price=as.numeric(future.data[i,buy])
    # инвестиции без "плеча"
    # цена количества фьючерсов покрывает весь банк
    n=ceiling(portfel.bank / as.numeric(future.price[i,buy]))
    # свободные денежные средства
    # уменьшаются только на величину ГО
    portfel.bank=portfel.bank - n * price
    # сохранение количества
    portfel.count=n
    # сохранение наименования фьючерса
    portfel.name=buy
    # цена фьючерса для вычисления маржинального требования
    portfel.price=as.numeric(future.price[i,buy])
    # общий размер ГО
    portfel.warranty=n * price
    # удаление переменных
    rm(price,cash,n)
}
...
```

Рис. 48. Заключение контракта по фьючерсу

Если в портфеле есть фьючерс на Индекс МосБиржи, то далее происходит расчёт актуального размера гарантийного обеспечения на данную итерацию цикла. Если новое значение гарантийного обеспечения больше того, что зарезервировано, тогда из свободных денежных средств данная разница перераспределяется. В случае, если требуемое гарантийное обеспечение

меньше, чем зарезервировано, тогда эта разница добавляется к свободным средствам. В результате выполняется условие обеспечения необходимого уровня ГО (гарантийное обеспечение).

Далее вычисляется размер маржинального требования, которое в случае отрицательного значения, вследствие уменьшения расчётной цены по отношению к предыдущему торговому дню, будет выплачено из свободных денежных средств другой стороне по контракту. В случае положительного значения – получено от другой стороны и добавлено к свободным средствам. Маржинальное требование в день покупки рассчитывается как разница между ценой покупки и расчётной ценой на данный день.

```
...
# если в портфеле есть фьючерс
if(portfel.count > 0){
  # актуальное значение ГО
  warranty.price = as.numeric(future.data[i,portfel.name])
  if(!is.na(warranty.price)){
    # вычисление актуального ГО покрывающее портфель
    warranty=warranty.price * portfel.count
    # разница между зарезервированным и актуальным ГО
    diff.warranty = portfel.warranty - warranty
    # изменение величины свободных средств
    portfel.bank = portfel.bank + diff.warranty
    # зарезервированное ГО становится актуальным
    portfel.warranty = warranty
  }
  # вычисление маржинального требования

  if(portfel.price==as.numeric(future.price[i,portfel.name])){
    # инвестировано было в данный день
    margin=(as.numeric(future.price.estim[i,portfel.name]) -
portfel.price) * portfel.count
  } else {
    # между двумя датами как разница расчётных цен
    margin=(as.numeric(future.price.estim[i,portfel.name]) -
as.numeric(future.price.estim[(i-1),portfel.name])) *
portfel.count
  }...
}
```

Рис. 49. Расчёт гарантийного обеспечения и маржинального требования

Если количества зарезервированного гарантийного обеспечения и величины свободных денежных средств в сумме не хватает на покрытие

маржинального требования, то возникает ситуация “margin call” – произойдет принудительное закрытие позиции, контракт по фьючерсу будет расторгнут автоматически, если по правилам биржи не внести требуемую сумму до конца торговой сессии следующего дня. Принудительное закрытие позиции означает большой убыток для клиента. В случае длинной позиции величина маржинального требования резко увеличивается из-за большого падения цены, в случае короткой позиции, когда клиент продавал фьючерс, из-за большого роста расчётной цены.

В конце каждой итерации цикла сохраняется стоимость портфеля, которая состоит из зарезервированного гарантийного обеспечения и свободных денежных средств.

```
...
# если свободных средств и ГО не хватает
# для покрытия маржинального требования
if((portfel.bank + portfel.warranty) < margin){
    # происходит margin call
    print("Банк опустел")
    break
}
# в конце каждого торгового дня свободные средства
# изменяются на величину маржинального требования
portfel.bank = portfel.bank + margin
# удаление переменных
rm(warranty,margin,diff.warranty,warranty.price)
}
portfel.history[i]=portfel.bank + portfel.warranty
}
```

Рис. 50. Ситуация margin call, сохранение стоимости портфеля

В результате данного алгоритма сформирована история стоимости портфеля, который инвестируется во фьючерс на Индекс МосБиржи с ближайшей датой исполнения. Данный портфель имеет дневную доходность, которая далее переводится в месячную, после чего сравнивается с Индексом МосБиржи.

```

# перевод к месячной доходности
portfel.history=monthlyReturn(Cl(to.monthly(portfel.history,
indexAt = 'lastof'))))
# пересечение дат с индексом
dt=intersect(index(prtf),index(moex))
# создание общего массива
test=cbind(moex[as.Date(dt)],portfel.history)
# наименование столбцов
colnames(test)=c("МОЕХ", "Portfel")
# построение графика
charts.PerformanceSummary(test)
# регрессия
summary(lm(test[,2]~test[,1]))

```

Рис. 51. Сравнение портфеля с Индексом МосБиржи

По значению альфы Intercept из линейной регрессии данного портфеля на индекс можно говорить, что доходность близка к нулю. Значение беты test[, 1] указывает на то, что портфель следует индексу, но не полностью совпадает с ним, и данный коэффициент также является значимым.

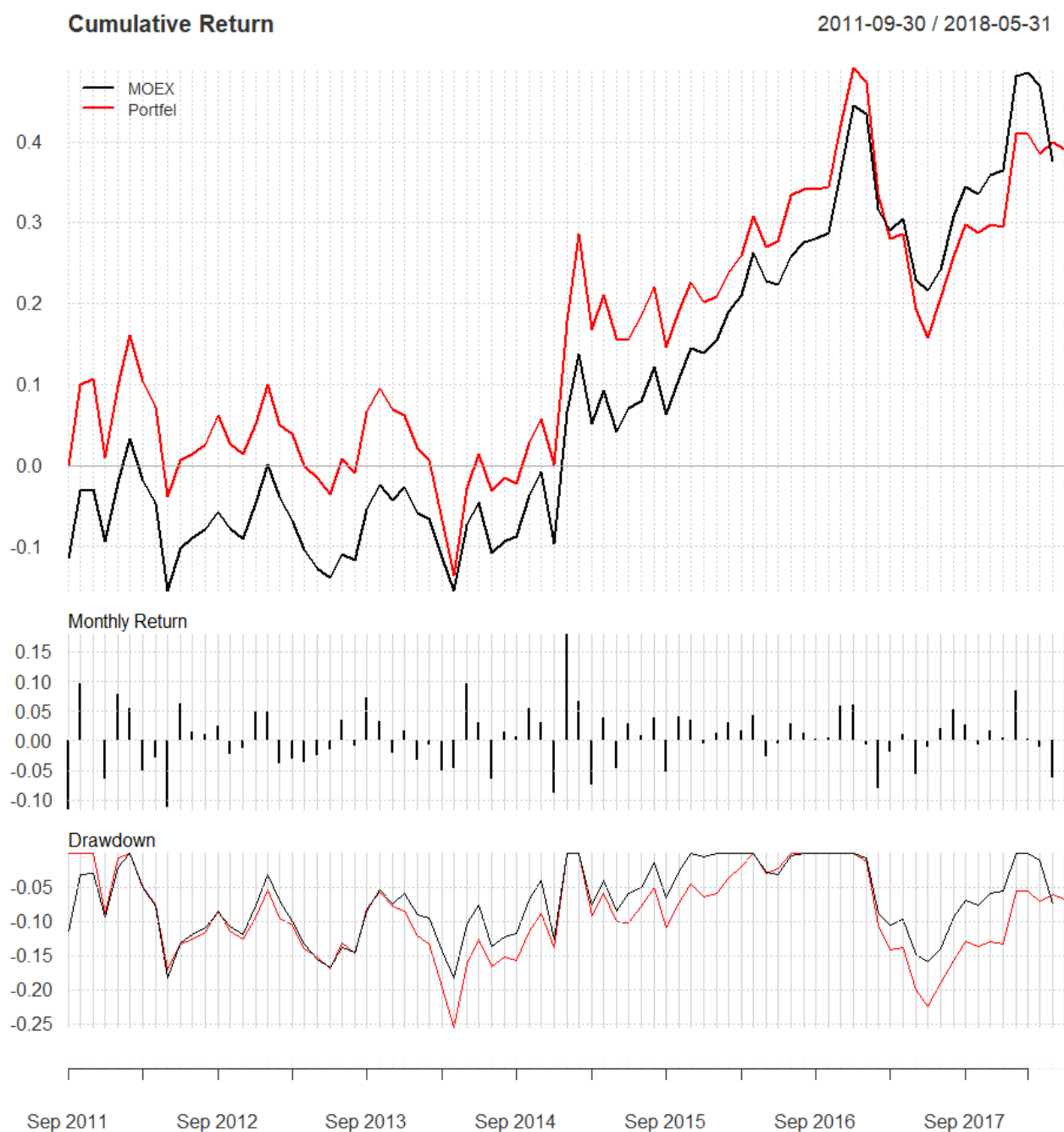
Coefficients:				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.0006133	0.0022059	0.278	0.782
test[, 1]	0.9263029	0.0439268	21.087	<2e-16 ***

Рис. 52. Результат регрессии портфеля на Индекс МосБиржи

Ниже представлено графическое сравнение построенного портфеля и Индекса МосБиржи. По графику видно, что портфель повторяет индекс, но не совпадает с ним. Когда график портфеля выше графика индекса, то это означает, что по данному фьючерсу происходит ситуация контанго – когда рынок оценивает будущую цену базового актива выше, чем цена на текущий момент, или в данном случае – значение Индекса МосБиржи. Контанго наблюдалось до октября 2016 года. Далее происходила ситуация бэквордация – когда рынок оценивает будущую стоимость базового актива фьючерса ниже его текущей стоимости. Поэтому график портфеля и Индекса МосБиржи не совпадают. Также не учитывались комиссии биржи при составлении портфеля, так как комиссии на Московской Бирже по фьючерсам составляют

фиксированную величину в три рубля за контракт, и данная величина мала по сравнению со стоимостью портфеля.

MOEX Performance



3. Бутстрап модель

3.1. Описание бутстрап подхода

Цель бутстрап подхода определить паевые инвестиционные фонды, которые получили не случайную избыточную доходность альфа относительно других фондов. В данной реализации бутстрап модели будет использована модель CAPM для определения характеристик фонда, таких как альфа и бета коэффициент, а также t-статистика, по отношению к бенчмарку. В качестве бенчмарка будут использованы три сформированных портфеля, которые были построены на основе Индекса МосБиржи и Индекса ММВБ 10. При построении данных портфелей преследовалась цель использовать финансовые инструменты, которые доступны инвестору для возможности инвестировать в рыночный индекс. Для портфеля на основе Индекса МосБиржи использовались два инструмента: индексные паевые инвестиционные фонды и фьючерсы на Индекс МосБиржи. Для портфеля на основе Индекса ММВБ 10 при помощи архивной базы расчёта данного индекса в равновзвешенный портфель были включены акции, которые когда-либо входили в состав индекса.

Технически модель CAPM реализуется с использованием линейной регрессии избыточной доходности инвестиционного портфеля, в качестве которого выступает определённый паевой инвестиционный фонд, на избыточную доходность бенчмарка. Избыточная доходность вычисляется как доходность инвестиционного портфеля или бенчмарка после вычета доходности по безрисковой ставке. В данной работе в качестве безрисковой ставки выступает кривая бескупонной доходности однолетних государственных облигаций Российской Федерации.

Особенностью бутстрап модели является удаление реального альфа коэффициента из модели CAPM и создание доходности данного инвестиционного портфеля заново без данного коэффициента. Альфа коэффициент является безрисковой доходностью портфеля, или, в данной работе

в качестве портфеля выступает паевой инвестиционный фонд, где альфа коэффициент можно назвать навыком управляющего фондом. После удаления альфа коэффициента из модели это означает, что из модели убрали навык управляющего. После чего снова строится регрессия созданной искусственно доходности фонда на бенчмарк. В результате данной регрессии получают симулированные коэффициенты, и новое значение альфа коэффициента говорит о том, влияет ли навык управляющего на доходность фонда или нет.

Регрессия с симулированными доходностями повторяется много раз, в результате чего получается массив альфа коэффициентов по одному паевому фонду. Затем данный метод повторяется для всех паевых инвестиционных фондов. В результате получается один массив реальных альфа коэффициентов, который включает в себя все фонды. И определенное количество наборов с симулированными альфа коэффициентами, количество данных массивов зависит от количества повторений регрессии с симулированными данными.

В качестве сравнения фондов друг с другом вместо альфа коэффициентов будет использоваться t-статистика коэффициента альфа, поскольку данная характеристика включает в себя волатильность доходности, полученной фондом. При сравнениях фондов между собой это говорит о том, что при двух одинаковых доходностях фондов будет лучше та, которая получена с меньшей волатильностью.

На последнем этапе бутстрап подхода все симулированные альфа коэффициенты или t-статистики, в данной работе, объединяются в один большой массив. Из данного массива выбираются лучшие значения из распределения массива, а точнее, выбираются значения из правого хвоста распределения с заданным уровнем значимости, который обычно выбирается на интервале от 97,5% до 99%. В конце реальные коэффициенты паевых инвестиционных фондов, альфа коэффициенты или t-статистики, полученные на первом этапе, сравниваются с выборкой лучших симулированных коэффициентов аналогичного типа. Фонды, коэффициенты которых вошли в данную симулированную выборку, отмечаются как лучшие фонды, среди

остальных фондов из данной выборки. В результате безрисковая доходность выбранных фондов была получена не случайно, что говорит о влиянии навыков управляющего фондом на доходность данного фонда.

3.2. Алгоритм бутстрап подхода

- Исходные данные представляют собой выборку месячных доходностей по открытым не отраслевым паевым инвестиционным фондам акций, где n – количество фондов, и данные о доходности бенчмарка и безрисковой доходности по месяцам;

- Далее для каждого фонда вычисляется линейная регрессия $R_i \sim R^M$ (2), соответствующая модели CAPM, где R_i – избыточная доходность i -го фонда ($i = 1 \dots n$), R^M – избыточная доходность бенчмарка. Избыточная доходность вычисляется по формуле: $R = r - r_f$, где r – доходность портфеля или бенчмарка, r_f – безрисковая доходность. В данной работе в качестве безрисковой доходности используется бескупонная доходность однолетних государственных облигаций Российской Федерации;

- Формула линейной регрессии:

$$R_i = \alpha_i + \beta_i * R^M + \varepsilon_i, \quad (3)$$

где: R_i – избыточная доходность i -го фонда;

α_i – коэффициент альфа i -го фонда;

β_i – коэффициент бета i -го фонда;

R^M – избыточная доходность бенчмарка;

ε_i – временной ряд остатков по i -му фонду.

Данные коэффициенты являются оценками на реальных данных. Для сравнения фондов между собой будет использоваться t -статистика альфа коэффициента из данной регрессии. Данный коэффициент вычисляется по следующей формуле:

$$\bar{t} = \frac{\alpha}{\sigma}, \quad (4)$$

где: α – коэффициент альфа доходности паевого фонда;

σ – среднее квадратичное отклонение доходности паевого фонда;

В данном коэффициенте учитывается волатильность доходности паевого инвестиционного фонда.

– На данном шаге происходит восстановление избыточной доходности фонда по следующей формуле:

$$R_i^* = \beta_i * R^M + \hat{\varepsilon}_i ,$$

где: R_i^* – воссозданная избыточная доходность по i -му фонду;

β_i –реальный бета коэффициент i -го фонда;

R^M – избыточная доходность бенчмарка;

$\hat{\varepsilon}_i$ – перемешанные ошибки из регрессии на реальных данных.

– Необходимо снова построить линейную регрессию $R_i^* \sim R^M$, но с использованием восстановленной избыточной доходности паевого фонда;

– Затем из данной регрессии вычисляется симулированный коэффициент альфа α^* или t-статистика коэффициента альфа α^* , в зависимости от того, какая оценка будет использована для сравнения фондов;

– Данную регрессию можно повторить k раз, при этом на каждом шаге необходимо перемешивать ошибки из регрессии на реальных данных, $k \in N$;

– Из каждой регрессии для все паевых фондов как на реальных данных, так и на каждом k повторений регрессии на симулированных данных значение коэффициента альфа или, в данной работе, значение t-статистики коэффициента альфа сохраняется в таблицу, размером $n \times (k + 1)$:

$$\begin{bmatrix} t_1 & t_{1,1}^* & t_{1,2}^* & \cdots & t_{1,k+1}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_n & t_{1,n}^* & t_{2,n}^* & \cdots & t_{n,k+1}^* \end{bmatrix}, \quad (5)$$

где: t_i – t-статистика коэффициента альфа на реальных данных для i -го паевого фонда;

t_i^* – t-статистика коэффициента альфа на восстановленных данных для i -го паевого фонда;

n – количество паевых инвестиционных фондов;

k – количество повторений бутстрапов, оценки регрессии на восстановленных данных.

Количество строк в таблице равняется количеству паевых фондов, количество столбцов равняется количеству повторений бутстрапов плюс первый столбец с оценками коэффициентов на реальных данных.

– На завершающем этапе происходит сравнение паевых фондов и выбор лучших фондов из данной выборки. По каждому из k столбцов симулированных коэффициентов выбираются значения из правого хвоста распределения для каждого столбца. Уровень квантили для выбора может варьироваться от 97,5% до 99% в зависимости от того, какое значение квантили будет задано в реализации алгоритма. В результате после выборки из k столбцов сформируется массив симулированных коэффициентов, которые являются лучшими среди всех коэффициентов, полученных на восстановленных данных.

– На последнем этапе коэффициенты альфа или t-статистики коэффициентов альфа, полученные на реальных данных и находящиеся в первом столбце, сравниваются с массивом лучших симулированных коэффициентов. В данном случае можно задать условие простого вхождения фонда в данную выборку, чтобы отметить данный фонд как лучший. Также можно задать уровень квантили для выборки лучших симулированных коэффициентов, и, если реальный коэффициент фонда выше данного значения, тогда отмечать фонд как лучший по сравнению с другими фондами из выборки.

– В результате будет получен список паевых инвестиционных фондов, которые обыгрывают бенчмарк и остальные паевые фонды из выборки. При этом доходность, получаемая лучшими фондами, является статистически значимой и может сохраняться на длительном промежутке времени в будущем. Также данный результат показывает положительное влияние навыков

инвестиционного управляющего на получаемую безрисковую доходность фонда.

3.3. Применение бутстрап подхода

Реализация бутстрап модели была сделана в виде функции, которая принимает в качестве входных параметров следующие переменные:

- таблица с доходностями паевых инвестиционных фондов;
- доходность бенчмарка;
- доходность по безрисковой ставке;
- количество повторений бутстрапа k ;
- минимальная длина периода для оценки коэффициентов;
- уровень квантили при выборе симулированных коэффициентов;
- уровень квантили при выборе лучших фондов по коэффициентам на

реальных данных относительно лучших симулированных коэффициентов.

В начале в функции происходит подключение библиотеки для перемешивания остатков линейной регрессии на реальных данных, имеющих временной формат данных. Также создаётся основная таблица размером $n \times (k + 1)$ для значений t -статистик коэффициентов альфа паевых фондов. По умолчанию у функции задано количество бутстрапов как 100, период оценки коэффициентов в 24 месяца, уровень квантили для лучших симулированных коэффициентов величиной 99%, уровень квантили при выборе лучших фондов равен 50%.

```
boot.all = function(return, bench, rf, boot.n = 100, period =
24, l1 = 0.99, l2 = 0.5){
  # библиотека для перемешивания значений массива
  require(tseries)
  # основная таблица t-статистик
  boot=matrix(NA,nrow = ncol(return),ncol = (boot.n + 1))
  # название строк соответствует названия фондов
  row.names(boot)=colnames(return)
```

Рис. 54. Начало функции бутстрапа, создание таблицы для t -статистик

Далее идёт цикл по каждому фонду, где в переменную считываются все данные по одному фонду из определённого столбца таблицы. Пустые значения в переменной с данными фонда удаляются. Если количество данных, в данном случае количество месяцев, по фонду больше, чем минимальное значение, заданное ранее, тогда цикл продолжается и находится пересечение дат по фонду, бенчмарку и безрисковой ставке для правильного сопоставления временного интервала данных.

```
# цикл по каждому фонду
for(j in 1:ncol(return)){
  # данные по одному фонду
  pif=return[,j]
  # удаление пустых значений
  pif=pif[!is.na(pif)]
  # если достаточное количество данных
  if(length(na.trim(pif))>period){
    # пересечение данных по фонду, бенчмарку и безрисковой
    ставке по времени
    time=intersect(intersect(index(pif),index(rf)),index(bench))
```

Рис. 55. Начало основного цикла функции бутстрапа

Если общий временной интервал данных удовлетворяет минимальному условию длины данных, тогда на основе общих дат вычисляется избыточная доходность паевого фонда и избыточная доходность бенчмарка. После чего, если длины избыточных доходностей совпадают, то можно приступить к регрессии на реальных данных. В переменную сарт записывается описание результата данной регрессии, где указаны значения всех необходимых коэффициентов. Соответствующая t-статистики коэффициента альфа записывается в общую таблицу, на основе бета коэффициента и избыточной доходности бенчмарка восстанавливаются значения избыточной доходности паевого фонда.

На следующем шаге производится бутстрап, где к восстановленным данным прибавляются перемешанные ошибки. Бутстрап повторяется в цикле заданное количество раз, где при каждой итерации перемешиваются ошибки из регрессии на реальных данных. После чего, строится линейная регрессия

восстановленной избыточной доходности фонда на восстановленную избыточную доходность бенчмарка. В результате данной регрессии значение t-статистики коэффициента альфа сохраняется в общую таблицу под соответствующим номер фонда и соответствующим номером итерации цикла бутстрапа. На этом получение значений коэффициентов завершается.

```
# если достаточно общего количества данных
if(length(time)>period){
  # избыточная доходность фонда
  Er=Return.excess(pif[as.Date(time)],rf[as.Date(time)])
  # избыточная доходность бенчмарка

Eb=Return.excess(bench[as.Date(time)],rf[as.Date(time)])
  # если количество данных совпадает
  if(length(Er)==length(Eb)){
    # регрессия на реальных данных
    capm=summary(lm(Er ~ Eb))
    # альфа коэффициент
    capm.alfa=capm$coefficients[1,1]
    # t-статистика альфа коэффициента
    capm.t=capm$coefficients[1,3]
    # бета коэффициент
    capm.beta=capm$coefficients[2,1]
    # остатки, ошибки регрессии
    capm.res=capm$residuals
    # сохранение реального значения t-статистики
    boot[j,1]=capm.t
    # восстановление избыточной доходности
    const = capm.beta * Eb
# бутстрап
    # цикл регрессий по восстановленным данным
    for(k in 1:boot.n){
      # перемешивание остатков
      res=xts(tsbootstrap(ts(capm.res)), order.by =
index(capm.res))
      # добавление остатков к восстановленным данным
      Er.b = const + res
      # регрессия на восстановленных данных
      capm.b = summary(lm(Er.b ~ const))
      # сохранение симулированной t-статистики
      boot[j,(1+k)]=capm.b$coefficients[1,3]
    }
  }
}
}
```

Рис. 56. Оценка регрессии на реальных и восстановленных данных

Далее создаётся массив, и в цикле по каждому столбцу в таблице, где столбец соответствует одной итерации бутстрапа, кроме первого, выбираются значения коэффициентов, которые имеют положительные значения и больше заданного уровня квантили распределения значений из данного столбца. Уровень квантили по умолчанию задан на уровне 99%. И выбранные значения объединяются в один массив.

```
# массив лучших симулированных коэффициентов
t.best={}
# цикл по столбцам бутстрапа
for (l in 2:boot.n){
  # отбор и добавление лучших коэффициентов из столбца

  t.best=c(t.best,boot[which(boot[,l]>=quantile(na.omit(boot[,l]),
,c(11)) & boot[,l]>0),l])
}
```

Рис. 57. Выбор лучших коэффициентов симулированных t-статистик по столбцам

На последнем шаге остаётся выбрать лучшие паевые фонды. В цикле по каждому коэффициенту фонда в первом столбце по результату регрессии на реальных данных происходит сравнение данного значения с массивом лучших коэффициентов, полученных выше. В данной работе задан уровень квантили 50% для лучших коэффициентов. Если значение t-статистики коэффициента альфа у фонда выше, чем 50% выборки лучших значений, тогда данный фонд отмечается как лучший, по сравнению с другими, и данный фонд добавляется в отдельный массив вместе с наименованием и значением t-статистики.

```

# цикл по первому столбцу таблицы
# с коэффициентами на реальных данных
for(l in 1:nrow(boot)){
  # если есть значение коэффициента у фонда
  if(!is.na(boot[l,1])){
    # если значение коэффициента выше уровня квантили массива
    лучших коэффициентов
    if(boot[l,1]>quantile(t.best,c(12))){
      # добавление значения коэффициента фонда в массив
      pif.best=c(pif.best,boot[l,1])
      # добавления названия фонда к коэффициенту
      names(pif.best)[length(pif.best)]=row.names(boot)[l]
    }
  }
}

```

Рис. 58. Выбор лучших фондов по значению t-статистик на реальных данных

В результате данной функции возвращается массив лучших паевых инвестиционных фондов, отсортированных в порядке убывания значений t-статистик коэффициентов альфа. Данная функция оценивает линейную регрессию на всей доступной выборке данных по выбранному паевому фонду.

```

# сортировка массива лучших фондов
# по убыванию значений коэффициентов
pif.best=sort(pif.best, decreasing = TRUE)
# возврат в качестве результата
# массив лучших фондов
return(pif.best)
}

```

Рис. 59. Возврат массива лучших фондов в качестве результата функции

Для вычисления лучших фондов и получения результата нужно вызвать функцию с необходимыми параметрами. Для получения бенчмарка на основе индексных фондов вычисляется историческая стоимость портфеля, затем доходность данного портфеля. После чего происходит вызов ранее описанной функции и в переменную `best.pif` возвращается массив лучших фондов. Затем из общей таблицы характеристик фондов по внутреннему номеру фонда выбирается наименование данного фонда.

```
# бенчмарк портфель из индексных паевых фондов
benchmark.pif=portfel.equity(pif.price, pif.act, 1000000,
с(0.005, 0.015))
# доходность бенчмарка
benchmark.pif.return=pif.graph(benchmark.pif, moex)
# функция вычисления лучших фондов
best.pif=boot.all(return = pif.return, bench =
benchmark.pif.return, rf = ru.ly, boot.n = 100, period = 24, l1
= 0.99, l2 = 0.5)
# результат
print(paste(all[all[,3] %in%
names(best.pif),1],round(best.pif,2)))
```

Рис. 60. Бутстрап подход на основе индексного портфеля

Далее печатается в консоль список лучших фондов и их значение t-статистик коэффициентов альфа.

```
> cat(paste(all[all[,3] %in%
names(best.pif),1],round(best.pif,2)),sep = "\n")
Апрель Капитал – Акции 3.36
Апрель Капитал – Акции сырьевых компаний 3.35
Арсагера – фонд акций 3.23
ВТБ – Фонд Акции 3.19
Регион Фонд Акции 2.9
```

Рис. 61. Список лучших фондов

Ниже представлена таблица с лучшими фондами с использованием бутстрап подхода, где в качестве бенчмарка выступает портфель, состоящий из индексных паевых инвестиционных фондов со стратегией следования Индексу МосБиржи.

Таблица 4

Лучшие фонды на основе индексного портфеля

Наименование фонда	t-стат.
Апрель Капитал – Акции	3,36
Апрель Капитал – Акции сырьевых компаний	3,35
Арсагера – Фонд Акции	3,23
ВТБ – Фонд Акции	3,19
Регион Фонд Акции	2,9

Далее вычисляются лучшие фонды на основе портфеля, состоящего из акций, которые входили в Индекс ММВБ 10. Данный портфель восстановлен по исторической базе расчёта данного индекса.

```
# вызов функции с параметрами
benchmark.stock=portfel.equity(data.10, data.act, 1000000,
с(0.0001, 0.0001))
# месячная доходность бенчмарка
benchmark.stock.return=monthlyReturn(Cl(to.monthly(benchmark.st
ock, indexAt = 'lastof'))))
# функция вычисления лучших фондов
best.stock=boot.all(return = pif.return, bench =
benchmark.stock.return, rf = ru.ly, boot.n = 100, period = 24,
l1 = 0.99, l2 = 0.5)
# результат
cat(paste(all[all[,3] %in%
names(best.stock),1],round(best.stock,2)),sep = "\n")
```

Рис. 62. Бутстрап подход на основе портфеля акций Индекса ММВБ 10

Вызывая функцию бутстрапа, где в качестве бенчмарка портфель на основе Индекса ММВБ 10, возвращается следующий результат:

```
> cat(paste(all[all[,3] %in%
names(best.stock),1],round(best.stock,2)),sep = "\n")
РСХБ – Фонд Акции 3.74
РСХБ – Лучшие отрасли 2.97
```

Рис. 63. Список лучших фондов

Ниже представлен список лучших фондов на основе бенчмарка, который следует Индексу ММВБ 10.

Таблица 5

Лучшие фонды на основе портфеля акций Индекса ММВБ 10

Наименование фонда	t-стат.
РСХБ – Фонд Акции	3,47
РСХБ — Лучшие отрасли	2,97

Далее вычисляются лучшие паевые инвестиционные фонды, где в качестве бенчмарка выступает портфель, основанный на инвестициях во фьючерсы с ближайшей датой исполнения на Индекс МосБиржи.


```
# вызов функции
benchmark.future=future.benchmark()
# месячная доходность бенчмарка
benchmark.future.return=monthlyReturn(Cl(to.monthly(benchmark.f
uture, indexAt = 'lastof'))))
# функция вычисления лучших фондов
best.future=boot.all(return = pif.return, bench =
benchmark.future.return, rf = ru.ly, boot.n = 100, period = 24,
l1 = 0.99, l2 = 0.5)
# результат
cat(paste(all[all[,3] %in%
names(best.future),1],round(best.future,2)),sep = "\n")
```

Рис. 64. Бутстрап подход на основе портфеля фьючерсов на Индекс МосБиржи

Функция возвращает следующий результат:

```
> cat(paste(all[all[,3] %in%
names(best.future),1],round(best.future,2)),sep = "\n")
Апрель Капитал – Акции 3.18
Апрель Капитал – Акции сырьевых компаний 2.9
РСХБ – Фонд Акции 2.85
```

Рис. 65. Список лучших фондов

Ниже представлены лучшие паевые фонды по результату бутстрап подхода, где в качестве бенчмарка используется портфель, стратегия которого следует Индексу МосБиржи, где в качестве инструментов используются фьючерсы на данный индекс с ближайшей датой исполнения.

Таблица 6

Лучшие фонды на основе портфеля фьючерсов на Индекс МосБиржи

Наименование фонда	t-стат.
Апрель Капитал – Акции	3,18
Апрель Капитал – Акции сырьевых компаний	2,9
РСХБ – Фонд Акции	2,85

В результате бутстрап подхода и на основе трёх бенчмарков, которые были построены из инструментов, которые доступны инвестору, и с использованием которых можно воспроизвести Индекс МосБиржи или Индекс ММВБ 10, были определены лучшие фонды. Для воссоздания Индекса МосБиржи было создано два портфеля, один из которых строился из паёв индексных паевых инвестиционных фондов, стратегия которых следовать

данному индексу. Второй портфель строился на основе фьючерсов с ближайшей датой исполнения на данный индекс, которые доступны инвестору на Московской Бирже. Другой портфель воспроизводил Индекс ММВБ 10, поскольку по данному индексу доступна полная историческая база расчёта, а также историческая цена акций на Московской Бирже, которые когда-либо составляли данный индекс.

Список лучших фондов:

- Апрель Капитал – Акции;
- Апрель Капитал – Акции сырьевых компаний;
- Арсагера – Фонд Акции;
- ВТБ – Фонд Акции;
- Регион – Фонд Акции;
- РСХБ – Фонд Акции;
- РСХБ – Лучшие отрасли.

Данные фонды являются лучшими среди всей выборки в 57 открытых не отраслевых паевых инвестиционных фондов акций. Также данные фонды превосходят по доходности не только построенные портфели, которые выступают в качестве бенчмарка, но и остальные паевые фонды в выборке.

3.4. Инвестиционная стратегия на основе бутстрап подхода

Данная стратегия похожа на стратегию, которая применялась при выборе индексного паевого инвестиционного фонда, где критерием выбора было максимально полное следование паевого фонда Индексу МосБиржи. В данном случае критерием выбора паевого инвестиционного фонда акций будет результат бутстрапа.

В стратегии используется движущийся период в 24 месяца. На данном периоде применяется бутстрап метод для определения лучших фондов за период. После чего лучшие фонды за данный период отмечаются в общей таблице всех фондов. Таким образом накапливается статистика о том, какие фонды становились лучшими среди других на разных периодах. Далее данная

полученная таблица оценивается с плавающим периодом в четыре месяца, и выбирается один лучший фонд по сумме данных четырёх месяцев, в который будут инвестироваться средства на следующие четыре месяца. Подобная стратегия применялась при создании портфеля на основе индексных паевых инвестиционных фондов, где период инвестирования составлял двенадцать месяцев. По прошествии четырех месяцев по стратегии данного портфеля, снова принимается решение об инвестировании в паевой фонд по результатам бутстрапа на прошедшем периоде в четыре месяца.

Для данной стратегии функция бутстрапа была модифицирована, при этом входные параметры остаются прежними, но переменная периода изменяет роль с длины минимального периода на роль длины периода для движущегося окна по всей протяженности данных.

В начале функции создаётся таблица, которая по структуре копирует таблицу доходностей паевых инвестиционных фондов. В данной таблице будут отмечаться лучшие фонды по результату бутстрапа за определенный период.

```
boot.history = function(return, bench, rf, boot.n = 10, period
= 24, l1 = 0.99, l2 = 0.5){
  # библиотека для перемешивания значений массива
  require(tseries)
  # основная таблица с результатами бутстрапа
  # структура копируется с таблицы доходностей активов
  boot.res=pif.return
  # очистка данных
  boot.res[]=NA
```

Рис. 66. Начало функции, создание общей таблицы

Далее начинается основной цикл с движущимся окном в 24 месяца по всей длине данных. В отдельную переменную `dta.m` записываются данные по доходности для всех фондов за данный период. Информация из данной таблицы будет участвовать в бутстрапе. Также создаётся таблица с оценками t-статистик коэффициентов альфа по фондам. В функции бутстрап подхода по всей длине данных создаётся только одна таблица на весь период, в данной версии такая таблица создаётся на каждом периоде.

```
# цикл с движущимся окном
for(i in 1:(nrow(return)-(period-1))){
  # данные по доходности активов за период
  dta.m=return[i:(i+(period-1)),]
  # таблица t-статистик
  boot=matrix(NA,nrow = ncol(dta.m),ncol = (boot.n + 1))
  # название строк соответствует названию активов
  row.names(boot)=colnames(dta.m)
```

Рис. 67. Основной цикл с движущимся периодом, таблицы t-статистик

Далее производятся действия, как в функции бутстрапа, описанной ранее. Так же происходит цикл по каждому фонду, где проверяется, что количество данных по фонду равняется длине периода в 24 месяца. Также данную проверку проходит длина общего временного периода для фонда, бенчмарка и безрисковой ставки для нахождения избыточной доходности фонда и избыточной доходности бенчмарка.

```
# цикл по каждому активу
for(j in 1:ncol(dta.m)){
  # данные по активу за период
  pif=dta.m[,j]
  # удаление пустых значений
  pif=pif[!is.na(pif)]
  # если достаточное количество данных
  if(length(na.trim(pif))==period){
    # пересечение данных по фонду, бенчмарку и безрисковой
    ставке по времени

time=intersect(intersect(index(pif),index(rf)),index(bench))
    # если достаточно общего количества данных
    if(length(time)==period){
      # избыточная доходность фонда

Er=Return.excess(pif[as.Date(time)],rf[as.Date(time)])
      # избыточная доходность бенчмарка

Eb=Return.excess(bench[as.Date(time)],rf[as.Date(time)])
```

Рис. 68. Цикл по каждому фонду, вычисление избыточной доходности

После того, как проверки пройдены, производится линейная регрессия на реальных данных и значение t-статистики коэффициента альфа записывается в таблицу. Далее происходит восстановление избыточной доходности бенчмарка.

Ошибки из регрессии сохраняются в отдельную переменную для дальнейшего перемешивания.

```
# если количество данных совпадает
if(length(Er)==length(Eb)){
  # регрессия на реальных данных
  capm=summary(lm(Er ~ Eb))
  # альфа коэффициент
  capm.alfa=capm$coefficients[1,1]
  # t-статистика альфа коэффициента
  capm.t=capm$coefficients[1,3]
  # бета коэффициент
  capm.beta=capm$coefficients[2,1]
  # остатки, ошибки регрессии
  capm.res=capm$residuals
  # сохранение реального значения t-статистики
  boot[j,1]=capm.t
  # восстановление избыточной доходности
  const = capm.beta * Eb
```

Рис. 69. Вычисление линейной регрессии на реальных данных

Далее производится бутстрап, где на каждом шаге перемешиваются ошибки регрессии на реальных данных, и производится восстановление избыточной доходности фонда без участия коэффициента альфа. Далее оценивается t-статистика на восстановленных данных и записывается в таблицу.

```
# бутстрап
# цикл регрессий по восстановленным данным
for(k in 1:boot.n){
  # перемешивание остатков
  res=xts(tsbootstrap(ts(capm.res)), order.by =
index(capm.res))
  # добавление остатков к восстановленным данным
  Er.b = const + res
  # регрессия на восстановленных данных
  capm.b = summary(lm(Er.b ~ const))
  # сохранение симулированной t-статистики
  boot[j, (1+k)]=capm.b$coefficients[1,3]
}
```

Рис. 70. Бутстрап цикл на восстановленных данных

На завершающем этапе заполняется массив лучших t-статистик коэффициентов альфа на симулированных данных за данный период в 24

месяца, путём выбора из каждого столбца с результатом бутстрапа значений, которые выше заданной квантили распределения уровня 99%.

На последнем этапе выбираются лучшие паевые фонды на периоде в 24 месяца, через сравнение значений t-статистик на реальных данных из первого столбца с массивом лучших симулированных коэффициентов отобранных ранее. Если значение оценки для фонда выше уровня квантили в 50% для данного массива, тогда фонд отмечается как лучший среди остальных на данном периоде.

```
# массив лучших симулированных коэффициентов
t.best={}
# цикл по столбцам бутстрапа
for (l in 2:boot.n){
  # отбор и добавление лучших коэффициентов из столбца

t.best=c(t.best,boot[which(boot[,l]>=quantile(na.omit(boot[,l]),
,c(11)) & boot[,l]>0),l])
}

# массив лучших фондов
pif.best={}
# цикл по первому столбцу таблицы
# с коэффициентами на реальных данных
for(l in 1:nrow(boot)){
  # если есть значение коэффициента у фонда
  if(!is.na(boot[l,1])){
    # если значение коэффициента выше уровня квантили
массива лучших коэффициентов
    if(boot[l,1]>quantile(t.best,c(12))){
      # добавление значения коэффициента фонда в массив
      pif.best=c(pif.best,boot[l,1])
      # добавления названия фонда к коэффициенту
      names(pif.best)[length(pif.best)]=row.names(boot)[l]
    }
  }
}
```

Рис. 71. Выбор лучших фондов на определённом периоде

Данные о лучших фондах за период заносятся в таблицу, созданную в самом начале. В данной таблице для выбранных фондов указывается вес, который равен порядковому номеру фонда в массиве значений t-статистик, отсортированного в порядке убывания. Лучший фонд с высоким значением,

который находится на первом месте в массиве лучших фондов, в таблице будет отмечен с весом в единицу, второй фонд с весом два и так далее.

```
# сортировка массива лучших фондов
# по убыванию значений коэффициентов
pif.best=sort(pif.best, decreasing = TRUE)
# лучшие активы за данный период отмечаются в общей таблице

boot.res[tail(index(dta.m),n=1),names(pif.best)]=c(1:length(pif
.best))
```

Рис. 72. Запись весов лучших фондов в таблицу

В конце функции, когда бутстрап с движущимся периодом сделан по всей длине данных, возвращается результат в виде таблицы, созданной в самом начале. Но перед этим из данной таблицы удаляются фонды, которые ни в одном периоде не входили в список лучших и не были отмечены. Поиск таких фондов происходит путём суммирования данных из данной таблицы по столбцам. Те фонды, сумма столбцов которых отлична от нуля, остаются в таблице.

```
# массив сумм по столбцам
sums=colSums(boot.res,na.rm = TRUE)
# удаляются те фонды
# которые ни разу не были отмечены
sums=sums[sums>0]
# удаление данных фондов из таблицы
boot.res=boot.res[,names(sums)]
# возвращение таблицы в качестве результата функции
return(boot.res)
```

Рис. 73. Выбор фондов, по которым есть данные

Для составления таблицы с индикаторами, которые указывают на то, какие паевые фонды составляют инвестиционный портфель в определенный момент, используется алгоритм, который применялся при составлении портфеля на основе индексных паевых инвестиционных фондов, где на протяжении 12 месяцев выбирался индексный фонд, наиболее близкий по доходности к Индексу МосБиржи, и в данный фонд инвестировался портфель на следующие 12 месяцев.

Данная функция принимает в качестве параметров таблицу с данными по бутстрапу о лучших фондах на определенный период и длину периода, на котором будут сравниваться фонды. Инвестиции также будут происходить на значение длины заданного периода. В процессе сравнения выбирается один лучший фонд на определенном периоде. В данный фонд происходят инвестиции на следующий период в четыре месяца по умолчанию.


```

boot.act=function(boot.res, period = 4){
  # создание копии таблицы
  copy=boot.res
  # стирание данных
  copy[]=NA
  # индикатор для начала
  # вычисления фонда в который
  # нужно инвестировать
  j=period - 1
  for(i in 1:(nrow(boot.res)-(period-1))){
    # выделение данных за период в переменную
    dta.m=boot.res[i:(i+(period-1)),]
    # проверка, достаточно ли данных в периоде
    if(length(sort(colSums(dta.m)))>0){
      # увеличение периода на 1 месяц
      j=j+1
      # если количество месяцев кратно периоду
      # то нужно вычислять новый фонд для инвестиций
      # на следующий период
      if(j%%period==0){
        # вычисляется сумма по каждому фонду
        # затем данные записываются в массив
        best=names(sort(colSums(dta.m)))
        n=as.numeric(floor(quantile(c(0:length(best)),.25)))
        # вычисление месяца начала инвестиций
        k=i+period
        # вычисление конечного месяца инвестиции
        k1=ifelse((k+(period-
1))>nrow(copy),nrow(copy),(k+(period-1)))
        # запись в таблицу информации о том,
        # в какой фонд будут сделаны инвестиции
        copy[index(copy[k:k1,]),best[1]]=1
      }
    }
  }
  # массив сумм по столбцам
  sums=colSums(copy,na.rm = TRUE)
  # удаляюся те фонды
  # которые ни разу не были отмечены
  sums=sums[sums>0]
  # удаление данных фондов из таблицы
  copy=copy[,names(sums)]
  # возвращение таблицы в качестве результата функции
  return(copy)
}

```

Рис. 74. Функция составления таблицы с индикаторами инвестирования

В конце в качестве результата функция возвращает таблицу с индикаторами, которые обозначают фонды, в которые нужно инвестировать на

определённом периоде. Из таблицы удалены фонды, которые ни разу не отмечены для инвестиций. С использованием данной таблицы и таблицы с ценами по данным паевым инвестиционным фондам можно построить портфель.

Функция вычисления лучших фондов по периодам выполняется продолжительное количество времени, поскольку используется движущееся окно и на каждом периоде задано сто повторений бутстрапа. Далее по функции описанной выше ставятся индикаторы для фондов, по которым будет строиться портфель. Далее по функции строится портфель, где в качестве параметров передаются историческая стоимость паевых фондов, таблица с индикаторами, количество денежных средств для инвестирования и комиссии на покупку и продажу паев.

```
# функция вычисления лучших фондов по периодам
best.pif.boot=boot.history(return = pif.return, bench =
benchmark.pif.return, rf = ru.ly, boot.n = 100, period = 24, l1
= 0.99, l2 = 0.5)
# функция для установки индикаторов
best.pif.act=boot.act(best.pif.boot,4)
# построение портфеля
portfel=portfel.equity(pif.price, best.pif.act, 10000000,
c(0.005, 0.015))
# доходность портфеля
portfel.return=pif.graph(prtf,moex)
```

Рис. 75. Построение портфеля по стратегии с использованием индексного портфеля

В конце вычисляется доходность портфеля и его сравнение с бенчмарком, в качестве которого выступает портфель индексных фондов по Индексу МосБиржи. Также для оценки портфеля относительно бенчмарка вычисляется регрессия по модели CAPM. По результатам сравнения коэффициент альфа портфеля созданного по данной стратегии имеет коэффициент значимости на уровне 1%, а само значение альфа близко к 1%. Коэффициент бета также имеет высокий уровень значимости, и значение бета равно 0,75. Графическое сравнение портфеля и бенчмарка демонстрирует, что портфель обыгрывает бенчмарк, который соответствует Индексу МосБиржи.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.009386	0.003419	2.745	0.00778	**
test[, 1]	0.750155	0.075081	9.991	7.82e-15	***

Рис. 76. Результат регрессии, коэффициенты альфа и бета

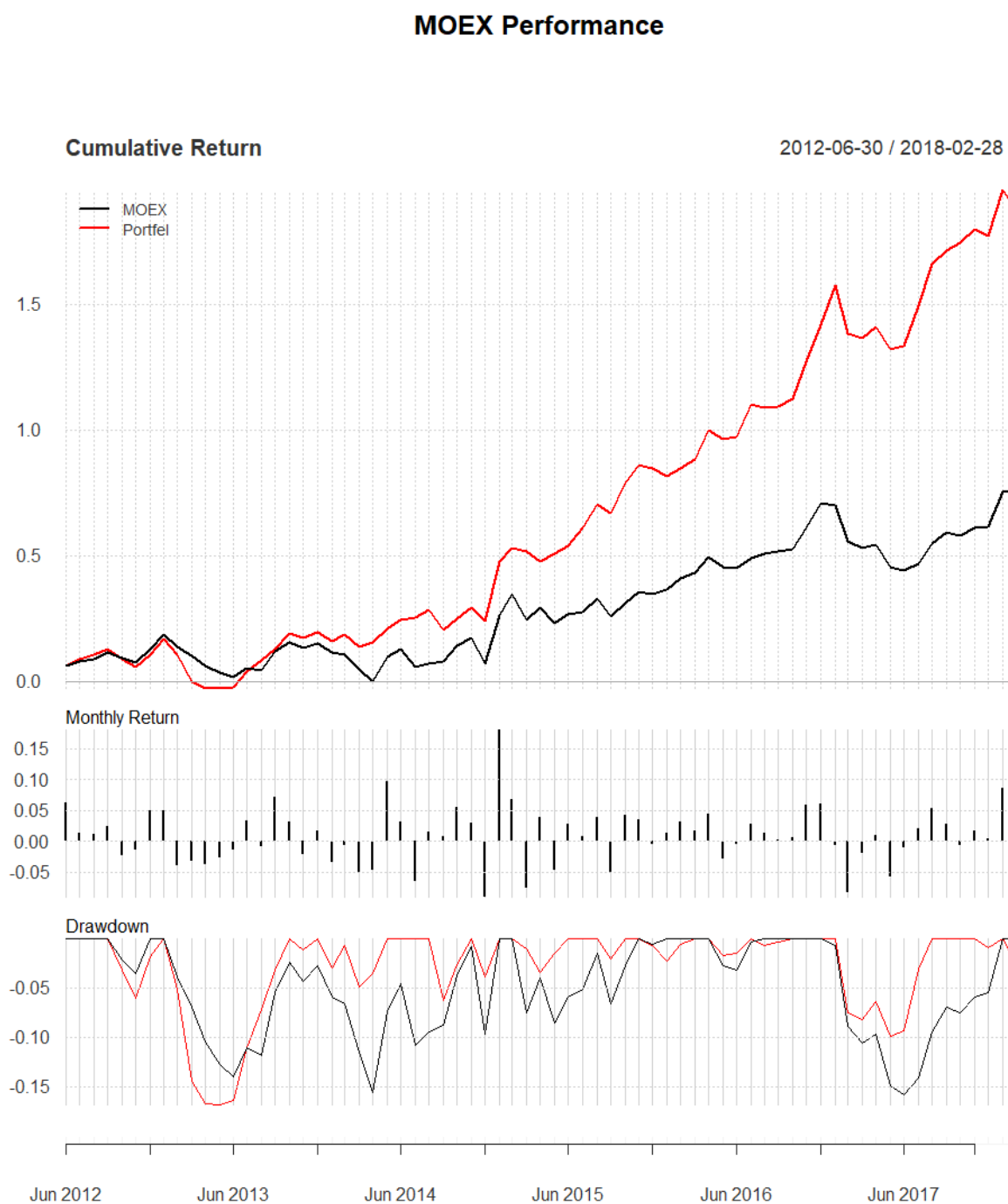


Рис. 77. Графики доходности стратегии и индексного портфеля

Также данную стратегию можно применить на основе бенчмарка, в основе которого портфель, построенный из акций, когда-либо составляющих базу расчёта Индекса ММВБ 10.

```
# функция вычисления лучших фондов по периодам
best.pif.boot=boot.history(return = pif.return, bench =
benchmark.stock.return, rf = ru.ly, boot.n = 100, period = 24,
l1 = 0.99, l2 = 0.5)
# функция для установки индикаторов
best.pif.act=boot.act(best.pif.boot, l2)
# построение портфеля
portfel=portfel.equity(pif.price, best.pif.act, 10000000,
c(0.005, 0.015))
# доходность портфеля
portfel.return=pif.graph(portfel, moex10)
```

Рис. 78. Построение портфеля по стратегии

Наилучший результат был достигнут при периоде инвестирования в определенный фонд на период в 12 месяцев, при использовании данного портфеля, построенного на основе Индекса ММВБ 10, для выбора наилучших паевых инвестиционных фондов. Альфа коэффициент построенного портфеля по данной стратегии равняется 0,7% и p-value на уровне значимости 5%. Бета коэффициент данного портфеля равен 0,93, что показывает высокую чувствительность доходности портфеля к бенчмарку, и данный коэффициент также имеет высокий уровень значимости.

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.006848	0.002909	2.354	0.0218 *
test[, 1]	0.930902	0.059131	15.743	<2e-16 ***

Рис. 79. Результат регрессии, коэффициенты альфа и бета

Также на графике построенный портфель на основе данной стратегии обыгрывает портфель, который основан на Индексе ММВБ 10.

MOEX10 Performance

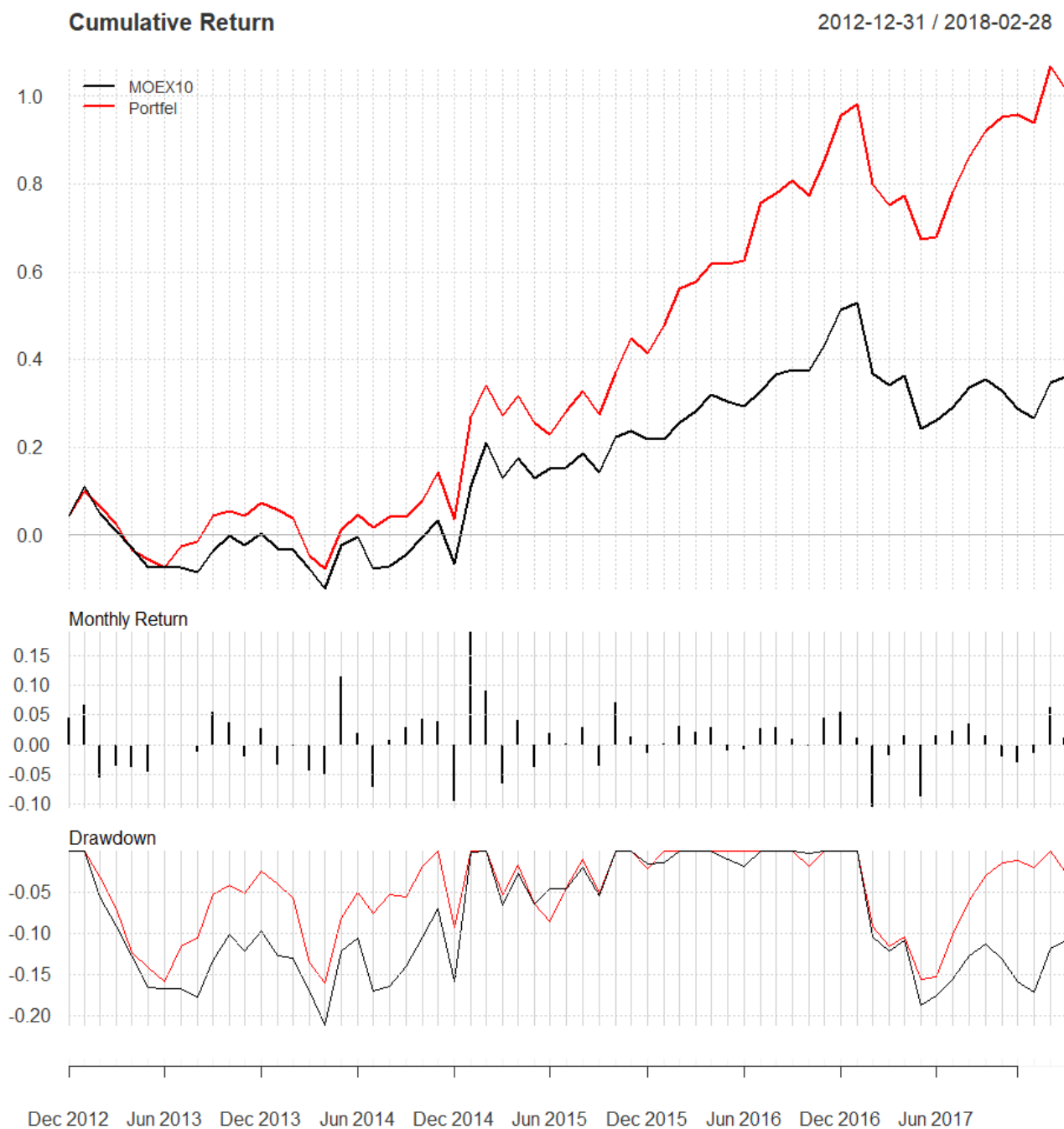


Рис. 80. Графики доходности стратегии и портфеля акций Индекса ММВБ 10

Далее на основе данной стратегии можно построить портфель на основе бенчмарка, в качестве которого выступает портфель, состоящий из фьючерсов на Индекс МосБиржи. Период инвестирования в лучший фонд также составляет четыре месяца.

```

# функция вычисления лучших фондов по периодам
best.pif.boot=boot.history(return = pif.return, bench =
benchmark.future.return, rf = ru.ly, boot.n = 100, period = 24,
l1 = 0.99, l2 = 0.5)
# функция для установки индикаторов
best.pif.act=boot.act(best.pif.boot,4)
# построение портфеля
portfel=portfel.equity(pif.price, best.pif.act, 10000000,
с(0.005, 0.015))
# доходность портфеля
portfel.return=pif.graph(portfel,моех)

```

Рис. 81. Построение портфеля по стратегии

В результате сравнивая построенный по стратегии портфель с бенчмарком были получены коэффициенты по модели CAPM через вычисление линейной регрессии. Коэффициент альфа для портфеля составил 0,7% при уровне значимости 10%. Коэффициент бета равняется 0,82 также при высоком уровне значимости, как и в предыдущих портфелях, созданных по данной стратегии.

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.007884	0.004205	1.875	0.0671 .
test[, 1]	0.819534	0.084773	9.667	9.43e-13 ***

Рис. 82. Результат регрессии, коэффициенты альфа и бета

Ниже приведено графическое сравнение доходностей портфеля, построенного по данной стратегии, и портфеля, который создан с использованием фьючерсов с ближайшей датой исполнения на Индекс МосБиржи. Данный график показывает, что данный портфель обыгрывает бенчмарк на основе фьючерса на Индекс МосБиржи.

MOEX Performance

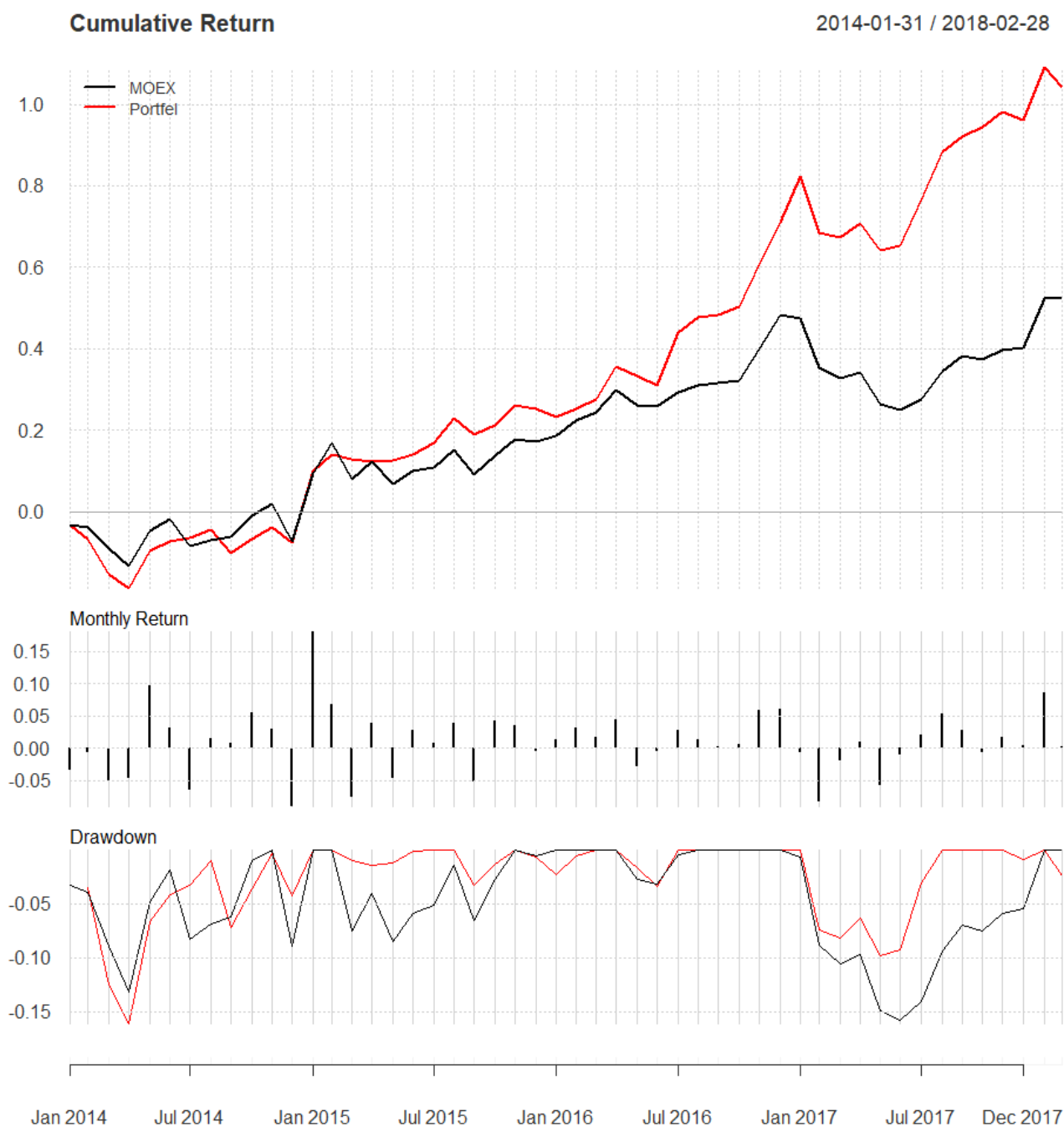


Рис. 83. Графики доходности стратегии и портфеля фьючерсов на Индекс МосБиржи

В данной главе была описана бутстрап модель, при использовании которой можно произвести оценку паевых инвестиционных фондов как между собой, так и относительно бенчмарка. Также разработан бутстрап подход, который реализует данную модель, в результате чего были оценены российские

паевые инвестиционные фонды акций и выбраны лучшие из них. Был реализован алгоритм на языке программирования R, с использованием которого был применён данный бутстрап подход на реальных данных.

При оценке фондов в модели было задействовано три бенчмарка. Данные бенчмарки были построены из инструментов, которые доступны инвесторам на финансовом рынке. Первый бенчмарк создан на основе инвестиций в индексные паевые инвестиционные фонды, стратегия которых следовать значению Индекса МосБиржи. Второй бенчмарк был воспроизведён по архивной базе расчёта Индекса ММВБ 10, которая доступна на сайте Московской Биржи. В основу данного портфеля вошли акции, которые в определённый момент составляли Индекс ММВБ 10 и торговались на фондовом рынке. Последний бенчмарк был построен на основе фьючерсов с ближайшей датой исполнения на Индекс МосБиржи, данные по которым доступны на сайте Московской Биржи, включая размер гарантийного обеспечения и значения расчётной цены на каждый торговый день.

В конце главы на основе данного подхода была предложена стратегия с движущимся периодом, на котором воспроизводился бутстрап подход и в результате чего отмечались лучшие фонды за определённый период. Далее на основе полученного результата на определённом периоде производился выбор одного фонда, в который будут инвестироваться средства портфеля на такой же период. По прошествии данного периода снова производился выбор фонда на следующий период.

Данная стратегия продемонстрировала хорошие показатели на основе коэффициентов альфа и бета из модели CAPM через вычисление линейной регрессии. Построенный портфель на основе стратегии с использованием бенчмарка, созданного из индексных фондов со стратегией следования Индексу МосБиржи, продемонстрировал наиболее высокие показатели доходности.

Заключение

В данной работе был продемонстрирован полный этап прохождения от получения и обработки данных, описания, построения и реализации модели, заканчивая получением результата на реальных данных и разработкой инвестиционной стратегии.

В первой главе описаны способы получения данных по паевым инвестиционным фондам акций, индексным инвестиционным фондам, кривой бескупонной доходности однолетних государственных облигаций Российской Федерации, исторической стоимости акций на Московской Бирже, архивной базе расчёта Индекса ММВБ 10 и данных по фьючерсам с ближайшей датой исполнения на Индекс МосБиржи, а также исторические значения Индекса МосБиржи и Индекса ММВБ 10. Далее в данной главе описываются методы обработки полученных данных для дальнейшей реализации бутстрап модели на их основе.

Во второй главе описаны методы построения бенчмарков из инструментов, которые доступны инвестору. Первый бенчмарк основан на инвестиции в портфель индексных паевых фондов, где фонды выбираются по критерию близости значений к Индексу МосБиржи на определённом периоде. Второй бенчмарк создавался по архивной базе расчёта Индекса ММВБ 10. И последний бенчмарк строился из фьючерсов с ближайшей датой исполнения на Индекс МосБиржи.

В третьей главе описана методика бутстрап модели, разработан подход по её реализации, и создан программный модуль, который воспроизводит данную модель. В результате применения данного метода были определены лучшие фонды, которые получили избыточную безрисковую доходность не только относительно бенчмарка, но и по отношению к другим паевым инвестиционным фондам. В конце данной главы на основе бутстрап подхода

была предложена стратегия, которая по результатам тестов обыгрывает бенчмарк и показывает коэффициент альфа на уровне значимости 1%.

Среди построенных портфелей на основе Индекса МосБиржи и Индекса ММВБ 10, наиболее подходящим для бенчмарка является портфель из индексных паевых инвестиционных фондов. Поскольку при создании данного портфеля применялась стратегия, которая выбирает фонд для инвестиций по критерию близости значений фонда к Индексу МосБиржи. Второй портфель из акций Индекса ММВБ 10 обладает недостатком, что данный индекс в меньшей степени характеризует российский фондовый рынок, чем Индекс МосБиржи. Но по Индексу ММВБ 10 есть полная архивная база расчёта, которую предоставляет Московская Биржа, и в данной работе продемонстрирован подход создания портфеля по данному индексу. Портфель из фьючерсов на Индекс МосБиржи также не в полной мере повторяет Индекс МосБиржи, что не всегда возможно, поскольку фьючерс может быть в состоянии контанго или бэквордация, что является следствием того, что цена по данному инструменту это ожидаемая участниками фондового рынка стоимость данного базового актива в будущем.

В дальнейшем данная работа может стать базисом для тестирования других методов оценки паевых инвестиционных фондов на собранных данных и с использованием реализованных способов получения актуальных данных. Также продолжением данного исследования может стать модификация предложенных методов оценки паевых фондов и улучшение программной реализации. Особо актуальным предметом дальнейшего исследования является продолжение разработки инвестиционной стратегии, предложенной в данной работе. Поскольку характеристика данной стратегии в сравнении с бенчмарком по модели САРМ демонстрируют большой потенциал, так как на данном этапе были получены значения безрисковой избыточной доходности на уровне значимости в 1% для коэффициента альфа. Также остаётся актуальным разработка новых стратегий на основе бутстрап оценки паевых инвестиционных фондов, реализованной в данном исследовании.

Список литературы

1. Муравьев Д.М. (2006), Обыгрывают ли российские паевые фонды рынок? / Препринт # BSP/2006/085 R. – М.: Российская Экономическая Школа, 2006. – 29 с. (Рус.)
2. Паршаков П.А. (2014), Оценка эффективности деятельности по управлению активами российских паевых инвестиционных фондов. – М.: Высшая Школа Экономики, 2014. – 184 с. (Рус.)
3. Центральный банк Российской Федерации (2018), Обзор ключевых показателей паевых и акционерных инвестиционных фондов 2017 № 4. – М.: Центральный банк Российской Федерации, 2018. – 21 с. (Рус.)
4. Консультант Плюс (2018), Федеральный закон от 29.11.2001 N 156-ФЗ (ред. от 03.07.2016) "Об инвестиционных фондах" (с изм. и доп., вступ. в силу с 01.01.2017)
5. Cuthbertson, K., Nitzsche, D. and O'Sullivan, N. (2010), Mutual Fund Performance: Measurement and Evidence. Financial Markets, Institutions & Instruments, 19: 95–187. doi:10.1111/j.1468-0416.2010.00156.x
6. Kosowski, R., Timmermann, A., Wermers, R. And White, H. (2006), Can Mutual Fund “Stars” Really Pick Stocks? New Evidence from a Bootstrap Analysis. The Journal of Finance, 61: 2551–2595. doi:10.1111/j.1540-6261.2006.01015.x
7. Fama, E. F. And French, K. R. (2010), Luck versus Skill in the Cross-Section of Mutual Fund Returns. The Journal of Finance, 65: 1915–1947. doi:10.1111/j.1540-6261.2010.01598.x
8. Cbonds group (2018), Investfunds, available at: <http://investfunds.ru/> (accessed 22 February 2018).
9. Harrison, J. (2018), “wdman: 'Webdriver'/'Selenium' Binary Manager”, available at: <https://CRAN.R-project.org/package=wdman> (accessed 22 February 2018).

10. Harrison, J. (2018), “RSelenium: R Bindings for 'Selenium WebDriver'”, available at: <https://CRAN.R-project.org/package=RSelenium> (accessed 22 February 2018).
11. Ryan, J., Ulrich, J., Thielen, W. (2007), “quantmod: Quantitative Financial Modelling Framework”, available at: <https://CRAN.R-project.org/package=quantmod> (accessed 22 February 2018).
12. Wickham, H., Bryan, J., RStudio, Kalicinski, M., Komarov, V., Leittenne, C., Colbert, B., Hoerl, D. (2018), “readxl: Read Excel Files”, available at: <https://CRAN.R-project.org/package=readxl> (accessed 15 March 2018).
13. Trapletti, A., Hornik, K., LeBaron, B. (2018), “tseries: Time Series Analysis and Computational Finance”, available at: <https://CRAN.R-project.org/package=tseries> (accessed 22 February 2018).
14. Wickham, H. (2018), “rvest: Easily Harvest (Scrape) Web Pages”, available at: <https://CRAN.R-project.org/package=rvest> (accessed 15 March 2018).
15. Ryan, J., Ulrich, J. (2018), “xts: eXtensible Time Series”, available at: <https://CRAN.R-project.org/package=xts> (accessed 15 March 2018).

Приложение

Исходный программный код:

```
require(RSelenium) # парсит сайт
require(wdman) # браузер
require(xts)
require(gdata) # скачивание файла

pjs <- phantomjs() # запуск сервера - браузера
pjs$output()
rem <- remoteDriver(remoteServerAddr = "localhost"
                    , port = 4567L
                    , browserName = "phantomjs"

) # описание подключения для Selenium
rem$open() # подключение к браузеру
rem$getStatus()
links=names=c() # массив ссылок и названий
for (i in 0:1){ # 2 страницы

link=paste0("http://pif.investfunds.ru/funds/index.phtml?page=funds&type[]=%D0%BE%D1%82%D0%BA%D1%
80%D1%8B%D1%82%D1%8B%D0%B9&c_val[2]=23&value=2&npage=",i) # адрес страницы
  rem$navigate(link) # открытие страницы
  rem$getCurrentUrl() # текущий адрес
  elem <- rem$findElements(using = 'xpath', "//tr/td[1]/a[@class='link']") # выбор элементов
  names <- append(names,unlist(lapply(elem, function(x){x$getElementText()}))) # запись названий
  ПИФов
  links <- append(links,unlist(lapply(elem, function(x){x$getElementAttribute("href")}))) #
  запись ссылок ПИФов
}

all=data.frame( # таблица с описанием и общей информацией ПИФов
  Name=as.character(), # название ПИФа
  Link=as.character(), # ссылка на ПИФ
  Number=as.character(), # внутренний номер
  FeeMan=as.numeric(), # вознаграждение менеджера
  FeeDep=as.numeric(), # вознаграждение депозитария
  FeeOth=as.numeric(), # прочие расходы
  stringsAsFactors=FALSE
)

allD=list() # данные по цене и СЧА для ПИФов
dir=getwd()
for(i in 1:length(names)){ # цикл по каждому ПИФу
  all[i,"Name"]<-names[i] # название ПИФа
  all[i,"Link"]<-links[i] # ссылка на ПИФ
  all[i,"Number"]<-regmatches(links[i], regexpr("[0-9]+", links[i], perl=TRUE)) # внутренний
  номер ПИФа на сайте
  rem$navigate(links[i]) # открытие страницы ПИФа

  elem<-rem$findElement(using = 'xpath', "//a[@class='popup-link arr-open-title']") # поиск
  кнопки на сайте для отображения информации
  elem$clickElement() # нажатие кнопки
  elem <- rem$findElements(using = 'xpath', "//td[@style='border-top:0px;text-align:center']") #
  список данных по вознаграждению
  z=unlist(lapply(elem, function(x){x$getElementText()})) # текстовое представление данных по
  вознаграждению
  all[i,c("FeeMan","FeeDep","FeeOth")]=as.numeric(regmatches(z, regexpr("[0-9.]+", z,
  perl=TRUE)))/100 # сохранения в числовом виде

download.file(paste0("http://pif.investfunds.ru/funds/export_to_excel.php?f2[0]="),all[i,"Number"]
```

```

,"&export=2&export_type=xls&start_day&finish_day=08&finish_month=02&finish_year=2018&rnd=5030"),d
estfile="pif.xls",mode="wb",quiet=TRUE) # скачивание файла с данными по цене и СЧА ПИФа
x<-read.xls(xls=paste0(dir,"/pif.xls"),perl="C:/Perl64/bin/perl.exe") # чтение файла в
переменную - таблицу
x<-x[which(x[,1]!="&x[,1]!="00.00.0000"),] # очистка пустых строк и неправильных дат
x<-x[2:length(x[,1]),1:3] # очистка от названий
x[,2]=as.numeric(gsub(" ", "", as.character(x[,2]))) # преобразование цены из текстового в
числовой формат
x[,3]=as.numeric(gsub(" ", "", as.character(x[,3]))) # преобразование СЧА из текстового в
числовой формат
x1<-as.xts(x[,2],order.by=as.Date(x[,1], "%d.%m.%Y")) # преобразование цены в xts формат
x2<-as.xts(x[,3],order.by=as.Date(x[,1], "%d.%m.%Y")) # преобразование СЧА в xts формат
allD[[i]]=list(x1,x2) # сохранение в лист двух xts фреймов
print(rem$getCurrentUrl()) # печать текущего адреса страницы
}
names(allD)=all[, "Number"] # название в листах соответствует внутреннему номеру ПИФов
save(all, file="F:/vkr/pifdata/index/all.RData") # сохранение в файл
save(allD, file="F:/vkr/pifdata/index/allD.RData") # сохранение в файл

require(quantmod)
require(PerformanceAnalytics)
load("F:/allD.RData")
load("F:/all.RData")
Sys.setlocale("LC_TIME", "English")
pif.price=xts()
pif.return=xts()
for(i in 1:length(allD)){
  if(i==1){
    # заполнение первой колонки таблицы
    pif.price=C1(to.monthly(allD[[i]][[1]],indexAt = 'lastof'))
    pif.return=monthlyReturn(to.monthly(allD[[i]][[1]],indexAt = 'lastof'))
  } else {
    # добавление следующих колонок в таблицу
    pif.price=cbind(pif.price,C1(to.monthly(allD[[i]][[1]],indexAt = 'lastof')))
    pif.return=cbind(pif.return,monthlyReturn(to.monthly(allD[[i]][[1]],indexAt = 'lastof')))
  }
}
colnames(pif.price)=names(allD)
colnames(pif.return)=names(allD)

for(i in 1:ncol(pif.return)){
  pr=tail(which(is.na(pif.return[,i])),n=1)
  if(length(pr)>0) pif.return[1:pr,i]=NA
  pp=tail(which(is.na(pif.price[,i])),n=1)
  if(length(pp)>0) pif.price[1:pp,i]=NA
  rm(pr,pp)
}

Sys.setlocale("LC_TIME", "English")
# чтение данных из файла
ru.1y=read.csv("F:/vkr/RU1Y.csv",stringsAsFactors = FALSE)
# преобразование в формат данных xts
ru=xts(as.numeric(ru.1y[(1:(nrow(ru.1y)-2)),2]),order.by = as.Date(ru.1y[(1:(nrow(ru.1y)-2)),1],
format = "%b %d, %Y"))
# перевод доходности за год к доходности за месяц
ru.1y=xts((1 + C1(to.monthly(ru,indexAt = 'lastof'))/100)^(1/12)-1)

# чтение данных из файла
moex=read.csv("F:/vkr/MOEX.csv",stringsAsFactors = FALSE)
# преобразование в формат данных xts
moex=xts(as.numeric(gsub(" ", "", moex[,2])),order.by = as.Date(moex[,1], format = "%b %d, %Y"))
# вычисление доходности за месяц
moex=monthlyReturn(to.monthly(moex, indexAt = 'lastof'))

# период оценки
period = 24
# таблица оценки
pif.act=pif.return

```

```

# очистка значений
pif.act[]=NA
# движущийся период оценки в цикле
for(i in 1:(nrow(pif.return)-(period-1))){
  # выделение данных за период в переменную
  dta.m=pif.return[i:(i+(period-1)),]
  # массив с весами оценок фондов
  diff.best=c()
  for(j in 1:ncol(dta.m)){
    # данные по каждому фонду записаны в столбец
    pif=dta.m[,j]
    diff.best[j]=NA
    # проверка на то, что количество данных по фонду
    # соответствует длине периода
    if(length(na.trim(pif))==period){
      # вычисление пересечения дат с Индексом МосБиржи
      time=intersect(index(pif),index(моех))
      # вычисление разности между фондом и индексом
      diff.index=abs(Return.excess(pif,моех[as.Date(time)]))
      # вычисление показателя для сравнения с другими фондами
      diff.best[j]=mean(diff.index)*sd(diff.index)
    }
  }
  # проверка, если все данные пустые
  if(length(diff.best[is.na(diff.best)])==length(diff.best)) next
  # заполнение названия фондов к их оценке
  names(diff.best)=colnames(dta.m)
  # останутся только те фонды, по которым сформирована оценка
  diff.best=diff.best[!is.na(diff.best)]
  #pif.best.name=names(diff.best)[which(diff.best==min(diff.best,na.rm = TRUE))]
  #pif.act[tail(index(pif),n=1),pif.best.name]=TRUE
  # запись в общую таблицу оценок фондов
  pif.act[tail(index(pif),n=1),names(sort(diff.best))]=c(1:length(diff.best))
}

# создание копии таблицы
pif.act.copy=pif.act
# период для инвестиций
period=12
# стирание данных
pif.act[]=NA
# индикатор для начала
# вычисления фонда в который
# нужно инвестировать
j=11
for(i in 1:(nrow(pif.act.copy)-(period-1))){
  # выделение данных за период в переменную
  dta.m=pif.act.copy[i:(i+(period-1)),]
  # проверка, достаточно ли данных в периоде
  if(sum(dta.m[,1],na.rm = TRUE)>0){
    # увеличение периода на 1 месяц
    j=j+1
    # если количество месяцев кратно периоду
    # то нужно вычислять новый фонд для инвестиций
    # на следующий период
    if(j%period==0){
      # вычисляется сумма по каждому фонду
      # затем данные записываются в массив
      best=names(sort(colSums(dta.m)))[1]
      # вычисление месяца начала инвестиций
      k=i+period
      # вычисление конечного месяца инвестиции
      k1=ifelse((k+(period-1))>nrow(pif.act.copy),nrow(pif.act.copy),(k+(period-1)))
      # запись в таблицу информации о том,
      # в какой фонд будут сделаны инвестиции
      pif.act[index(pif.act.copy[k:k1,]),best[1]]=1
    }
  }
}

```



```

}

portfel.equity=function(prices, action, bank, fee, log = ""){
  # индикатор, нужно ли вести логи
  diag = FALSE
  if(log!="") diag=TRUE
  # значение комиссии на покупку
  fee.buy=fee[1]
  # значение комиссии на продажу
  fee.sell=fee[2]
  # запись текущей даты в логи
  if(diag) write(as.character(Sys.time()),log,,FALSE)
  # портфель, как массив активов
  portfel=c()
  # история изменения стоимости портфеля
  portfel.history=action[,1]
  portfel.history[]=NA

  for(i in 1:nrow(action)){
    # запись даты
    if(diag) write(paste0("-----",index(action)[i],"-----"),log,,TRUE)
    # если был сплит активов
    n.act=action[i,names(portfel)]
    if(length(n.act)>0) n.act=n.act[1,!is.na(n.act)]
    if(length(n.act)>0) n.act=n.act[1,n.act>1]
    if(length(n.act)>0) portfel[colnames(n.act)]=portfel[colnames(n.act)]*n.act
    # если нет цен по всем активам в портфеле
    pr=prices[i,names(portfel)]
    if(length(pr)>0) pr=pr[1,!is.na(pr)]
    if(length(pr)!=length(portfel)){
      portfel.history[i]=equity
      if(diag) write(paste0("По данным активам торги остановлены: ",paste(setdiff(names(portfel),colnames(pr)),collapse = " ")),log,,TRUE)
      next
    }
    # стоимость портфеля
    equity=sum(as.numeric(prices[i,names(portfel)]*portfel))+bank
    # запись диагностической информации
    if(diag) write(paste("Стоимость портфеля:",equity),log,,TRUE)
    if(diag) write(paste("Банк:",bank),log,,TRUE)
    if(diag) write(paste("Портфель:",paste(paste0(names(portfel),":"),portfel,collapse=""),log,,TRUE)
    if(diag) write(paste("Количество:",length(portfel)),log,,TRUE)
    # новый список активов для портфеля
    buy=colnames(action)[which(!is.na(action[i,]))]
    # запись стоимости портфеля в историю изменений
    portfel.history[i]=equity
    # если активы в портфеле такие же, в которые нужно инвестировать
    if(length(setdiff(buy,names(portfel)))==0) next

    if(diag) write(paste("Новый портфель:",paste(buy,collapse = " ")),log,,TRUE)
    if(diag) write(paste("Количество:",length(buy)),log,,TRUE)

    # если портфель пустой и покупать ничего не надо
    if(length(portfel)==0 & length(buy)==0) next
    # вес одной части в портфеле
    part=0
    # если покупать ничего не надо
    if(length(buy)==0){
      # будет ребалансировка
      if (length(portfel)!=0) part=1/length(portfel)
    } else {
      # часть каждого актива (равновзвешенный)
      part=1/length(buy)
    }
    # стоимость одной части портфеля

```

```

part.money=equity*part
if(diag) write(paste("Вс 1 части:",part),log,,TRUE)

# новые активы, которые нужно купить в портфель: разница между новым списком активов для
портфеля и текущим портфелем
buy.new=setdiff(buy,names(portfel))

# акивы из портфеля, которые нужно продать: не входят в новой список активов для портфеля
sell.new=setdiff(names(portfel), buy)

#### сначала продаём активы из портфеля, чтобы пополнить количество денежных средств ####
if(length(sell.new)>0) {
  if(diag) write(paste("##### Продажа ",paste(sell.new,collapse = " ")),log,,TRUE)
  for(p in sell.new){
    # текущая цены пифа
    price=as.numeric(prices[i,p])
    # если нет цены - переход к следующему активу
    if(is.na(price)) next
    # вычисление комиссии
    fee=price * fee.sell
    # количество данного актива в портфеле
    n=as.numeric(portfel[p])
    if(diag) write(paste("ПИФ:",p,"Цена:",price,"Кол-
во:",n,"Итого:",n*price,"Комиссия:",n*fee),log,,TRUE)
    # изменение цены с учётом комиссии
    price=price - fee
    # продажа и пополнение количества денежных средств
    bank=bank + n*price
  }
  # исключение проданных пифов из портфеля
  portfel=portfel[! names(portfel) %in% sell.new]
}
# созранение информации о стоимости портфеля
portfel.history[i]==sum(as.numeric(prices[i,names(portfel)]*portfel))+bank
# портфель стал пустым и покупать ничего не нужно
if(length(portfel)==0 & length(buy.new)==0) next

#### далее ребалансировка оставшихся пифов в портфеле (не покупка новых) ####

# цикл по оставшимся пифам в портфеле
#### ребалансировка на продажу ####
if(diag) write("##### Допродажа #####",log,,TRUE)
for(p in names(portfel)){
  # текущая цены актива
  price=as.numeric(prices[i,p])
  # если нет цены - переход к следующему активу
  if(is.na(price)) next
  # количество пифа в портфеле
  n=as.numeric(portfel[p])
  # стоимость доли данного пифа
  part.price=price*n
  # величина части в портфеле
  part.p=part.price/equity
  # если стоимости доли актива больше новой стоимости одной части, то продажа избытка
  if(part.p > part){
    # разница для продажи
    diff.m=part.price - part.money
    # количество, которое будет продано (округление вверх)
    n.new= ceiling(diff.m / price)
    # если ничего не нужно продавать, то выход
    if(n.new==0) next
    # вычисление комиссии
    fee=price*fee.sell
    if(diag) write(paste("ПИФ:",p,"Цена:",price,"Кол-
во:",n.new,"Итого:",n.new*price,"Комиссия",n.new*fee),log,,TRUE)
    # изменение цены с учётом комиссии
    price=price-fee
    # вычитание проданного количества пифов

```

```

    portfel[p]=portfel[p] - n.new
    # увеличение банка
    bank=bank + n.new*price
    # если был продан весь актив
    portfel[portfel!=0]
  }
}

#### ребалансировка на покупку ####
if(diag) write("##### Допопкупка #####",log,,TRUE)
for(p in names(portfel)){
  # текущая цены актива
  price=as.numeric(prices[i,p])
  # если нет цены - переход к следующему активу
  if(is.na(price)) next
  # количество актива в портфеле
  n=as.numeric(portfel[p])
  # стоимость актива в портфеле
  part.price=price*n
  # величина части в портфеле
  part.p=part.price/equity
  # если часть меньше, то нужно докупать
  if(part.p < part){
    # разница для покупки
    diff.m=part.money - part.price
    # вычисление комиссии
    fee=price*fee.buy
    # цена без комиссии
    price.old=price
    # цена с комиссией
    price=price + fee
    # количество, которое можно докупить
    n.new=floor(diff.m / price)
    # если ничего не надо докупать - переход к следующему активу
    if(n.new==0) next
    if(diag) write(paste("ПИФ:",p,"Цена:",price.old,"Кол-
во:",n.new,"Итого:",n.new*price.old,"Комиссия:",n.new*fee),log,,TRUE)
    # добавление купленного количества актива в портфель
    portfel[p]=portfel[p] + n.new
    # уменьшение свободных денежных средств
    bank=bank-n.new*price
  }
}

#### покупка новых пифов ####
if(length(buy.new)>0) {
  if(diag) write(paste("##### Покупка ",paste(buy.new,collapse = " ")),log,,TRUE)
  # если портфель - это один актив, то стоимость одной части
  # это все доступные денежные средства
  if(part==1) part.money=bank
  for(p in buy.new){
    # текущая цены актива
    price=as.numeric(prices[i,p])
    # если нет цены - переход к следующему активу
    if(is.na(price)) next
    # вычисление комиссии
    fee=price*fee.buy
    # цена без комиссии
    price.old=price
    # цена с комиссией
    price=price+fee
    # количество актива, которое можно купить
    n=floor(part.money / price)
    if(diag) write(paste("Бюджет:",part.money,"ПИФ:",p,"Цена:",price.old,"Кол-
во:",n,"Итого:",n*price.old,"Комиссия:",n*fee),log,,TRUE)
    # если ничего не купить, то выход
    if(n==0) next
    # добавление актива в портфель

```

```

    portfel=append(portfel,n)
    # уменьшение свободных денежных средств
    bank=bank - n*price
    # добавление названия актива в портфель
    names(portfel)[length(portfel)]=p
  }
}
#####
if(diag) write("#####",log,,TRUE)
# вычисление стоимости портфеля
equity=sum(as.numeric(prices[i,names(portfel)]*portfel))+bank
if(diag) write(paste("Стоимость портфеля:",equity),log,,TRUE)
if(diag) write(paste("Банк:",bank),log,,TRUE)
if(diag) write(paste("Портфель:",paste(paste0(names(portfel),":"),portfel,collapse=",")),log,,TRUE)
if(diag) write(paste("Количество:",length(portfel)),log,,TRUE)
# сохранение информации о стоимости портфеля
portfel.history[i]=equity
}
# в качестве результата возвращается
# таблица данных о временной стоимости портфеля
return(portfel.history)
}

require(PerformanceAnalytics)

Sys.setlocale("LC_TIME", "English")
moex10=read.csv("~/MOEX10.csv",stringsAsFactors = FALSE)
moex10=xts(as.numeric(gsub(",","", moex10[,2])),order.by = as.Date(moex10[,1], format = "%b %d, %Y"))
moex10=monthlyReturn(to.monthly(moex10, indexAt = 'lastof'))

boot.all = function(return, bench, rf, boot.n = 10, period = 24, l1 = 0.99, l2 = 0.5){
  # библиотека для перемешивания значений массива
  require(tseries)
  # основная таблица t-статистик
  boot=matrix(NA,nrow = ncol(return),ncol = (boot.n + 1))
  # название строк соответствует названия фондов
  row.names(boot)=colnames(return)
  # цикл по каждому фонду
  for(j in 1:ncol(return)){
    # данные по одному фонду
    pif=return[,j]
    # удаление пустых значений
    pif=pif[!is.na(pif)]
    # если достаточное количество данных
    if(length(na.trim(pif))>period){
      # пересечение данных по фонду, бенчмарку и безрисковой ставке по времени
      time=intersect(intersect(index(pif),index(rf)),index(bench))
      # если достаточно общего количества данных
      if(length(time)>period){
        # избыточная доходность фонда
        Er=Return.excess(pif[as.Date(time)],rf[as.Date(time)])
        # избыточная доходность бенчмарка
        Eb=Return.excess(bench[as.Date(time)],rf[as.Date(time)])
        # если количество данных совпадает
        if(length(Er)==length(Eb)){
          # регрессия на реальных данных
          carm=summary(lm(Er ~ Eb))
          # альфа коэффициент
          carm.alfa=carm$coefficients[1,1]
          # t-статистика альфа коэффициента
          carm.t=carm$coefficients[1,3]
          # бета коэффициент
          carm.beta=carm$coefficients[2,1]
          # остатки, ошибки регрессии
          carm.res=carm$residuals
          # сохранение реального значения t-статистики

```

```

boot[j,1]=capm.t
# восстановление избыточной доходности
const = capm.beta * Eb
# бутстрап
# цикл регрессий по восстановленным данным
for(k in 1:boot.n){
  # перемешивание остатков
  res=xts(tsbootstrap(ts(capm.res)), order.by = index(capm.res))
  # добавление остатков к восстановленным данным
  Er.b = const + res
  # регрессия на восстановленных данных
  capm.b = summary(lm(Er.b ~ const))
  # сохранение симулированной t-статистики
  boot[j,(1+k)]=capm.b$coefficients[1,3]
}
}
}
}

# массив лучших симулированных коэффициентов
t.best={}
# цикл по столбцам бутстрапа
for (l in 2:boot.n){
  # отбор и добавление лучших коэффициентов из столбца
  t.best=c(t.best,boot[which(boot[,l]>=quantile(na.omit(boot[,l]),c(l1)) & boot[,l]>0),l])
}

# массив лучших фондов
pif.best={}
# цикл по первому столбцу таблицы
# с коэффициентами на реальных данных
for(l in 1:nrow(boot)){
  # если есть значение коэффициента у фонда
  if(!is.na(boot[l,1])){
    # если значение коэффициента выше уровня квантили массива лучших коэффициентов
    if(boot[l,1]>quantile(t.best,c(l2))){
      # добавление значения коэффициента фонда в массив
      pif.best=c(pif.best,boot[l,1])
      # добавления названия фонда к коэффициенту
      names(pif.best)[length(pif.best)]=row.names(boot)[l]
    }
  }
}

# сортировка массива лучших фондов
# по убыванию значений коэффициентов
pif.best=sort(pif.best, decreasing = TRUE)
# возврат в качестве результата
# массив лучших фондов
return(pif.best)
}

# бенчмарк портфель из индексных паевых фондов
benchmark.pif=portfel.equity(pif.price, pif.act, 10000, c(0.005, 0.015))
# доходность бенчмарка
benchmark.pif.return=pif.graph(benchmark.pif, moex)
# функция вычисления лучших фондов
best.pif=boot.all(return = pif.return, bench = benchmark.pif.return, rf = ru.1y, boot.n = 100,
period = 24, l1 = 0.99, l2 = 0.5)
# результат
cat(paste(all[,3] %in% names(best.pif),1),round(best.pif,2)),sep = "\n")

# вызов функции с параметрами
benchmark.stock=portfel.equity(data.10, data.act, 10000, c(0.0001, 0.0001))
# месячная доходность бенчмарка
benchmark.stock.return=monthlyReturn(Cl(to.monthly(benchmark.stock, indexAt = 'lastof'))))
# функция вычисления лучших фондов

```

```

best.stock=boot.all(return = pif.return, bench = benchmark.stock.return, rf = ru.1y, boot.n =
100, period = 24, l1 = 0.99, l2 = 0.5)
# результат
cat(paste(all[,3] %in% names(best.stock),1],round(best.stock,2)),sep = "\n")

# вызов функции
benchmark.future=future.benchmark()
# месячная доходность бенчмарка
benchmark.future.return=monthlyReturn(Cl(to.monthly(benchmark.future, indexAt = 'lastof'))))
# функция вычисления лучших фондов
best.future=boot.all(return = pif.return, bench = benchmark.future.return, rf = ru.1y, boot.n =
100, period = 24, l1 = 0.99, l2 = 0.5)
# результат
cat(paste(all[,3] %in% names(best.future),1],round(best.future,2)),sep = "\n")

boot.history = function(return, bench, rf, boot.n = 10, period = 24, l1 = 0.99, l2 = 0.5){
  # библиотека для перемешивания значений массива
  require(tseries)
  # основная таблица с результатами бутстрапа
  # структура копируется с таблицы доходностей активов
  boot.res=return
  # очистка данных
  boot.res[]=NA
  # цикл с движущимся окном
  for(i in 1:(nrow(return)-(period-1))){
    # данные по доходности активов за период
    dta.m=return[i:(i+(period-1)),]
    # таблица t-статистик
    boot=matrix(NA,nrow = ncol(dta.m),ncol = (boot.n + 1))
    # название строк соответствует названию активов
    row.names(boot)=colnames(dta.m)
    # цикл по каждому активу
    for(j in 1:ncol(dta.m)){
      # данные по активу за период
      pif=dta.m[,j]
      # если достаточное количество данных
      if(length(na.trim(pif))==period){
        # пересечение данных по фонду, бенчмарку и безрисковой ставке по времени
        time=intersect(intersect(index(pif),index(rf)),index(bench))
        # если достаточно общего количества данных
        if(length(time)==period){
          # избыточная доходность фонда
          Er=Return.excess(pif[as.Date(time)],rf[as.Date(time)])
          # избыточная доходность бенчмарка
          Eb=Return.excess(bench[as.Date(time)],rf[as.Date(time)])
          # если количество данных совпадает
          if(length(Er)==length(Eb)){
            # регрессия на реальных данных
            capm=summary(lm(Er ~ Eb))
            # альфа коэффициент
            capm.alfa=capm$coefficients[1,1]
            # t-статистика альфа коэффициента
            capm.t=capm$coefficients[1,3]
            # бета коэффициент
            capm.beta=capm$coefficients[2,1]
            # остатки, ошибки регрессии
            capm.res=capm$residuals
            # сохранение реального значения t-статистики
            boot[j,1]=capm.t
            # восстановление избыточной доходности
            const = capm.beta * Eb
            # бутстрап
            # цикл регрессий по восстановленным данным
            for(k in 1:boot.n){
              # перемешивание остатков
              res=xts(tsbootstrap(ts(capm.res)), order.by = index(capm.res))
              # добавление остатков к восстановленным данным
              Er.b = const + res
            }
          }
        }
      }
    }
  }
  boot.res=boot
}

```

```

        # регрессия на восстановленных данных
        capm.b = summary(lm(Er.b ~ const))
        # сохранение симулированной t-статистики
        boot[j,(1+k)]=capm.b$coefficients[1,3]
    }
}
}

# массив лучших симулированных коэффициентов
t.best={}
# цикл по столбцам бутстрапа
for (l in 2:boot.n){
    # отбор и добавление лучших коэффициентов из столбца
    t.best=c(t.best,boot[which(boot[,l]>=quantile(na.omit(boot[,1]),c(11)) & boot[,1]>0),1])
}

# массив лучших фондов
pif.best={}
# цикл по первому столбцу таблицы
# с коэффициентами на реальных данных
for(l in 1:nrow(boot)){
    # если есть значение коэффициента у фонда
    if(!is.na(boot[l,1])){
        # если значение коэффициента выше уровня квантили массива лучших коэффициентов
        if(boot[l,1]>quantile(t.best,c(12))){
            # добавление значения коэффициента фонда в массив
            pif.best=c(pif.best,boot[l,1])
            # добавления названия фонда к коэффициенту
            names(pif.best)[length(pif.best)]=row.names(boot)[l]
        }
    }
}

# сортировка массива лучших фондов
# по убыванию значений коэффициентов
pif.best=sort(pif.best, decreasing = TRUE)
# лучшие активы за данный период отмечаются в общей таблице
boot.res[tail(index(dta.m),n=1),names(pif.best)]=c(1:length(pif.best))
}

# массив сумм по столбцам
sums=colSums(boot.res,na.rm = TRUE)
# удаляюся те фонды
# которые ни разу не были отмечены
sums=sums[sums>0]
# удаление данных фондов из таблицы
boot.res=boot.res[,names(sums)]
# возвращение таблицы в качестве результата функции
return(boot.res)
}

boot.act=function(boot.res, period = 4){
    # создание копии таблицы
    copy=boot.res
    # стирание данных
    copy[]=NA
    # индикатор для начала
    # вычисления фонда в который
    # нужно инвестировать
    j=period - 1
    for(i in 1:(nrow(boot.res)-(period-1))){
        # выделение данных за период в переменную
        dta.m=boot.res[i:(i+(period-1)),]
        # проверка, достаточно ли данных в периоде
        if(length(sort(colSums(dta.m)))>0){
            # увеличение периода на 1 месяц
            j=j+1
            # если количество месяцев кратно периоду

```

```

# то нужно вычислять новый фонд для инвестиций
# на следующий период
if(j%%period==0){
  # вычисляется сумма по каждому фонду
  # затем данные записываются в массив
  best=names(sort(colSums(dta.m)))
  n=as.numeric(floor(quantile(c(0:length(best)),.25)))
  # вычисление месяца начала инвестиций
  #k=i+period
  k=ifelse((i+period)>=nrow(copy),nrow(copy),(i+period))
  # вычисление конечного месяца инвестиции
  k1=ifelse((k+(period-1))>=nrow(copy),nrow(copy),(k+(period-1)))
  # запись в таблицу информации о том,
  # в какой фонд будут сделаны инвестиции
  copy[index(copy[k:k1,]),best[1]]=1
}
}
}
# массив сумм по столбцам
sums=colSums(copy,na.rm = TRUE)
# удаляюся те фонды
# которые ни разу не были отмечены
sums=sums[sums>0]
# удаление данных фондов из таблицы
copy=copy[,names(sums)]
# возвращение таблицы в качестве результата функции
return(copy)
}

for(i in 1:nrow(copy)){
  if(sum(copy[i,],na.rm = TRUE)>0){
    break
  }
}

# функция вычисления лучших фондов по периодам
best.pif.boot=boot.history(return = pif.return, bench = benchmark.pif.return, rf = ru.1y, boot.n
= 10, period = 24, l1 = 0.99, l2 = 0.5)
# функция для установки индикаторов
best.pif.act=boot.act(best.pif.boot,4)
# построение портфеля
portfel=portfel.equity(pif.price, best.pif.act, 10000000, c(0.005, 0.015))
# доходность портфеля
portfel.return=pif.graph(portfel,moex)

pif.graph = function(portfel, bench, file=""){
  i=index(portfel)[which(as.numeric(portfel)!=as.numeric(portfel[1]))[1]]
  # пересечение дат с индексом
  dt=intersect(index(portfel),index(bench))
  i=which(as.Date(dt)==i)
  dt=dt[i:length(dt)]
  test=cbind(bench[as.Date(dt)],Return.calculate(portfel[as.Date(dt)]))
  # наименование столбцов
  colnames(test)=c("MOEX", "Portfel")
  # построение графика
  if(file!=""){
    png(file,width = 1000,height = 1155, units = "px", pointsize = 20)
    charts.PerformanceSummary(test)
    dev.off()
  } else {
    charts.PerformanceSummary(test)
  }
  # регрессия
  print(summary(lm(test[,2]~test[,1])))
  portfel.return=Return.calculate(portfel[as.Date(dt)])
  portfel.return[1]=0
  return(portfel.return)
}

```



```

future.benchmark = function(){
  # чтение данных из файла
  future=read.csv("F:/vkr/MX.csv", stringsAsFactors = F)
  # наименование фьючерсов
  future.name=unique(future[,2])
  # таблица со значениями ГО по фьючерсам
  future.data=as.data.frame(matrix(NA,nrow = nrow(future), ncol = length(future.name)))
  # название строк
  row.names(future.data)=future[,1]
  # название столбцов
  colnames(future.data)=future.name
  # преобразование во временной формат данных
  future.data=as.xts(future.data, order.by = as.Date(row.names(future.data), "%d.%m.%Y"))
  # создание новых таблиц копированием
  future.price=future.price.estim=future.action=future.data

  # заполнение данных по каждому фьючерсу
  for(name in future.name){
    # временная переменная
    x=future[future[,2]==name,]
    # преобразование формата даты
    date=as.Date(x[,1], "%d.%m.%Y")
    # заполнение значений ГО
    future.data[date,name]=x[,16]
    # заполнение актуальности фьючерса
    future.action[date,name]=1
    # заполнение значений средней взвешенной цены
    future.price[date,name]=x[,3]
    # заполнение значений цены расчета
    future.price.estim[date,name]=x[,4]
    # удаление временных переменных
    rm(x,date)
  }

  # величина свободных денежных средств
  portfel.bank=1000000
  # наименование текущего фьючерса в портфеле
  portfel.name=""
  # история изменения стоимости портфеля
  portfel.history=future.data[,1]
  # стирание данных
  portfel.history[]=NA

  for(i in 1:nrow(future.data)){
    # фьючерс в который нужно инвестировать на данной итерации
    buy=colnames(future.action)[which(!is.na(future.action[i,]))]
    # если фьючерс отличается от фьючерса в портфеле
    # и портфель не пустой то происходит освобождение ГО
    if(buy!=portfel.name & portfel.name!=""){
      # ГО возвращается в свободные средства
      portfel.bank=portfel.bank + portfel.warranty
      # удаление переменных
      rm(portfel.warranty,portfel.count,portfel.price)
    }

    # если новый фьючерс и фьючерс в портфеле не совпадают
    # то нужно инвестировать в данный фьючерс
    # ГО было освобождено на предыдущем шаге
    if(buy!=portfel.name){
      # текущая стоимость фьючерса - это ГО
      price=as.numeric(future.data[i,buy])
      # инвестиции без "плеча"
      # цена количества фьючерсов покрывает весь банк
      n=ceiling(portfel.bank / as.numeric(future.price[i,buy]))
      # свободные денежные средства
      # уменьшаются только на величину ГО
      portfel.bank=portfel.bank - n * price
      # сохранение количества

```

```

    portfel.count=n
    # сохранение наименования фьючерса
    portfel.name=buy
    # цена фьючерса для вычисления маржинального требования
    portfel.price=as.numeric(future.price[i,buy])
    # общий размер ГО
    portfel.warranty=n * price
    # удаление переменных
    rm(price,cash,n)
  }
  # если в портфеле есть фьючерс
  if(portfel.count > 0){
    # актуальное значение ГО
    warranty.price = as.numeric(future.data[i,portfel.name])
    if(!is.na(warranty.price)){
      # вычисление актуального ГО покрывающее портфель
      warranty=warranty.price * portfel.count
      # разница между зарезервированным и актуальным ГО
      diff.warranty = portfel.warranty - warranty
      # изменение величины свободных средств
      portfel.bank = portfel.bank + diff.warranty
      # зарезервированное ГО становится актуальным
      portfel.warranty = warranty
    }
    # вычисление маржинального требования
    if(portfel.price==as.numeric(future.price[i,portfel.name])){
      # инвестировано было в данный день
      margin=(as.numeric(future.price.estim[i,portfel.name]) - portfel.price) * portfel.count
    } else {
      # между двумя датами как разница расчётных цен
      margin=(as.numeric(future.price.estim[i,portfel.name]) -
as.numeric(future.price.estim[(i-1),portfel.name])) * portfel.count
    }
    # если свободных средств и ГО не хватает
    # для покрытия маржинального требования
    if((portfel.bank + portfel.warranty) < margin){
      # происходит margin call
      print("Банк опустел")
      break
    }
    # в конце каждого торгового дня свободные средства
    # изменяются на величину маржинального требования
    portfel.bank = portfel.bank + margin
    # удаление переменных
    rm(warranty,margin,diff.warranty,warranty.price)
  }

  portfel.history[i]=portfel.bank + portfel.warranty
}
return(portfel.history)
}

# получение наименования файлов
files=list.files("~/dat",full.names=TRUE)
# общая таблица с данными
data.equity=data.frame()
for(file in files){
  # чтение одного файла
  eq=read.csv(file,stringsAsFactors=FALSE)
  # добавление данных в одну таблицу
  data.equity=rbind(data.equity,eq)
}
tickers<-unique(data.equity[,1])
setwd("F:/vkr")
dir=getwd()

# информация о базе расчета Индекс ММББ 10
mic.10<-read.csv(paste0(dir,"/10micexdata.csv"),header = FALSE,stringsAsFactors = FALSE)

```

```

# изменения наименования сокращенных названий акций
mic.10.replace=read.csv(paste0(dir,"/10micex.csv"),header = FALSE,stringsAsFactors = FALSE)
# замена наименования сокращенных названий акций в базе расчета
for(i in 1:nrow(mic.10.replace)){
  mic.10[mic.10==mic.10.replace[i,1]]=mic.10.replace[i,2]
}

mic.ticker=c()
mic.ticker=unique(unlist(sapply(c(18:nrow(mic.10)),function(i){mic.ticker=append(mic.ticker,mic.10[i,3:12])})))
mic.ticker=mic.ticker[mic.ticker!=""]

date.all=c() # массив со всеми датами
for(i in 18:nrow(mic.10)){
  # база расчета Индекса ММВБ 10 за определенный период
  tickers=unlist(mic.10[i,3:length(mic.10[i,])])
  # массив календарных дат для базы расчета
  dates=seq.Date(as.Date(mic.10[i,1],"%d.%m.%Y"), as.Date(mic.10[i,2],"%d.%m.%Y"), by = "day")
  date.t=c() # массив дат для текущей базы расчета
  for(t in tickers){
    # данные по одной акции из базы расчёта
    y=data.equity[which(data.equity[,1]==t),]
    # пересечение календарных дат и торговых дней
    date.inter=intersect(dates,as.Date(y[,2]))
    # добавление пересечения дат в массив
    date.t=append(date.t,date.inter)
  }
  # добавление уникальных дат в общий массив
  date.all=append(date.all,unique(date.t))
}

# структурированная таблица с данными
data.10=xts(matrix(NA, nrow = length(date.all), ncol = length(mic.ticker),dimnames =
list(date.all,mic.ticker)),order.by = as.Date(date.all))

for(t in colnames(data.10)){
  # данные по одной акции из базы расчёта
  y=data.equity[which(data.equity[,1]==t),]
  # цены закрытия по данной акции
  y=xts(Cl(y),order.by=as.Date(y[,2]))
  # заполнение данных в таблице
  data.10[index(y),t]=y[index(data.10)]
}

for(i in 18:nrow(mic.10)){
  # база расчета Индекса ММВБ 10 за определенный период
  tickers=unlist(mic.10[i,3:length(mic.10[i,])])
  for(t in tickers){
    if(t=="") next
    # данные по одной акции из базы расчёта
    y=data.equity[which(data.equity[,1]==t),]
    # цены закрытия по данной акции
    y=xts(Cl(y),order.by=as.Date(y[,2]))
    # пересечение дат в таблице и торговых дат для акции
    #date.inter=intersect(index(data.10),index(y))
    # заполнение данных в таблице
    data.10[index(y),t]=y
  }
}

for(t in colnames(data.10)){
  y=data.equity[which(data.equity[,1]==t),]
  # цены закрытия по данной акции
  y=xts(Cl(y),order.by=as.Date(y[,2]))
  # заполнение данных в таблице
  data.10[index(y),t]=y[index(data.10)]
}

```

```

for(i in 1:nrow(data.10)){
  r=data.10[i,]
  if(length(r[!is.na(r)])<10) print(index(r))
}

# структурированная таблица с данными
data.act=xts(matrix(NA, nrow = length(date.all), ncol = length(mic.ticker),dimnames =
list(date.all,mic.ticker)),order.by = as.Date(date.all))
for(i in 18:nrow(mic.10)){
  # база расчёта индекса
  tickers=unlist(mic.10[i,3:length(mic.10[i,])])
  # удаление пустых значений
  tickers=tickers[tickers!=""]
  # период действия базы расчёта
  dates=seq.Date(as.Date(mic.10[i,1],"d.%m.%Y"), as.Date(mic.10[i,2],"d.%m.%Y"), by = "day")
  # заполнение индикаторов, о вхождении акций в портфель
  data.act[dates,tickers]=TRUE
}
# сплит акций
data.act["2007-07-18","SBER"]=1000
data.act["2007-07-18","SBERP"]=20
# заполнение индикаторов
# чтобы акции не были исключены из портфеля
data.act[c("2007-07-19", "2007-07-20"),c("SBER", "SBERP")]=TRUE

```