

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования

«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

Кафедра «Прикладная математика и информатика»

09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В ОБРАЗОВАНИИ И
ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЯХ

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему **МЕТОДЫ И ИНСТРУМЕНТАРИИ РАЗРАБОТКИ СЕТЕВЫХ
УЧЕБНЫХ КОМПЛЕКСОВ**

Студент

Б.С. Асоев

Научный
руководитель

О.В. Аникина

Руководитель программы

А.Н. Ярыгин

д.п.н., профессор

« » _____ 2017 г.

Допустить к защите

Заведующий кафедрой доцент, к.т.н., доцент А.В. Очеповский

« » _____ 2017 г.

Тольятти 2017

ОГЛАВЛЕНИЕ:

ВВЕДЕНИЕ.....	4
Глава 1 АНАЛИЗ МЕТОДОВ И ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РАЗРАБОТКИ ЭЛЕКТРОННЫХ УЧЕБНЫХ КОМПЛЕКСОВ	8
1.1 Обзор литературных источников.....	8
1.2 Обзор технологий и продуктов, поддерживающих систему электронного обучения	12
Глава 2 ТЕХНОЛОГИЯ РАЗРАБОТКИ ИНТЕРАКТИВНЫХ УЧЕБНЫХ WEB-ПРИЛОЖЕНИЙ SHINY	26
2.1 Структура приложения Shiny	26
2.2 Разметка HTML-страниц и каскадные таблицы стилей в приложении Shiny	30
2.3 Раскладка web-страницы в приложении Shiny	33
2.4 Компоненты ввода и вывода в приложении Shiny.....	36
2.5 Серверный сценарий приложения Shiny.....	43
2.6 Рекомендации по созданию интерактивных учебных приложений.	55
Глава 3 РАЗРАБОТКА ИНТЕРАКТИВНЫХ УЧЕБНЫХ WEB- ПРИЛОЖЕНИЙ.....	58
3.1 Анализ выборки данных Фишера, пользовательский интерфейс интерактивного web-приложения кластеризации данных	58
3.2 Логический анализ и программный код интерактивного web- приложения кластеризации данных.....	65
3.3 Анализ выборки данных mtcars, пользовательский интерфейс интерактивного web-приложения Shiny.....	67
3.4 Логический анализ и программный код интерактивного web- приложения mtcars.....	70

Глава 4 ТЕСТИРОВАНИЕ И АНАЛИЗ ЭФФЕКТИВНОСТИ ИНТЕРАКТИВНЫХ WEB-ПРИЛОЖЕНИЙ	74
4.1 Тестирование и анализ эффективности интерактивного web- приложения кластеризации	74
4.2 Тестирование и анализ эффективности интерактивного web- приложения mtcars	77
4.3 Апробация и оценка эффективности интерактивных учебных web- приложений	79
ЗАКЛЮЧЕНИЕ	87
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	88
ПРИЛОЖЕНИЕ А. HTML-код приложения кластеризации Ирисы Фишера	94
ПРИЛОЖЕНИЕ Б. HTML-код приложения mtcars	100

ВВЕДЕНИЕ

В современном учебном процессе все большее внимание уделяется использованию компьютерных технологий, реализуемых в виде сетевых сервисов, открывающих новые возможности распределенной учебной и профессиональной деятельности. Однако факт использования компьютера в учебном процессе еще не является залогом успеха, и значимых результатов можно добиться только при грамотном его применении.

Современный уровень компьютерной техники, быстрое развитие интернет открывают возможности для создания информационно-обучающей среды, под которой понимают комплекс современных информационных технологий, обеспеченных необходимыми методическими, программными и техническими средствами, ориентированными на процесс обучения. В связи с этим развиваются информационные ресурсы и инструменты их создания. Все больше программ ориентированы на создание web-приложений и интерактивных программных продуктов. Хорошая интерактивность и визуализация достигается путем использования встроенных инструментальных средств, создания графических объектов и эффективного языка программирования. Одним из средств формирования информационно-обучающей среды в университете является создание электронных учебно-методических комплексов по дисциплинам, прежде всего естественно-научным и общепрофессиональным.

Актуальность исследования обусловлена необходимостью разработки современных интерактивных средств обучения. Актуальным направлением разработки образовательных ресурсов является создание интерактивных учебных модулей/компонентов, способствующих повышению эффективности учебного процесса.

Целью исследования является исследование и разработка учебных интерактивных web-приложений для демонстрации физических процессов и динамических систем по дисциплинам естественно-научного профиля.

Объект исследования – процесс профессиональной подготовки студентов вузов, **предмет исследования** – использование интерактивных электронных образовательных технологий в процессе профессиональной подготовки студентов (на примере естественно-научных направлений).

Гипотеза исследования состоит в предположении, что внедрение в учебный процесс интерактивных электронных образовательных технологий упрощает формирование учебного материала и его визуальное представление, а также способствует усвояемости учебного материала и улучшению показателей успеваемости студентов.

Цель исследования определила необходимость решения следующих **задач**:

1. На основе исследования научной литературы определить требования к технологии разработки интерактивных учебных web-компонентов.

2. Исследовать возможности программного инструментария для разработки интерактивных учебных web-компонентов.

3. Предложить ряд рекомендаций по созданию интерактивных учебных web-приложений.

4. Разработать и протестировать интерактивные учебные web-приложения на примере решения классических задач.

5. Выработать практические рекомендации и порядок работы с разработанными учебными web-приложениями.

6. Экспериментально проверить разработанные интерактивные учебные web-приложения и определить результативность их внедрения в учебный процесс.

Методологическую базу исследования составляют фундаментальные работы, опубликованные как отечественными, так и зарубежными авторами, в области дистанционного образования (Андреев А.А., Соловов А.В., Полат Е.С., Носкова Т.Н., Батаев А.В., Ульченко Е.Н., Лисицына Л.С., Першин А.А.), различных аспектов разработки электронных учебных комплексов и материалов (Хабибуллин Р., Плещинская И.Е., Галкина Л.А., Миллз Крис,

Усмонов М.С., Василевский Ю.В., Алексеев Г.В., Бриденко И.И., Савельев А.О., Ларина Э.С., Джеймс Г., Уиттон Д., Хасты Т., Тибширани Р., Вильямс Г.), разработке интерактивных компьютерных моделей и web-приложений shiny (Zev Ross, Garrett Grolemond, Joseph J. Allaire, Winston Chang и др.).

Теоретической основой исследований послужили научные труды отечественных и зарубежных исследователей в области разработки интерактивных компьютерных моделей и web-приложений и их применения в учебном процессе.

Основные методы исследования:

- теоретические – теоретический анализ, моделирование и проектирование, анализ состояния электронного обучения и образования в мире и России;
- общенаучные – моделирование, обобщение, проектирование, анализ, классификация и систематизация.

Основные этапы исследования: исследование велось с 2015 по 2017 гг. в три этапа.

1. этап (2015 г.) - изучение научной литературы, ресурсов электронных библиотек, сравнительно-сопоставительный анализ теоретических подходов к исследуемой проблеме разработки и использования в учебном процессе интерактивных технологий;

2. этап (2015-2016 гг.) - разработка интерактивных учебных моделей для образовательного процесса, проведение формирующего этапа эксперимента;

3. этап (2016-2017 гг.) - обобщение и систематизация полученной информации, статистическая обработка и качественный анализ результатов исследования.

Апробация результатов исследования осуществлялась на кафедре «Прикладная математика и информатика» Тольяттинского государственного университета. Результаты исследования были представлены на III научно-практической всероссийской конференции (школе-семинаре) молодых ученых

«Прикладная математика и информатика: современные исследования в области естественных и технических наук» (Тольятти, 24-25 апреля 2017 года).

Научная новизна исследования состоит в том, что в нем обоснованы принципы и внедрена технология создания интерактивных учебных web-приложений в среде R с использованием пакета расширения Shiny. Разработаны рекомендации по созданию интерактивных web-приложений и порядок работы с разработанными учебными демонстрациями.

Практическая значимость исследования состоит в том, что спроектированы и реализованы интерактивные учебные модели решения классических задач, позволяющие виртуализировать лабораторный практикум по дисциплинам естественно-научного профиля.

Положения, выносимые на защиту:

1. Принципы создания интерактивных учебных web-приложений в среде R с использованием пакета расширения Shiny.
2. Разработанные интерактивные учебные web-приложения.
3. Результаты тестирования разработанных учебных web-приложений, демонстрирующих эффективность внедрения интерактивных технологий в учебный процесс.

Работа является результатом теоретической и практической деятельности в области разработки интерактивных учебных компонентов для сетевых учебных комплексов.

Объем и структура диссертационного исследования: диссертационное исследование состоит из введения, четырех глав, заключения, библиографии и приложений. Работа изложена на 93 страницах, содержит 37 рисунков, 7 таблиц.

Глава 1 АНАЛИЗ МЕТОДОВ И ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РАЗРАБОТКИ ЭЛЕКТРОННЫХ УЧЕБНЫХ КОМПЛЕКСОВ

1.1 Обзор литературных источников

Основным критерием качества подготовки специалистов можно считать достижение необходимого уровня знаний, умений и навыков, соответствующих требованиям Государственных образовательных стандартов высшего образования. С этих позиций предъявляются все более высокие требования к технологиям и средствам обучения.

В связи с стремительным развитием информационных и коммуникационных технологий появляются новые возможности оптимизации образовательного процесса. Поэтому целесообразно организовать систему доступа к учебно-методическим материалам с использованием современных компьютерных технологий.

Важным средством организации учебного процесса, в том числе в дистанционной форме, является сетевой учебно-методический комплекс [40].

Учебно-методический комплекс представляет собой систему взаимосвязанных и взаимодополняющих средств обучения, разрабатываемых в соответствии с учебной программой, достаточных для реализации целей и содержания образовательного стандарта [10]. Учебный объект – блок содержания электронного обучения, предназначенный для многократного использования [31]. Учебный объект имеет конкретную дидактическую цель и состоит из объектов содержания, к которым можно отнести, например, модуль учебного курса для выполнения какой-либо локальной дидактической задачи.

Сетевой учебный комплекс представляет собой учебное средство, ориентированное на возможность использования, прежде всего, в Web-среде.

Таким образом, сетевой учебный комплекс предназначен для оказания помощи в изучении и систематизации теоретических знаний, формирования практических навыков работы как в предметной области, так и в системе

традиционного и дистанционного образования, а также в организации самостоятельной работы обучающегося.

Интерактивность электронного учебного курса обеспечивается совокупностью мультимедийных технологий, которые в свою очередь обеспечивают обучающимся оперативную и персональную обратную связь в процессе освоения учебного материала за счет взаимодействия с элементами контента, таким образом расширяя возможности организации самостоятельной учебной работы студентов.

Изучение особенностей разработки и применения в учебном процессе интерактивных образовательных ресурсов становится все более актуальной задачей, которая обусловлена целым рядом причин:

- расширение возможности использования материалов не только в рамках изучения основной дисциплины, но и как курс дополнительного образования;
- возможность использования электронного учебного комплекса не только в локальной версии, но посредством систем дистанционного обучения.

Электронный учебный комплекс в отличие от традиционного учебника, обеспечивает:

- мощные иллюстративные возможности - использование картинок, анимации и мультимедийных материалов;
- интерактивность – представление учебного материала может изменяться в зависимости от действий обучающегося;
- различные варианты контроля и оценки полученных знаний (тесты, упражнения).
- использование пиктограмм для обозначения ключевых образов.

Вопросы дистанционного образования отражены в трудах Андреева А.А., Полат Е.С., Носковой Т.Н., Батаева А.В., Ульченко Е.Н., Лисицыной Л.С., Першина А.А. и других авторов.

Анализ научных исследований и практики применения новых информационных технологий в образовательном процессе показал, что компьютерные модели являются новым и эффективным средством обучения.

Хабибуллин Р. подчеркивает необходимость разработки современных интерактивных средств обучения. «Компьютерные модели как средство обучения могут и должны занять дидактически целесообразную «нишу» в учебном процессе. Речь идет, в первую очередь, о моделях физических явлений, недоступных непосредственному наблюдению учащимися. Компьютерное моделирование в этом случае дает учащимся один из важнейших инструментов, облегчающих понимание физических процессов и явлений» [34].

С точки зрения преподавателя очевидное достоинство компьютерного моделирования заключается в возможности создавать запоминающиеся зрительные образы. Такие образы способствуют пониманию сущности изучаемого физического явления и наиболее важных его особенностей. Моделирование позволяет привлечь внимание учащихся к существенным признакам изучаемых явлений, придать наглядность абстрактным концепциям и законам. Графическое отображение на экране компьютера физических объектов и процессов, эффекты анимации позволяют учащимся воспринимать большие объемы учебной информации, глубже и основательнее усваивать учебный материал [49].

Хабибуллин Р. приводит аргументы в пользу необходимости применения учебных компьютерных моделей в процессе обучения [34]. Интерактивные модели позволяют:

- визуализировать физические явления не только в виде привычных статистических иллюстраций, но и в динамике;
- создавать модели явлений природы, изучение которых в лабораторном эксперименте в условиях учебной аудитории невозможно в связи с необходимостью специального оборудования;

- значительно расширить спектр возможных условий наблюдения явления, а также исследовать его особенности протекания при изменении различных параметров в достаточно широких диапазонах;
- изучать опыты, включающие работу с материалами и устройствами, прямой контакт с которыми небезопасен или нежелателен (например, опыты с применением высоких напряжений и токов, высоких температур, кислотных и щелочных растворов, радиоактивных препаратов и др.);
- проводить сложные лабораторные и практические работы в условиях отсутствия в учебной аудитории соответствующей материально-технической базы (например, имитация реального физического эксперимента – симулятор);
- отрабатывать у учащихся отдельные экспериментальные умения в виде тренажеров.

Использование интерактивных моделей в учебном процессе обеспечивает активное восприятие нового учебного материала и мотивирует учащихся к его изучению. Работа с интерактивными учебными моделями повышает наглядность представления учебного материала и способствует более прочному его усвоению, а также позволяет преподавателю организовать новые, нетрадиционные виды учебной работы.

Компьютерные модели можно применять на любом этапе учебного занятия: при объяснении материала, его повторении и закреплении, а также на этапе контроля знаний и умений учащихся.

Исследованию различных аспектов разработки электронных учебных комплексов и материалов с использованием информационных и коммуникационных технологий посвящены работы многих зарубежных и отечественных исследователей (Соловов А.В., Плещинская И.Е., Галкина Л.А., Хабибуллин Р., Миллз Крис, Усмонов М.С., Василевский Ю.В., Алексеев Г.В., Бриденко И.И., Савельев А.О., Ларина Э.С., Джеймс Г., Уиттон Д., Хасты Т., Тибширани Р., Вильямс Г. и другие)

Анализ источников показал, что разработке интерактивных компьютерных моделей и web-приложений посвящены работы многих зарубежных авторов (Zev Ross, Garrett Grolemond, Joseph J. Allaire, Winston Chang и др.).

1.2 Обзор технологий и продуктов, поддерживающих систему электронного обучения

Ведущую роль в построении сетевого электронного учебно-методического комплекса играют технологии компьютерной визуализации информации. Это относится как к повышению наглядности структуры информации, включенной в электронный учебно-методический комплекс, мультимедийному воплощению его отдельных компонентов, так и к визуализации образовательных действий обучающегося. Широко используются интерактивные панорамы, хронологические шкалы, интеллект-карты, компьютерная анимация, аудио и видеофрагменты и многое другое. Высокий образовательный потенциал заключен в использовании технологий компьютерного моделирования, позволяющих не только демонстрировать, но и производить практические действия с виртуальными объектами. На основе компьютерного моделирования разрабатываются такие образовательные ресурсы как виртуальные демонстрации, экскурсии, лаборатории, практикумы [30].

Одной из современных технологий, позволяющей обеспечивать визуальное представление учебного материала, является метод интеллект-карт (Mind Maps) [7].

Метод интеллект-карт представляет собой подход для структурирования информации в наглядной форме и имеет ряд характерных особенностей:

- объект изучения сконцентрирован в центральном образе;
- основные темы и идеи, связанные с объектом изучения, расходятся от центрального образа в виде изогнутых ветвящихся линий;

- принимаются во внимание как логические, так и ассоциативные взаимосвязи предметной области;
- построения образуют связанную многоуровневую узловую структуру, в которой каждый узел представляет собой целенаправленный пункт новой системы ассоциаций.

Современные технические и программные средства помогают создавать компьютерные максимально приближенные к реальности модели объектов и процессов. Сочетание видео- и звуковых эффектов обеспечивает одновременное воздействие на два важнейших органа чувств человека: зрение и слух, что существенно повышает информативность учебного процесса и эффективность восприятия учебного материала. Комплекс средств воздействия (красочная графика, звуки, тексты, устная речь и ее интонации) обеспечивает многообразие ощущений у учащегося, которые анализируются, сравниваются, сопоставляются с уже имеющимися у него представлениями и понятиями [30].

Многие программы можно рассматривать как системы разработки контента для e-learning, включая Flash и PowerPoint. Однако, только маленькая группа программ включает поддержку стандартов контента для e-learning, таких как SCORM или AICC (CBT). Примеры: Articulate Storyline, Compositica, Adobe Authorware, Camtasia, eXe Learning.

Традиционные платформы электронного обучения представляют собой среды для предоставления учебных курсов и управления ими. Они предлагают пакеты инструментальных средств, поддерживающих создание онлайн-курсов, их обслуживание, средства регистрации студентов, администрирование процесса обучения и генерацию отчетов об успеваемости обучающихся.

SCORM (Sharable Content Object Reference Model) - сборник спецификаций и стандартов, содержащий требования к организации учебного материала, основанный на стандарте XML. Одна из базовых целей SCORM – составление электронных образовательных ресурсов из блоков учебного материала, называемых совместно используемыми объектами содержания, которые, в свою очередь, могут быть представлены в виде локального

фрагмента текста, графического объекта, компьютерной программы, видеоклипа, а также любого другого элемента гипермедиа или комбинации типовых объектов [31]. Данный стандарт позволяет обеспечить совместимость электронных образовательных ресурсов и возможность их многократного использования в образовательном процессе. Учебный материал представлен в виде небольших отдельных блоков. Данные блоки могут включаться в разные учебные курсы и использоваться в рамках различных систем дистанционного обучения независимо от того, какими средствами они были созданы.

Основными видами SCORM-объектов являются [1, 31]:

- цифровой актив. Представляет собой локальный цифровой ресурс, к которому можно отнести фрагмент текста, рисунок, анимация, аудио- или видеоклип и другие. Имеется возможность изменять и использовать для получения различных результатов по целевому назначению;
- объект содержания. Представляет собой цифровой актив или комбинацию цифровых активов, которыми могут быть, например, текст с изображением или компьютерная программа для расчета определенных параметров объекта исследования. Целевое назначение возможно изменять для получения различных желаемых дидактических результатов;
- учебный объект. Представляет собой блок содержания электронного обучения, предназначенного для использования многократно. Как правило, учебный объект имеет конкретную дидактическую цель и состоит из объектов содержания, например, модуля учебного курса, выполняющего определенную локальную дидактическую задачу.

Tin Can API представляет собой спецификацию программ в сфере дистанционного обучения, позволяющая обучающим системам общаться между собой путём отслеживания и записи учебных занятий всех видов. Информация об учебной деятельности сохраняется в специальную базу — хранилище учебных записей (Learning Record Store, LRS). Может являться как частью системы дистанционного обучения, так и быть самостоятельной системой.

К особенностям Tin Can API можно отнести:

- Tin Can API — улучшенная версия спецификации SCORM.
- Позволяет записывать любой опыт обучения, тем самым более полную картину обучения конкретного человека.
 - Снимает с данных ограничения, накладываемые системой дистанционного обучения.
 - Способен оказать помощь учебным отделам, сопоставляя данные о качестве выполнения работы с учебными данными, повышая эффективность обучения.

Tin Can API имеет ряд преимуществ по сравнению с SCORM.

1. Позволяет учитывать виды учебной активности, недоступные в SCORM: мобильное обучение, игры, симуляции, неформальное обучение, действия обучающихся в реальном мире.

2. Позволяет получать данные практически отовсюду: устройств, в которых соединение с интернет нестабильно или есть только в ограниченный период времени; с любых устройств; с любого внешнего сервера; из любых приложений, а не только из браузера, например, приложение для iPhone, эмулятор полета и другое.

Оболочки, спецификации и принципы организации определяют поуровневые подходы к созданию систем электронного обучения из наборов ранее определенных сервисов, определяют представления личной и групповой информации (IMS Enterprise), оценки (IMS Question and Test Interface), группировки изучаемого контента (IMS Content Package и SCORM), профиля студента и истории его обучения (IMS Learner Information Package and ePortfolio), динамического программирования контента (IMS Simple Sequencing), поиска в федеративных базах данных (IMS Digital Repositories Interoperability), компетентности учащегося (IMS Reusable Definition for Competence and Educational Objectives), и связывания различных инструментов электронного обучения. На низком уровне эти стандарты и спецификации

описывают синтаксис, который различные сервисы должны реализовать для внешнего представления информации.

Сервисы электронного обучения могут покрывать любой технический аспект дистанционного обучения. Например, такие инициативы, как ELF, IMS Abstract Framework OKI определяют широкий спектр высокоуровневой функциональности, которые обеспечивают сервисы электронного обучения, которые в свою очередь включают в себя традиционную функциональность (аутентификация, слежение, управление курсами, расписание, операции, инструменты и оценка), а также новые возможности, такие как презентации, сбор ресурсов, управление контекстом, моделирование, игры, wiki, блоги и многое другое.

Таким образом, можно сделать вывод о том, что для обеспечения гибкости платформа электронного обучения должна поддерживать постоянно расширяющийся спектр высокоуровневых и новых низкоуровневых сервисов.

Дистанционная система подготовки специалистов основывается на применении систем:

- управления обучением (Learning Management System – LMS),
- управления контентом (Content Management System – CMS),
- управления учебным контентом (Learning Content Management System – LCMS).

Развертывать CMS-LCMS систему дистанционного обучения можно как на локальном сервере с предоставлением к нему удаленного доступа, так и с использованием для размещения контента некоторого удаленного хостинга или облачного хранилища. Преимущества применения облачного хранилища зависят от конкретной выбранной сервисной модели.

Существует несколько подходов к организации облачных сервисов.

Программное обеспечение как сервис (Software as a Service (SaaS) - программное обеспечение по требованию) представляет собой модель удаленного использования программного обеспечения, в которой приложение размещается у провайдера или в облаке.

Основное преимущество модели SaaS для потребителя услуги состоит в отсутствии затрат, связанных с установкой, обновлением и поддержкой работоспособности оборудования и работающего на нём программного обеспечения, он лишь использует приложения, которые расположены в облаке. Пользователи получают доступ к готовым приложениям через веб-браузер с любого мобильного устройства.

Платформа как сервис (Platform as a Service (PaaS)) – модель, представляющая возможность удаленного использования операционной системы и сервисов, которая позволяет создавать и разворачивать собственные сервисы на базе арендуемой платформы. Модель PaaS предоставляет максимальную масштабируемость сервисов. Пользователи получают доступ к созданному для них удаленному виртуальному компьютеру с установленной операционной системой, а также доступ к данной операционной системе как к локальной.

Инфраструктура как сервис (Infrastructure as a Service, IaaS) – модель, предоставляющая возможность логического расширения информационного пространства предприятия за счет аренды аппаратных ресурсов, включающих серверы и устройства хранения данных. При этом потребитель имеет полную свободу в установке требуемого программного обеспечения и сервисов. Модель инфраструктуры как сервис предоставляет значительно большие возможности по настройке отдельных компонентов, чем модели PaaS и SaaS: пользователи получают доступ к созданному для них удаленному виртуальному компьютеру с установленной операционной системой и предустановленным программным обеспечением.

Рабочий стол как сервис (Desktop as a Service (DaaS)), когда в качестве услуги пользователь получает не конкретное приложение, а полностью готовое рабочее место с необходимым набором программного обеспечения. Такой рабочий стол будет доступен с любого устройства, подключенного к локальной сети или Интернету.

База данных как сервис (Database as a Service)), модель, основанная на SaaS, в которой потребителю предоставляется база данных требуемой конфигурации.

Для реализации системы дистанционного обучения больше всего подходят облака, предоставляющие SaaS сервис, на базе которого можно развернуть LMS-CMS-LCMS систему.

Все существующие программные решения можно разделить на две категории:

- бесплатные программы с открытым исходным кодом (Open Source),
- коммерческие продукты.

Коммерческие продукты являются надежными, но нет возможности переделать систему для своих требований.

Бесплатные программные оболочки позволяют доработать систему под функционал организации. Наличие огромного количества бесплатных дополнительных модулей и плагинов существенно экономят время на разработку системы дистанционного обучения, а также уменьшают затраты на ее сопровождение.

К LMS и LCMS системам, получившим наибольшее распространение можно отнести следующие системы.

Blackboard – одна из наиболее популярных LMS в мире, особенно в системе высшего образования, представляющая собой виртуальную среду обучения на базе серверного программного обеспечения. Характерными чертами этой программы являются: удобное управление, настраиваемая панель инструментов, расширяемый дизайн, позволяющий объединение всех студенческих информационных систем и аутентификации протоколов. Основные задачи: добавление онлайн элементов на курсы, которые обычно встречаются только на лекции, и разработка полностью интерактивных курсов с небольшим количеством встреч.

Sakai – свободно распространяемый программный продукт с полностью открытым исходным кодом, основанный на технологии Java. Развитием проекта

занимается Sakai Community – сообщество разработчиков, в которое входят использующие его университеты и ряд коммерческих фирм. Данное программное обеспечение включает в себя множество возможностей по разработке курсов и организации системы управления курсами, а также управление прохождением документов, форумы, чаты, онлайн-тестирование и т.п. Состоит из отдельных модулей, которые можно собирать как конструктор с возможностью добавления недостающих элементов.

Moodle (модульная объектно-ориентированная динамическая учебная среда) – свободная система управления обучением, распространяющаяся по лицензии GNU General Public License. Moodle – среда дистанционного обучения с открытым исходным кодом, ориентированная на организацию взаимодействия между преподавателем и учащимися. Также подходит для организации традиционных дистанционных курсов и поддержки очного обучения. Moodle переведена на десятки языков, в числе и русский и используется в 197 странах мира и позволяет эффективно организовать процесс обучения, используя такие возможности как проведение семинаров, тестов, заполнение электронных журналов.

Adobe Captivate 5 – программный продукт, являющийся удобным средством создания и публикации материалов, основанный на технологии Flash. Предоставляет широкий спектр таких возможностей, как создание учебных материалов на основе презентаций Microsoft PowerPoint, захват изображения с экрана компьютера, создание различных тестовых заданий с возможностью перехода к вопросу в зависимости от ответа обучающегося.

В обучающие материалы могут быть встроены интерактивные элементы, например такие, как поля ввода текстовых данных, опросы с возможностью выбора правильного варианта ответа и другие. Компактные размеры и высокое разрешение файлов данного программного продукта позволяют широко использовать его для приобретения навыков работы с приложением, предоставления пользователю справочной информации, а также демонстрации возможностей новых продуктов.

Несмотря на рассмотренные выше возможности Adobe Captivate, следует отметить, что эта технология является проприетарной, закрытой и не поддерживается на планшетных устройствах и смартфонах. Кроме того, эта технология является чрезвычайно требовательной к вычислительным ресурсам.

Articulate – набор программных продуктов для создания учебных материалов, основанных на Flash-технологии. Состоит из продуктов, предназначенных для конвертации презентаций MS Power Point во Flash, создания интерактивных Flash-слайдов, а также различных тестовых заданий, монтажа видео и многого другого.

Каждый из продуктов функционирует в виде отдельных приложений и позволяет комбинировать материалы, созданные в различных продуктах. С целью облегчения базовых действий по конвертации средство для конвертации презентаций встраивается непосредственно в ленту MS PowerPoint. Для публикации разработанных материалов поддерживаются стандарты SCORM.

iBooksAuthor –инструмент для создания offline-учебников, бесплатно доступное в App Store для Mac и позволяющее создать электронные книги с технологией Multi-Touch и любые другие книги специально для iPad. Данный пакет в настоящее время можно считать стандартом для разработки учебных материалов для планшетных компьютеров, полностью соответствующим технологии создания открытых электронных образовательных ресурсов. Позволяет создавать offline-учебники для устройств от компании Apple, чтение которых происходит через программу iBooks, а загрузка доступна через iTunes, а также позволяет создавать электронные книги с планшетного компьютера, сопровождать текст видеороликами, галереями изображений и различными интерактивными элементами, такими как диаграммы, 3D анимация и т.д.

Существует ряд бесплатных русифицированных сред разработки электронных учебников.

eAuthor СВТ – достаточно функциональное средство разработки учебных курсов, разработанное компанией ГиперМетод [39]. Позволяет разрабатывать курсы с различным медиа-контентом, включая такие файлы, как SWF, AVI,

MPEG, MP3, DWF, VRML, также позволяет добавлять в курсы различные виды тестов и разрабатывать функциональные тренажеры и практикумы.

Структура будущего курса зависит от встроенных в редактор нескольких шаблонов. При создании нового курса программа сама предлагает выбрать, каким будет курс: Учебный курс, Интерактивное руководство, Учебно-методический комплекс, Слайд-курс, Тренинг, Конструктор УМК или Конструктор упражнений. В целом, редактор eAuthor CBT создан для разработки достаточно профессиональных слайдовых учебных курсов.

LCDS – система для создания учебных материалов от Microsoft. С помощью этого программного обеспечения можно создавать простые электронные курсы и публиковать их в формате SCORM. Система не является официальным продуктом Microsoft, поэтому лишена технической поддержки. Зато она бесплатна и создана специально для участников сообщества Microsoft Learning. Чтобы скачать эту программу, нужно быть участником сообщества, что легко реализуется обычной регистрацией на сайте сообщества. В системе LCDS можно создавать курсы с различными медиа-объектами (фото, видео), а также с определенными встроенными шаблонными интерактивностями и проверочными заданиями. Программа предназначена для создания простых линейных курсов и справочных материалов и не предназначена для создания сложных нестандартных курсов со сложными сценариями и индивидуальным дизайном.

Недостатки программы:

- возможность делать курсы только по готовым встроенным шаблонам;
- справка на английском языке.

Достоинства программы:

- очень понятный и простой интерфейс;
- имеется русификатор (кроме справки);
- поддерживает работу с различными медиа-объектами.

Udutu – это средство разработки электронных курсов через браузер. В результате получается SCORM-пакет, который можно сохранить на

компьютере, а в последствие загрузить в любую LMS, поддерживающую стандарт SCORM. Позволяет конструировать курсы, поддерживающие аудио и видео, большой выбор тестовых заданий, настройки анимации и др, а также совместно работать над одним проектом нескольким экспертам одновременно.

К недостаткам можно отнести нерусифицированный интерфейс и документацию. При разработке учебных курсов кириллица поддерживается, что делает этот инструмент востребованным и при создании русскоязычных курсов.

Несмотря на свои достоинства имеет также ряд недостатков. С помощью сервиса Udutu можно сделать достаточно красочные слайдовые курсы, но в рамках определенных шаблонов, хоть и достаточно гибко настраиваемых. Таким образом, данный инструмент не подходит для разработки уникального дизайна или нестандартного курса.

CourseLab – один из самых популярных отечественных редакторов электронных курсов, позволяющий создавать, как простые слайдовые курсы, так и очень сложные курсы с нелинейным сценарием, вплоть до создания полноценных видеоигр. Редактор CourseLab достаточно функционален и одновременно прост в использовании [43].

Бесплатная версия CourseLab полностью функциональна, не имеет никаких ограничений по времени или количеству контента. Однако бесплатность влечет за собой и многие неудобства: 1) версия полностью англоязычна; 2) она несколько устарела; 3) эта версия не является кросс-браузерной, и поддерживает работу курсов только в браузере Internet Explorer; 4) пользователи этой версии лишены возможности какой-либо поддержки.

eXe learning xhtml editor является бесплатно распространяемым программным средством для создания материалов электронных курсов, позволяющее создавать учебные материалы, состоящие из текстовых материалов, Java-апплетов, импортировать материалы внешних веб-сайтов и упаковывать учебные материалы в стандарте SCORM 1.2. Дает возможность

добавлять тестовые задания разного типа, включая вопросы с открытым ответом, для проверки преподавателем [57].

Программа обладает простым интерфейсом и не требовательна к аппаратным ресурсам компьютера, имеет вариант, не требующий установки, который может загружаться с Flash-носителя. Материалы, разрабатываемые с помощью eXe, основываются на технологиях html и Java-script. XHTML-редактор eXe Learning позволяет преподавателям самостоятельно создавать и публиковать учебные материалы на web-основе без знания HTML или разработки сложных web-приложений.

Среда разработки электронных учебников eXe имеет ряд преимуществ:

- является простым и интуитивно понятным средством, позволяющим преподавателям создавать для целей обучения профессионально выглядящие web-страницы;

- современные системы управления обучением LMS не предоставляют пользователям средств создания автономных web-курсов. Электронный web-учебник, созданный в eXe легко импортируется в любые LMS, поддерживающие SCORM- или IMS-стандарты;

- большинство современных LMS используют модель централизованного web-сервера, требуют регистрации и сильно «тормозят» при работе. Среда разработки eXe разработана как автономное средство и лишена этого недостатка;

- встроенный в eXe редактор WYSIWYG поддерживает множество функций создания и форматирования учебного контента, главной из которых для нас является эффективный импорт в eXe документов редактора Word.

Shiny – это пакет R, который позволяет легко строить интерактивные web-приложения прямо из R. R - язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом [44, 52].

Shiny дает возможность пользователям R создавать онлайн приложения без глубокого знания HTML + CSS и JavaScript.

Применение фреймворка Shiny для разработки интерактивных веб-приложений на основе R имеет ряд преимуществ:

- применение R позволяет быстро и достаточно просто производить статистическую обработку данных;
- R – программное обеспечение с открытым кодом, что позволяет использовать его свободно и бесплатно;
- R – является кроссплатформенным программным обеспечением;
- R используется более чем 2 миллионами специалистов по всему миру;
- R активно развивается и на данный момент разработано уже более 9 тысяч библиотек;
- R работает с различными системами графики.

Наряду с многочисленными достоинствами R так же имеет ряд недостатков:

- R – программное обеспечение с открытым кодом под копилефтной лицензией GNU GPL 2;
- R имеет ряд проблем, связанных с использованием оперативной памяти;
- отсутствие GUI, достаточно исчерпывающей документации и т. д.

Проведенные исследования позволили сформулировать **требования к технологии разработки:**

1. Среда разработки должна быть мультимедийной, то есть должна предоставлять возможность работы с выборками данных, рисунками, графиками, аудио- и видеоинформацией.

2. Учебный web-компонент должен быть интерактивным по принципу работы электронных таблиц, то есть любое действие пользователя должно приводить к автоматическому обновлению учебного контента.

3. Технология должна обеспечивать разработку сетевых ресурсов, распространяемых через интернет.

4. Среда разработки должна быть бесплатной или с минимальной стоимостью.

Таким образом, наиболее эффективной средой разработки интерактивных учебных web-компонентов является использование пакета Shiny из бесплатной среды анализа данных R.

Вывод по первой главе

В ходе работы над первой главой была определена актуальность исследования, которая обусловлена необходимостью использования в учебных комплексах интерактивных элементов, предоставляющих возможность обработки больших массивов данных, моделирования процессов и явлений по дисциплинам естественно-научного профиля, проведения серии модельных экспериментов, визуализации результатов на экране компьютера.

Использование нестандартных интерактивных средств обучения в вузе позволяет решать наиболее важные проблемы - вовлечение студента в образовательный процесс, отсутствие необходимости приобретения дорогостоящего оборудования и программного обеспечения.

Глава 2 ТЕХНОЛОГИЯ РАЗРАБОТКИ ИНТЕРАКТИВНЫХ УЧЕБНЫХ WEB-ПРИЛОЖЕНИЙ SHINY

2.1 Структура приложения Shiny

Клиент-серверная архитектура типового web-приложения Shiny представлена на рисунке 2.1.

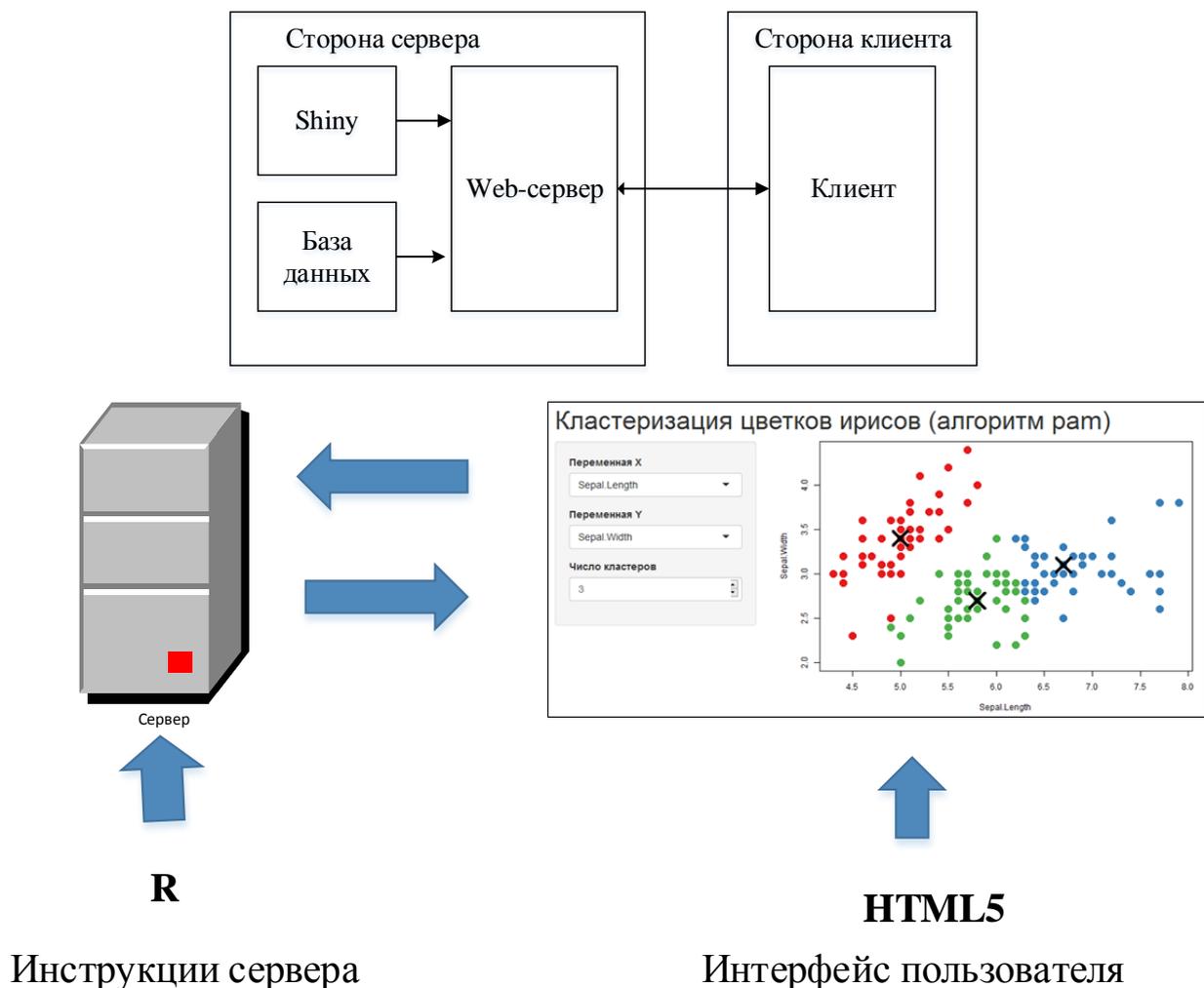


Рисунок 2.1 - Структура web-приложения Shiny

Браузер посылает запрос веб-серверу, в котором указывает какой файл он запрашивает (html или сценарий): если это html-страница, то он находит эту html-страницу и присылает клиенту, если же это приложение Shiny, то он выполняет скрипты этого приложения, при необходимости обращаясь к R, затем сервер генерирует веб-страницу и присылает ее клиенту.

Приложения Shiny состоят из двух компонентов [54]:

- скрипт пользовательского интерфейса `ui.R`, который контролирует раскладку и вид `web`-страницы;
- серверный скрипт `server.R` содержит команды, нужные для построения интерактивных объектов приложения.

Минимальный исходный код `R` для каждого скрипта представлен на рисунке 2.2.

ui.R:

```
library(shiny)
shinyUI(pageWithSidebar(
  headerPanel("Template"),
  sidebarPanel(),
  mainPanel()
))
```

server.R:

```
library(shiny)
shinyServer(function(input, output) {
})
```

Рисунок 2.2 - Минимальный исходный код `R` для каждого скрипта

позволяет создать работающее приложение `Shiny` с пустым пользовательским интерфейсом.

Созданные скрипты помещаются в отдельный каталог и запускаются на выполнение командой:

```
> runApp("~/shinyapp")
```

например:

```
> runApp("D:/My_R/Shiny/template")
```

Три функции `headerPanel()`, `sidebarPanel()` и `mainPanel()` определяют разные области пользовательского интерфейса `web`-страницы. Приложение называется “`Template`”, что отражено в его заголовке

headerPanel и в названии каталога команды runApp. Другие панели пока пустые.

Начиная с версии 0.10.2, Shiny поддерживает однофайловые приложения, т.е. можно создать один R-файл, который включает оба скрипта приложения [56].

Программный код однофайлового приложения представлен на рисунке 2.3. Пример однофайлового приложения представлен на рисунке 2.4.

```
library (shiny)
## server.R
server <- shinyServer(function(input, output) {
  output$distPlot <- renderPlot ({
    x <- iris[, 3]
    bins <- seq(min(x), max(x), length.out =
input$bins + 1)
    hist (x, breaks = bins, col = 'darkgray', border =
'white')
  })
})
## ui.R
ui <- fluidPage (
  titlePanel ("Гистограмма"),
  sidebarLayout (
    sidebarPanel (
      sliderInput ("bins",
                  "Number of bins: ",
                  min = 1,
                  max = 50,
                  value = 30)
    ),
    mainPanel (
      plotOutput ("distPlot")
    )
  )
)
shinyApp (ui=ui, server=server)
```

Рисунок 2.3 – Программный код однофайлового приложения

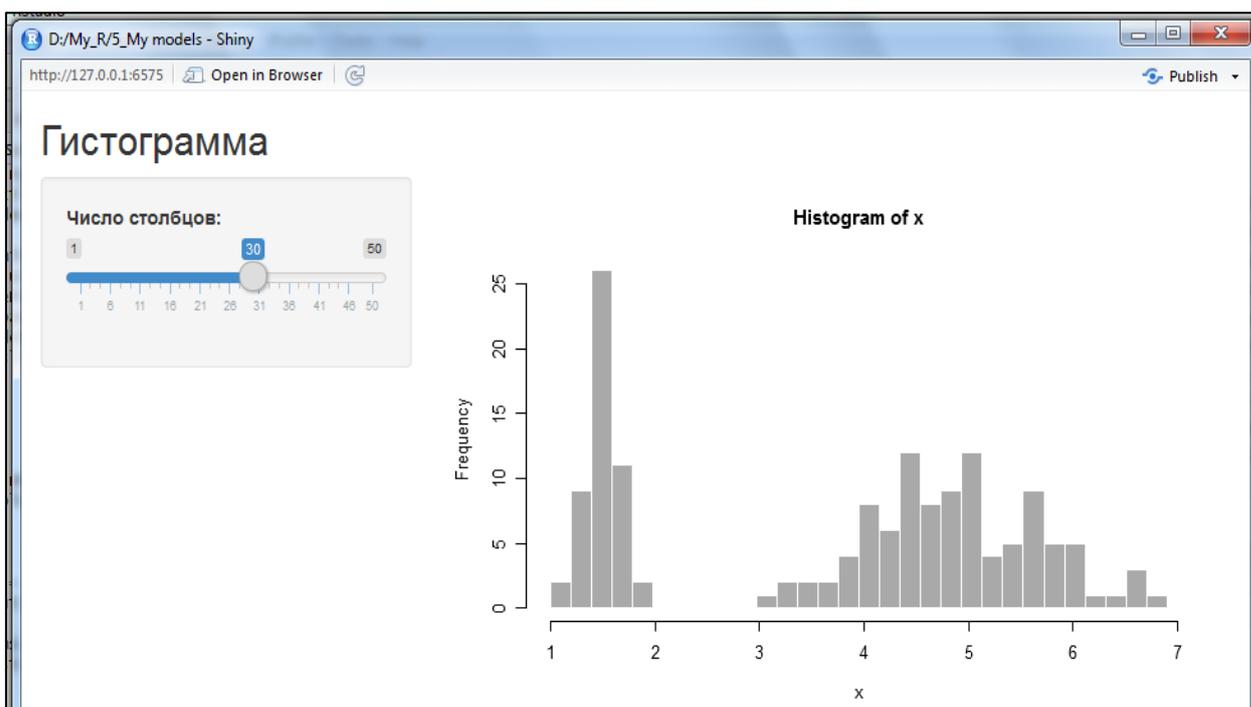


Рисунок 2.4 – Пример вывода гистограммы

Использование в скрипте `ui.R` функции `fluidPage()`, позволяет создавать окно приложения, автоматически приспособивающееся к размерам окна браузера пользователя.

Shiny Server - это веб-сервер, специально разработанный для размещения приложений Shiny, позволяющий размещать свои приложения в контролируемой среде, поэтому разработанное приложение Shiny (и любые необходимые ему данные) всегда будут под контролем.

Shiny Server размещает каждое приложение на своем собственном веб-адресе и автоматически запускает приложение, когда пользователь вводит этот адрес. Когда пользователь уходит с web-страницы, Shiny Server автоматически остановит приложение.

2.2 Разметка HTML-страниц и каскадные таблицы стилей в приложении Shiny

Чтобы сформировать актуальный контент в любой панели web-страницы, используются R-функции Shiny - аналоги тегов HTML5 [48]. Основные R-функции-аналоги тегов HTML представлены в таблице 2.1.

Таблица 2.1 – Основные R-функции-аналоги тегов HTML

Функция Shiny	Тег HTML	Формирует
p	<p>	Абзац текста
h1 – h6	<h1> - <h6>	Заголовки от 1 до 6 уровня
a	<a>	Гиперссылка
br	 	Перенос строки
div	<div>	Абзац текста с уникальным стилем
span		Часть текста в строке с уникальным стилем
pre	<pre>	Текст с фиксированной шириной символов
code	<code>	Форматированный блок кода
img		Изображение
strong		Жирный текст
em		Наклонный текст
HTML		Обрабатывает строку символов как HTML код

Имена других тегов конфликтуют с именами родных функций R, поэтому их нужно вызывать с помощью R-функции `tags$code`. В таблице 2.2 представлены R-функции объекта `shiny::tags`.

Таблица 2.2 - R-функции объекта shiny::tags

[1]	“a”	“abbr”	“address”	“area”	“article”
[6]	“aside”	“audio”	“b”	“base”	“bdi”
[11]	“bdo”	“blockquote”	“body”	“br”	“button”
[16]	“canvas”	“caption”	“cite”	“code”	“col”
[21]	“colgroup”	“command”	“data”	“datalist”	“dd”
[26]	“del”	“details”	“dfn”	“div”	“dl”
[31]	“dt”	“em”	“embed”	“eventsourse”	“fieldset”
[36]	“figcaption”	“figure”	“footer”	“form”	“h1”
[41]	“h2”	“h3”	“h4”	“h5”	“h6”
[46]	“head”	“header”	“hgroup”	“hr”	“html”
[51]	“i”	“iframe”	“img”	“input”	“ins”
[56]	“kdb”	“keygen”	“label”	“legend”	“li”
[61]	“link”	“mark”	“map”	“menu”	“meta”
[66]	“meter”	“nav”	“noscript”	“object”	“ol”
[71]	“optgroup”	“option”	“output”	“p”	“param”
[76]	“pre”	“progress”	“q”	“ruby”	“rp”
[81]	“rt”	“s”	“samp”	“script”	“section”
[86]	“select”	“small”	“source”	“span”	“strong”
[91]	“style”	“sub”	“summary”	“sub”	“table”
[96]	“tbody”	“td”	“textarea”	“tfoot”	“th”
[101]	“thead”	“time”	“title”	“tr”	“track”
[106]	“u”	“ul”	“var”	“video”	“wbr”

Форматирование web-страницы можно осуществить с помощью каскадных таблиц стилей CSS.

Рекомендации по использованию CSS для форматирования web-страницы:

- Создайте подкаталог `www` в каталоге приложения Shiny. Этот подкаталог назван `www` специально: Shiny делает каждый файл в `www` доступным браузеру пользователя. Подкаталог `www` – это место для размещения CSS-файлов, рисунков и других ресурсов web-приложения Shiny.
- Поместите CSS-файл в подкаталог `www` приложения. В примере ниже используется файл `bootstrap.css`, скачанный с сайта [<http://bootswatch.com/>].
- Добавьте атрибут `theme` в файл `ui.R`:

Программный код использования CSS-файла для оформления web-страницы представлен в листинге 2.1.

Листинг 2.1. Программный код использования CSS-файла для оформления web-страницы

```
# ui.R
shinyUI(fluidPage(theme = "bootstrap.css",
  headerPanel("New Application"),
  sidebarPanel(
    sliderInput("obs", "Number of observations:",
      min = 1, max = 1000, value = 500)
  ),
  mainPanel(plotOutput("distPlot"))
))

# server.R
library(shiny)
shinyServer(function(input, output) {
  # сюда вводить ваш код R
})
```

Результат работы программного кода представлен на рисунке 2.5.

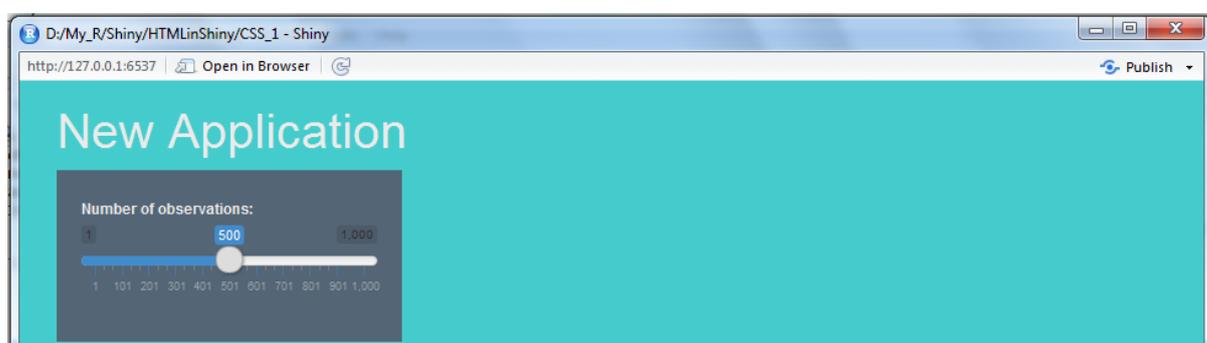


Рисунок 2.5 – Пример использования CSS-файла для оформления web-страницы

В большинстве случаев лучший способ создать UI-интерфейс приложения – это написать код R, используя функции `fluidPage()`, `div()` и др.

Однако, иногда требуется интегрировать Shiny с существующим HTML. Это можно сделать с помощью шаблонов HTML. Шаблоны можно использовать для генерации целых web-страниц, а также для генерации HTML компонентов, включаемых в приложение Shiny.

Чтобы использовать шаблон HTML для UI, сначала необходимо создать HTML – файл в той же папке, в которой находится приложение.

Шаблон представляет собой чистый HTML-код, кроме фрагментов в скобках `{{ }}`, содержащих любой код R, который обрабатывается во время обработки шаблона.

Таким образом, при создании контента собственного пользовательского приложения существует возможность:

- создавать пользовательский интерфейс с помощью функций `fluidPage`, `titlePanel` и `sidebarLayout`;
- добавлять на панели `sidebarPanel` или `mainPanel` виджеты, теги, HTML-код и реактивные выходные объекты;
- добавлять изображения, разместив их в каталоге `www` приложения Shiny.

2.3 Раскладка web-страницы в приложении Shiny

Shiny включает ряд возможностей для раскладки компонентов web-страницы [50]:

- раскладка по умолчанию с боковой панелью виджетов и большой главной областью для вывода;
- настраиваемые раскладки `Grid`, использующие систему сеток компоновки;
- сегментирование раскладок с помощью функций `tabsetPanel()` и `navlistPanel()`;
- создание приложений с многоуровневыми компонентами с помощью функции `navbarPage()`.

Все они основаны на раскладках Bootstrap 2 -чрезвычайно популярного фреймворка HTML/CSS.

Раскладка sidebarPanel используется по умолчанию и выглядит, как показано на рис.2.6.

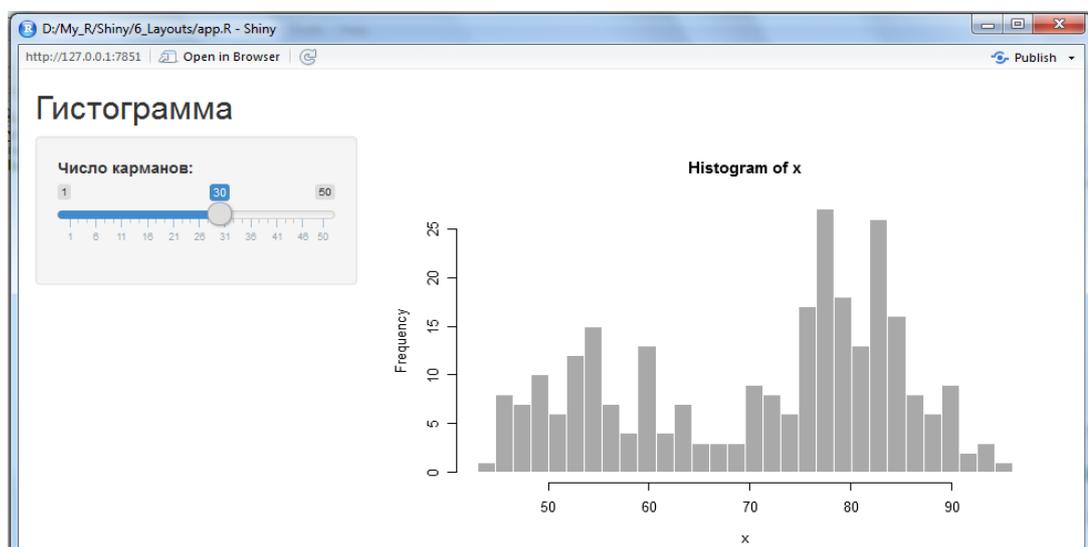


Рисунок 2.6 – Пример использования раскладки Grid

Функции `titlePanel()` и `sidebarLayout()` являются двумя наиболее популярными элементами в `fluidPage()`. Они создают основу Shiny приложения с боковой панелью. Функция `sidebarLayout()` всегда имеет два аргумента: функции `sidebarPanel()` и `mainPanel()`, которые позволяют разместить контент в соответствующей панели. По умолчанию боковая панель находится с левой стороны созданного приложения, можно также переместить ее в правую сторону, дополнив функцию `sidebarLayout()` дополнительным параметром `position = "right"`.

Раскладка Grid позволяет очень гибко размечать HTML-страницу путем разбиения ее на строки и столбцы с помощью функций `fluidRow()` и `column()`. Ширины столбцов основываются на системе Bootstrap, использующей контейнер из 12 столбцов в строке (рис.2.7).

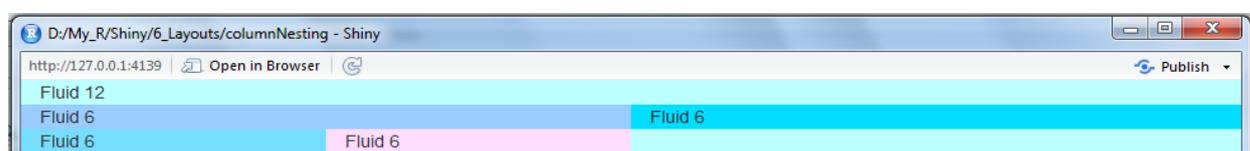


Рисунок 2.7 – Пример использования вложенных столбцов

Раскладка Tabset позволяет создавать на HTML-странице несколько закладок, как показано на рис.2.8. Закладки могут быть расположены вверху (умолчание), внизу, слева или справа контента.

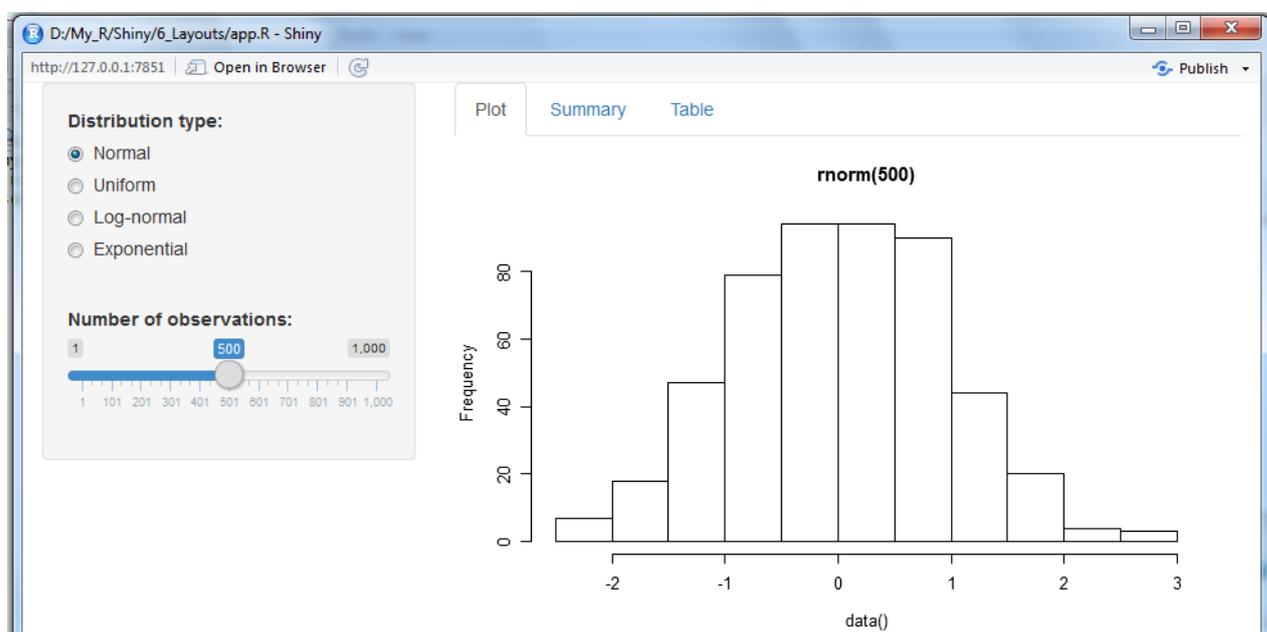


Рисунок 2.8 – Пример использования закладок Tabset

Раскладка Secondary Navigation позволяет создавать с помощью функции navbarPage() закладки с выпадающим списком - меню (рис.2.9).

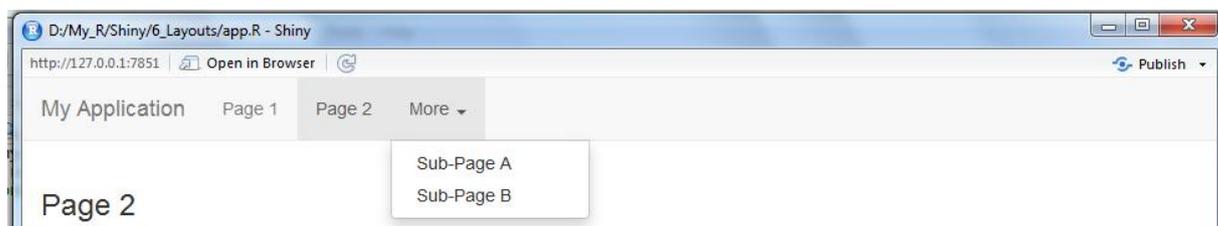


Рисунок 2.9 – Пример использования раскладки Secondary Navigation

У функции navbarPage() есть несколько дополнительных аргументов:

- header - тег или список тегов для общего заголовка над всеми панелями;
- footer - тег или список тегов для общего подвала под всеми панелями;
- inverse – инвертирование фона и текста навигационной панели: True - темный фон и светлый текст, False – светлый фон и темный текст.

- collapsable - автоматическое сворачивание навигационной панели в меню, если ширина браузера меньше 940 px.

Есть два типа раскладок Bootstrap: масштабируемые (fluid) и фиксированные (fixed).

Раскладка fluid всегда занимает всю ширину web-страницы и изменяет размеры ее компонентов автоматически, если изменяется размер страницы. Раскладка fixed имеет по умолчанию фиксированную ширину 940 px, но по желанию пользователя ширину решетки можно изменить.

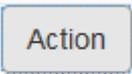
2.4 Компоненты ввода и вывода в приложении Shiny

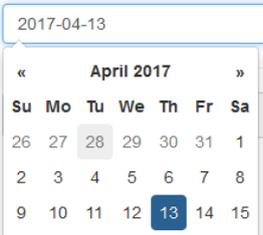
Для обеспечения интерактивного взаимодействия пользователя с Shiny приложением предназначены виджеты. Виджеты - это интерактивные веб-элементы, которые пользователь может использовать для управления приложением [53].

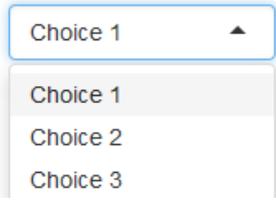
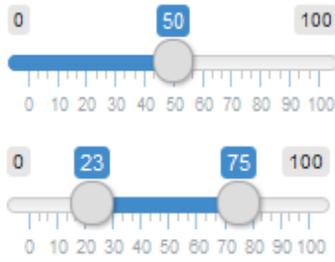
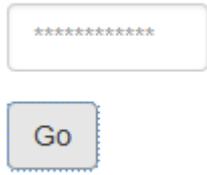
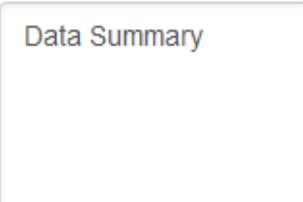
В пакет Shiny включен большой набор готовых виджетов, каждому из которых соответствует функция R с «понятным» названием. Например, функция с именем `actionButton()`, создает кнопку действия, а `dateInput()` – текстовое поле для задания даты.

В таблице 2.3 перечислены функции создания стандартных виджетов управления с помощью коллекции функций `*Input`.

Таблица 2.3 – Основные виджеты управления

Виджет	Вид	Описание
<code>actionButton</code>		Создает кнопку, первоначальное значение которой равно 0. Значение инкрементируется на 1 при каждом нажатии кнопки
<code>checkboxInput</code>		Создает флажок, посылающий серверу значение TRUE, если флажок установлен, иначе посылается FALSE.

Виджет	Вид	Описание
checkboxGroupInput	<input checked="" type="checkbox"/> Choice 1 <input type="checkbox"/> Choice 2 <input type="checkbox"/> Choice 3	Создает группу флажков. Посылает серверу символьный вектор, содержащий значения выбранных флажков.
dateInput		Создает текстовое поле, щелчком по которому пользователь получает доступ к календарю и может выбрать дату. Посылает результат серверу как объект Date.
dateRangeInput		Создает два текстовых поля, щелчком по которым пользователь получает доступ к календарям, чтобы выбрать диапазон дат. Посылает результат серверу в виде вектора с двумя объектами Date.
fileInput		Создает менеджер загрузки на сервер одного и более файлов сразу. Виджет передает серверу фрейм данных, содержащий одну строку для каждого загружаемого и 4 столбца (Name, Size, type, datapath).
numericInput		Создает список для ввода числовых величин. Можно ввести число или выбирать из списка заданных значений. Серверу посылается как число типа double.
radioButtons	<input checked="" type="radio"/> Выбор 1 <input type="radio"/> Выбор 2 <input type="radio"/> Выбор 3	Создает набор радиокнопок, возвращает значение выбранной радиокнопки в виде символьной строки.

Виджет	Вид	Описание
selectInput		Создает выпадающий список для выбора одного и больше его элементов. Виджет передает значение выбранных элементов серверу в виде вектора символьных строк.
sliderInput		Создает ползунок для выбора одного или двух значений. Передает выбранные значения серверу в виде вектора из 2-х чисел. Slider Bar и Slider Range используют одну и ту же функцию sliderInput.
textInput		Создает текстовое поле ввода, передается серверу как символьная строка.
passwordInput		Создает элемент ввода паролей.
textAreaInput		Создает текстовую область (textarea) для ввода неструктурированного текста.

Добавить виджеты на веб-страницу можно так же, как добавляются другие виды контента HTML.

Каждая функция *Input принимает несколько аргументов. Первые два аргумента для каждого виджета одинаковые:

1. *Имя* (идентификатор Id) виджета используется для доступа к его значению и прозрачно для пользователя. Имя должно быть символьной строкой.

2. *Метка* (надпись label) виджета, показывается рядом с виджетами в приложении. Это должна быть символьная строка, которая может быть пустой строкой "".

Синтаксис функции ввода представлен на рисунке 2.10.

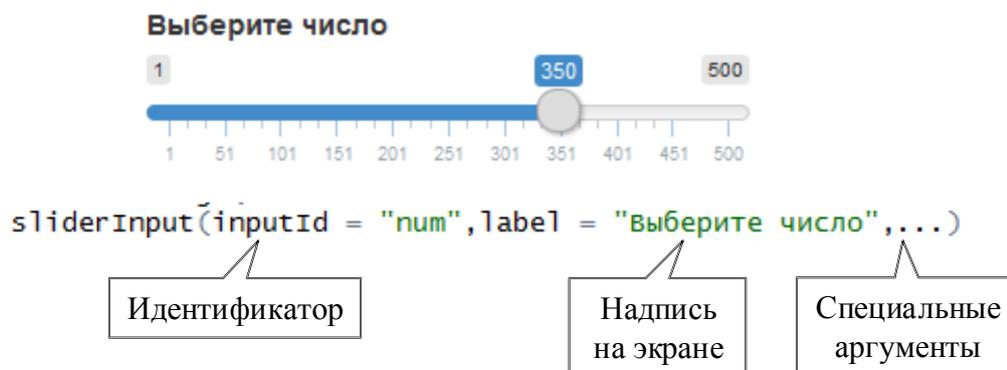


Рисунок 2.10 – Синтаксис функции *Input ()

Остальные аргументы варьируются в зависимости от назначения виджета.

Пример использования виджета:

```
sliderInput(inputId = "num",  
            label = "Выберите число",  
            value = 350, min = 1, max = 500  
)
```

В результате выполнения данного кода получим следующий результат (рис.2.11).

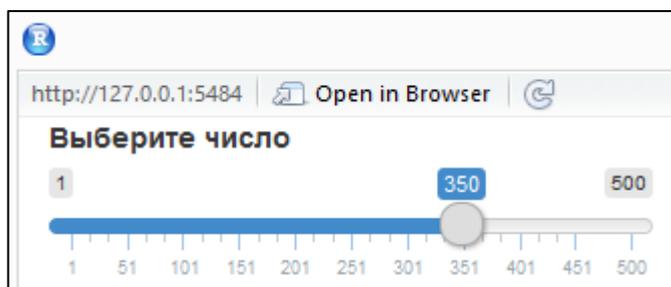


Рисунок 2.11 - Пример использования виджета sliderInput

Интерактивный вывод является автоматической реакцией на взаимодействие пользователя с виджетами. Интерактивный вывод можно создать за 2 шага:

1. Добавить объект *Output в пользовательский интерфейс ui.R.
2. Сообщить Shiny, как построить объект *Output в server.R. Этот объект будет интерактивным, если его код использует значение виджета.

Shiny предоставляет семейство функций, которые превращают объект в интерактивный вывод для пользовательского интерфейса. Каждая функция создает специфический тип вывода (табл. 2.4).

Таблица 2.4 - Семейство функций *Output

Функция	Тип вывода
dataTableOutput()	интерактивная таблица
htmlOutput()	HTML-код
imageOutput()	изображение
plotOutput()	график
tableOutput()	таблица
textOutput()	текст
uiOutput()	элемент Shiny UI
verbatimTextOutput()	текст

Интерактивный вывод нужно добавить в пользовательский интерфейс UI, поместив выходную функцию внутри панели sidebarPanel или mainPanel сценария ui.R.

Например, в следующем примере файл ui.R содержит функцию textOutput(), которая добавляет интерактивную строку текста в главную панель приложения Shiny (Листинг 2.2).

Листинг 2.2. Пример использования функции textOutput()

```
library (shiny)
# server.R
server <- function(input,output) {}
# ui.R
ui <- shinyUI (fluidPage (
  titlePanel ("censusVis"),
```

```

sidebarLayout (
  sidebarPanel (
    helpText ("Создает демографическую карту США"),

    selectInput ("var",
      label = "Выбрать переменную для вывода",
      choices = c ("Процент белых", "Процент черных",
        "Процент испанцев", "Процент азиатов"),
      selected = "Процент белых"),

    sliderInput ("range",
      label = "Интересующий диапазон:",
      min = 0, max = 100, value = c(0, 100))
  ),

  mainPanel (
    # textOutput ("text1")
  )
)
))
shinyApp (server=server, ui=ui)

```

Каждая из функций вывода `*Output()` принимает один аргумент - символьную строку, которую Shiny будет использовать в качестве идентификатора пользовательского интерактивного элемента.

Положение функции вывода в `ui.R` говорит Shiny, где на экране отобразить нужный объект. Этот объект строится с помощью неименованной функции `function(input, output)` в скрипте `server.R`. Неименованная функция в нем играет специальную роль в Shiny. Она строит похожий на список объект `output`, который содержит код, который требуется для обновления объектов R в приложении.

Имя объекта должно согласовываться с именем `text1` интерактивного элемента, который создали в `ui.R`:

```
server <- shinyServer(function(input, output) {
  output$text1 <-renderText({
    "Вы выбрали ..."
  })
})
```

Каждый объект должен содержать вывод одной из функций `render*`. При использовании функции `render*` нужно учитывать тип используемого реактивного объекта (табл. 2.5).

Таблица 2.5 - Семейство функций `render*`

Render функция	Создает
<code>renderImage</code>	изображение (сохраняет ссылку на файл)
<code>renderPlot</code>	график
<code>renderPrint</code>	вывод на печать
<code>renderTable</code>	фрейм данных, матрица, другая структурированная таблица
<code>renderText</code>	символьная строка
<code>renderUI</code>	Shiny тег объект или HTML

Shiny отслеживает, какие выходы зависят от каких виджетов. Когда пользователь изменяет виджет, Shiny перестроит все выходы, которые зависят от виджета, используя новое значение виджета. В результате все перестроенные объекты будут всегда актуальными.

Исследование интерактивных возможностей пакета Shiny позволило обобщить изложенный материал и выработать общие рекомендации по созданию реактивных приложений:

- использовать функцию `*Output` в файле `ui.R` для размещения в приложении реактивных выходных объектов;

- использовать функцию `render*` в `server.R`, чтобы указать Shiny как строить эти объекты;
- сохранять выражения `render*` в списке `output`, по одному для каждого реактивного выходного объекта вашего приложения;
- создавать реакции путем включения входных значений в выражения `render*`.

Эти рекомендации используются в главе 3 диссертации при разработке собственных интерактивных web-приложений Shiny.

2.5 Серверный сценарий приложения Shiny

В серверной части приложения важно определить необходимые функции и их предметный охват.

Реактивные значения, получаемые сервером, должны быть вложены в одну из функций, предназначенных для обработки интерактивных выходов виджетов UI. К реактивным функциям относятся `observe()`, `observeEvent()`, `reactive*()`, `eventReactive()` и `render*`.

Функции `update*()`

Функции `update*`: `updateActionButton`, `updateCheckboxGroupInput`, `updateCheckboxInput`, `updateDateInput`, `updateDateRangeInput`, `updateNumericInput`, `updateQueryString`, `updateRadioButtons`, `updateSelectInput`, `updateSliderInput`, `updateTabsetPanel`, `updateTextAreaInput`, `updateTextInput`.

Функции `update*()` предназначены для интерактивного обновления значений виджетов UI, таких как текстовое поле, выпадающий список и др. (Листинг 2.3).

Листинг 2.3. Пример использования функции `update*()`

```
## server
server <- function(input, output, session) {
  observe({
    txt <- paste("Value above is:", input$mytext)
    updateTextInput(session, "myresults", value=txt)
  })
}
```

```

    })
  }
## ui
ui <- basicPage(
  h3("An example of an update* function"),
  textInput("mytext", "Input goes here"),
  textInput("myresults", "Results will be printed here",
"Initial value")
)
## App
shinyApp(ui = ui, server = server)

```

Функции observe()

Функции `observe()` (в приложении их может быть несколько) запускаются при изменении реактивных значений в них. Они разработаны, чтобы слушать реактивные элементы, и отвечать на их изменение боковыми эффектами, наподобие обновления текстового поля или выпадающего списка в UI. В отличие от реактивных функций, рассматриваемых ниже, они не возвращают данные или значения.

Есть две тонкости при работе с функциями `observe()`. Код функции `observe()` будет выполняться, если изменяется любой из реактивных элементов внутри нее. Код функции `observeEvent()` будет выполняться при изменении специфического реактивного значения. Пример использования функции `observe()` представлен в листинге 2.4.

Листинг 2.4. Пример использования функции observe()

```

library(shiny)
server <- function(input, output, session) {
  observe({
    updateTextInput(session, inputId = "myresults", value
= input$mytext)
  })
}

```

```

    })
  }
  ui <- basicPage(
    h3("Значение первого текстового поля будет напечатано
во втором текстовом поле."),
    textInput("mytext", "Введите текст"),
    textInput("myresults", "Результат будет напечатан
здесь", "Initial value")
  )
  shinyApp(ui = ui, server = server)

```

Результат выполнения приведенного программного кода представлен на рисунке 2.12.

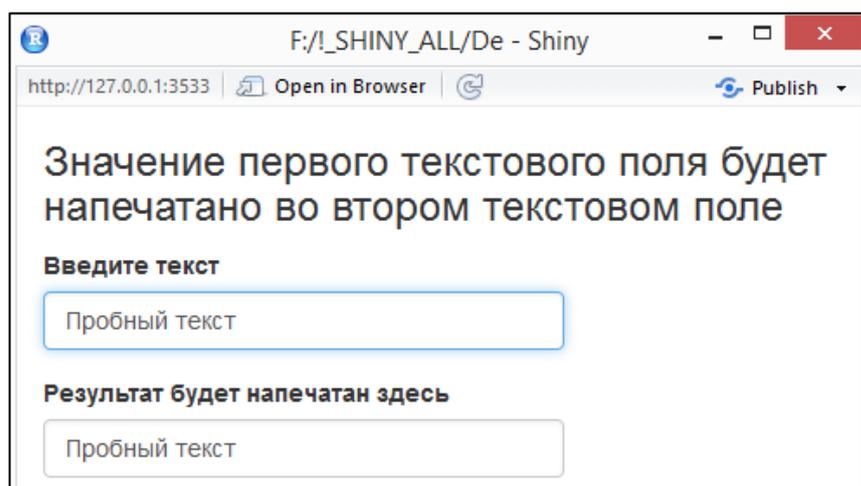


Рисунок 2.12 – Пример выполнения функции `observe()`

Рассмотрим пример использования функции `observeEvent()` (Листинг 2.5). В данном примере значение второго текстового поля будет изменяться при любом изменении первого текстового поля, но изменения слайдера никаких изменений на выходе не вызовет. Результат выполнения приведенного программного кода представлен на рисунке 2.13.

Листинг 2.5. Пример использования функции `observeEvent()`

```

library(shiny)
server <- function(input, output, session) {

```

```

observeEvent(input$mytext, {
  input$myslider
  txt <- paste(input$mytext, sample(1:10000, 1))
  updateTextInput(session, inputId = "myresults", value
= txt)
})
}
ui <- basicPage(
  h3("Текстовое поле результатов обновляется только при
изменении верхнего текстового поля (изменения слайдера не
вызывают обновления)."),
  sliderInput("myslider", "A slider:", min=0, max=1000,
value=500),
  textInput("mytext", "Введите текст в текстовое поле",
value = "Initial value"),
  textInput("myresults", "Результат будет отображаться
здесь")
)
shinyApp(ui = ui, server = server)

```

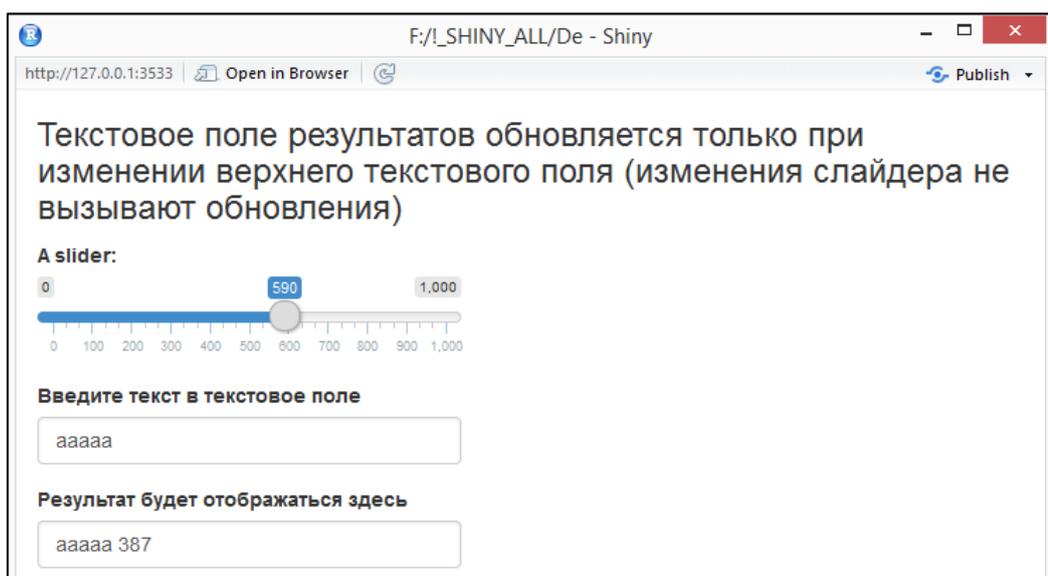


Рисунок 2.13 – Пример выполнения функции `observeEvent()`

В больших приложениях может встретиться ситуация, когда один `observe` требуется запустить раньше других. Для этого в функции `observe()` используется параметр `priority`. По умолчанию `priority=0`, и большие значения имеют больший приоритет (можно использовать и отрицательные числа).

Функции `reactive()`

Список функций `reactive()`: `reactive`, `reactiveFileReader`, `reactivePlot`, `reactivePoll`, `reactivePrint`, `reactiveTable`, `reactiveText`, `reactiveTimer`, `reactiveUI` (перечеркнуты устаревшие, вместо них используются аналогичные функции `render*`).

Функция `reactive()` используется как обычная функция R, за исключением того, что она запускается реактивным элементом. Т.к. функция `reactive()` возвращает результат, его обычно сохраняют как объект и используют где-нибудь в другом месте серверного сценария так же, как любую функцию R. Но есть одна большая разница: функция должна выполняться только внутри «реактивного контента» (т.е. в другой функции `reactive()`, `observe()` или `render*()`).

Не предполагается, что функции `reactive()` должны генерировать побочный эффект, как функции `observe()`. Так в следующем примере `reactive` используется для создания функции `myresults()`. Так как результат нужно вывести на консоль (боковой эффект), функция `reactive()` запускается из `observe`. Обратим внимание, что в `observe` используется функция `myresults()`, а не значение `myresults`. На листинге 2.6 представлен пример программного кода.

Листинг 2.6. Пример использования функции `reactive()`

```
library(shiny)
## server
server <- function(input, output, session) {
  mystring <- reactive({
    paste(input$mytext, " is what the user types")
  })
  observe({
```

```

    txt <- mystring()
    updateTextInput(session, inputId = "myresults", value
= txt)
  })
}
## ui
ui <- basicPage(
  h3("The reactive generates a string output which is
added to the results text box"),
  textInput("mytext", "Input goes here"),
  textInput("myresults", "Results will be printed
here", "")
)
## App
shinyApp(ui = ui, server = server)

```

Аналогично `observeEvent()`, функция `eventReactive()` определяет, какие реактивные значения запускают код этой функции. Пример использования функции `eventReactive()` представлен на листинге 2.7.

Листинг 2.7. Пример использования функции `eventReactive()`

```

library(shiny)
## server
server <- function(input, output, session) {
  myresults <- reactive({
    paste(input$mytext, input$myslider)
  })
  myresults_lim <- eventReactive(input$mytext, {
    paste(input$mytext, input$myslider)
  })
}

```

```

    observe(updateTextInput(session, "myresults", value =
myresults()))
    observe(updateTextInput(session, "myresults_lim", value
= myresults_lim()))
}
## ui
ui <- basicPage(

  h3("Использование eventReactive для ограничения
реакций"),
  sliderInput("myslider", "", min=0, max=1000,
value=500),
  textInput("mytext", "Введите текст"),
  textInput("myresults", "Текстовое поле + слайдер,
обновляется при каждом изменении", "Initial value"),
  textInput("myresults_lim", "Текстовое поле + слайдер,
обновляется при изменении текстового поля", "Initial
value")

)
## App
shinyApp(ui = ui, server = server)

```

Результат выполнения программного кода представлен на рисунке 2.14.

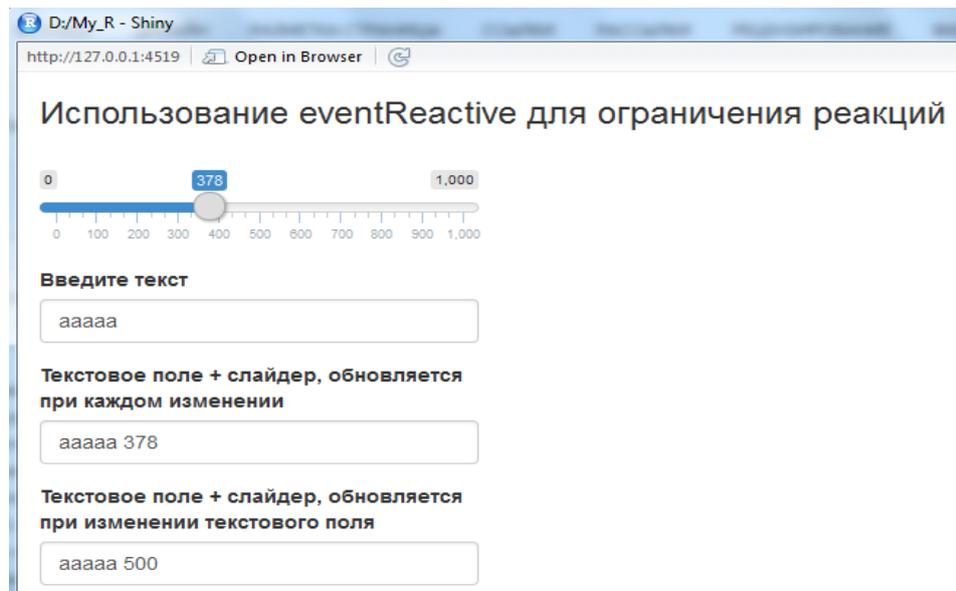


Рисунок 2.14 – Пример выполнения функции eventReactive()

Если функция observe (reactive или render*) не имеет внутри себя реактивного значения, она будет выполняться только один раз при запуске приложения и никогда больше. И наоборот, если включить реактивное значение в функцию observe (reactive или render*), ее код будет запускаться при запуске приложения всякий раз, когда изменяется реактивное значение, даже если это реактивное значение не используется в расчетах:

Так как реактивные значения в реактивных функциях будут запускать код последних, нужно стараться разбивать код на модули. Другими словами, нужно как можно сильнее разделять расчеты. Реактивные функции – хороший способ сделать это: можно использовать реактивные функции, чтобы изолировать код и запускать его только там, где это необходимо. Пример программного кода приведен на листинге 2.8.

Листинг 2.8. Пример использования функций observe() и reactive()

```
library(shiny)
## server
server <- function(input, output, session) {
  txt <- reactive({paste(input$mytext, sample(1:100,
1))})}
```

```

  val <- reactive({paste(input$myslider, sample(1:100,
1), sep="-")})
  observe({
    res <- paste0(txt(), " | Slider ", val())
    updateTextInput(session, "myresults", value = res)
  })
}
## ui
ui <- basicPage(
  h3("Changes to the text box and slider are separated
so that a change to the text box will not affect the
slider part of the results textbox"),
  sliderInput("myslider", "A slider:", min=0, max=1000,
value=500),
  textInput("mytext", "Input goes here", "Text"),
  textInput("myresults", "Results will be printed
here", "Initial value")
)
## App
shinyApp(ui = ui, server = server)

```

Функции render*()

Список функций render*: renderDataTable, renderImage, renderPlot, renderPrint, renderTable, renderText, renderUI.

Каждая функция render* принимает один аргумент – выражение R, окруженное скобками { }. Выражение может быть простой строкой текста или включать много строк кода, как если бы это был вызов сложной функции. Это выражение можно представить как множество инструкций, которые предоставляются Shiny для хранения на будущее. Shiny будет выполнять эти инструкции всякий раз, когда нужно обновить объект. Чтобы это работало,

выражение должно возвращать нужный объект (фрагмент текста, график, фрейм и т.д.).

Аргумент `input` сохраняет текущие значения всех виджетов приложения под именами, присвоенными виджетам в `ui.R`. Shiny автоматически сделает объект интерактивным, если объект использует значения `input`.

Если нужно вернуть блок элементов UI – например, абзац, затем текстовое поле и селектор, – то нужно использовать `renderUI` и `outputUI` и вместо возврата одного объекта вернуть список или `tagList` объектов (эти функции взаимозаменяемы).

Пример использования `observe`, `reactive` и `render*` в одном приложении представлен на листинге 2.9.

Листинг 2.9. Пример использования функций `observe`, `reactive` и `render*`

```
## server
library(shiny)
server <- function(input, output, session) {
  observe({
    updateTextInput(session, "mytext",
value=input$myslider)
  })
  dat <- reactive({
    input$myslider
    cars[1:input$myslider,]
  })
  output$mytable <- renderTable({
    dat()
  })
}
## ui
ui <- fluidPage(
```

```

titlePanel("Пример приложения, использующего observe,
reactive и render"),
sidebarLayout(
  sidebarPanel(
    sliderInput("myslider", "Количество отображаемых
строк", min=1, max=50, value=5),
    textInput(inputId = "mytext", label = "Значение
слайдера")
  ), # end sidebar panel
  mainPanel(
    tabsetPanel(
      tabPanel("Таблица", tableOutput("mytable"))
    )
  ) # end main panel
)
)
## App
shinyApp(ui = ui, server = server)

```

Результат работы приведенного программного кода представлен на рисунке 2.15.



Рисунок 2.15 - Пример использования функций observe, reactive и render* в одном приложении

Особенности создания Shiny приложения

Приложение Shiny – это просто каталог, содержащий определение пользовательского интерфейса, серверный скрипт и дополнительные данные, другие скрипты и ресурсы, которые нужны для поддержки приложения.

Для начала нужно создать новый пустой каталог, затем создать в нем пустые файлы `ui.R` и `server.R`.

Рассмотрим пример, когда приложение будет создаваться в каталоге `D:/My_R/Shiny/template`:

```
D:/My_R/Shiny/template
|-- ui.R
|-- server.R
```

Минимальный исходный код для каждого файла будет иметь вид, представленный в листинге 2.10.

Листинг 2.10. Минимальный исходный код приложения

ui.R:

```
library(shiny)
shinyUI(pageWithSidebar(
  headerPanel("Template"),
  sidebarPanel(),
  mainPanel()
))
```

server.R:

```
library(shiny)
shinyServer(function(input, output) {
})
```

Положение кода R в сценарии `server.R` определяет, как часто он выполняется. Есть три зоны, в которых код выполняется с разной частотой:

```
# Zone 1
library(shiny)
library(ggplot2)
```

```
shinyServer(function(input, output) {
  # Zone 2
  output$dynamicPlot <- renderPlot({
    # Zone 3
  })
})
```

Zone 1 – код в этой зоне выполняется только один раз при старте приложения; здесь обычно загружаются библиотеки и считываются большие наборы статистических данных;

Zone 2 - код в этой зоне выполняется каждый раз, когда пользователь запрашивает приложение; этот код можно использовать, например, для загрузки данных за последние x дней, т.к. для каждого пользователя эти данные могут быть разными;

Zone 3 - код в этой зоне выполняется каждый раз, когда происходят изменения в UI; здесь помещается реактивный код.

Созданное приложение можно запустить командами:

```
> library(shiny)
> runApp("~/shinyapp")
```

Например:

```
> runApp("D:/My_R/Shiny/template")
```

Три функции `headerPanel()`, `sidebarPanel()` и `mainPanel()` определяют разные области пользовательского интерфейса.

2.6 Рекомендации по созданию интерактивных учебных приложений

Анализ возможностей технологии Shiny позволил сформулировать общие рекомендации по созданию интерактивных учебных web-приложений.

Любое Shiny приложение имеет две составляющие: скрипт пользовательского интерфейса и серверный скрипт. Таким образом, приложение строится на основе как минимум двух файлов с фиксированными

именами `ui.R` и `server.R`, которые целесообразно разместить в отдельном каталоге.

Рекомендации по созданию пользовательского сценария.

1. Разработать макет интерфейса интерактивного web-приложения Shiny.
2. Создать функцию `ui <- fluidPage()`, содержащую определение html-страницы.
3. Выбрать оптимальную раскладку html-страницы.
4. Определить виджеты ввода данных.
5. Определить выходные объекты html-страницы.
6. Разместить входные и выходные объекты в нужных местах html-страницы.
7. Добавить необходимый html-код и отформатировать web-страницу с использованием каскадных таблиц стилей CSS.

Рекомендации по созданию серверного сценария.

1. Создать функцию `server <- function(input, output)` для получения сервером значений клиентских параметров.
2. Использовать функции `render*()` для создания выходных объектов различных типов (текст, графики, таблицы и т.п.).
3. Добавить в функции `render*()` необходимый R-код.

Запуск web-приложения Shiny может осуществляться с любого компьютера, на котором установлены R, пакет Shiny и другие необходимые библиотеки.

Сделать разработанные web-приложения Shiny доступными для пользователей через Интернет можно двумя способами:

1. Разместить приложение на собственном web-сервере Shiny.
2. Разместить приложение на хостинге `shinyapps.io` - специализированном сайте от компании RStudio для размещения web-приложений Shiny.

Любое приложение Shiny, размещенное на хостинге shinyapps.io, выполняется бесплатно 25 часов в месяц. При превышении этого предела сервис становится платным.

Вывод по второй главе

Исследование возможностей пакета Shiny как инструмента для создания интерактивных учебных компонентов позволило выявить следующие основные преимущества:

1. На клиентской стороне создается HTML-страница в стандарте HTML5, что существенно экономит объем трафика, передаваемого от сервера клиенту.

2. R – мощный язык программирования, приспособленный для анализа, обработки и визуализации огромных выборок данных, что позволяет применять его и интерактивные приложения Shiny в Data Mining, статистической обработке данных, моделировании и др.

3. Гибкая и мощная интерактивность позволяет создавать сложные интерактивные web-приложения для визуализации данных, кластеризации, нейронных сетей, моделирования процессов и систем, разработки тестов, интерактивный лабораторный практикум для учебного процесса, информационные системы и многое другое.

Глава 3 РАЗРАБОТКА ИНТЕРАКТИВНЫХ УЧЕБНЫХ WEB-ПРИЛОЖЕНИЙ

Данная глава посвящена разработке интерактивных учебных web-приложений, иллюстрирующих принцип работы и возможности технологии Shiny. В параграфах 3.1, 3.2 создается и анализируется интерактивное web-приложение кластеризации цветков ирисов с использованием известной выборки данных Фишера. Параграфы 3.3, 3.4 посвящены разработке интерактивного web-приложения анализа характеристик легковых автомобилей на примере выборки данных mtcars из пакета R.

Разработанные интерактивные web-приложения можно использовать в учебном процессе по дисциплинам естественно-научного профиля, связанным с интеллектуальным анализом данных.

3.1 Анализ выборки данных Фишера, пользовательский интерфейс интерактивного web-приложения кластеризации данных

Ирисы Фишера – один из самых популярных в литературе по статистике и нейронным сетям набор данных, обычно используемый для иллюстрации работы различных алгоритмов классификации, кластеризации и анализа данных [44, 52]. Выборка состоит из 150 экземпляров ирисов, для которых измерялись 4 характеристики: длина и ширина чашелистика (Sepal.Length и Sepal.Width), длина и ширина лепестка (Petal.Length и Petal.Width). Выборка данных состоит из трех классов цветков ирисов – setosa, versicolor, virginica, - по 50 экземпляров каждого класса и представлена на рисунке 3.1.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
.....					
49	5.3	3.7	1.5	0.2	setosa
50	5.0	3.3	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
.....					
99	5.1	2.5	3.0	1.1	versicolor
100	5.7	2.8	4.1	1.3	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
.....					
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

Рисунок 3.1 – Фрагмент выборки данных Фишера

Описательная статистика выборки данных для трех классов представлена в таблицах и иллюстрируется диаграммами размахов на рисунках 3.2 – 3.4.

```
> summary(iris[iris$Species=="setosa",])      ## класс setosa
Sepal.Length   Sepal.width   Petal.Length   Petal.width   Species
Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.100   setosa   :50
1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400   1st Qu.:0.200   versicolor: 0
Median :5.000   Median :3.400   Median :1.500   Median :0.200   virginica : 0
Mean   :5.006   Mean   :3.428   Mean   :1.462   Mean   :0.246
3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575   3rd Qu.:0.300
Max.   :5.800   Max.   :4.400   Max.   :1.900   Max.   :0.600
> boxplot(iris[iris$Species=="setosa",1:4])
```

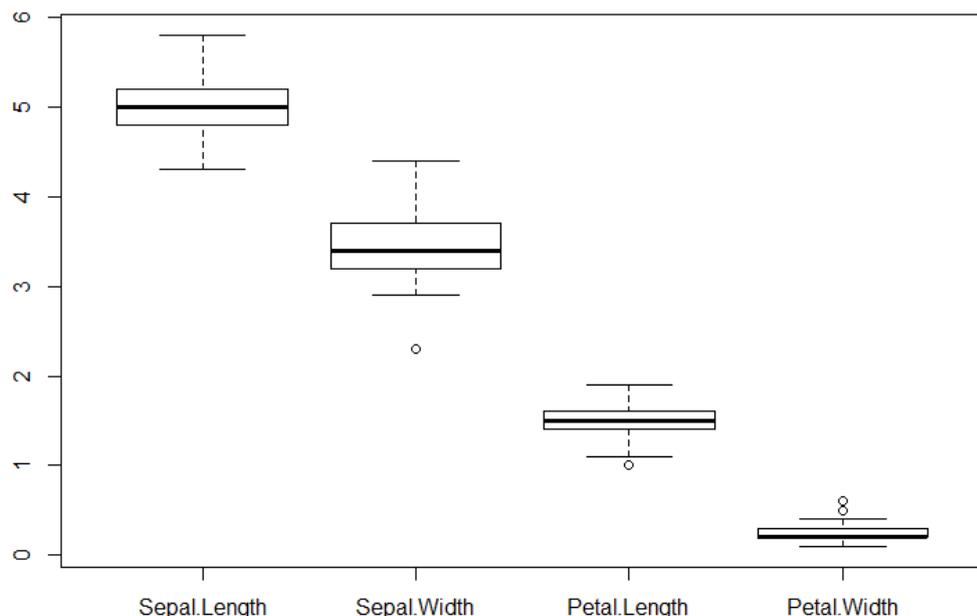


Рисунок 3.2 – Диаграмма размахов элементов выборки данных Фишера класса setosa

```

> summary(iris[iris$Species=="versicolor",]) ## класс versicolor
  Sepal.Length   Sepal.width   Petal.Length   Petal.width   Species
Min.   :4.900   Min.   :2.000   Min.   :3.00   Min.   :1.000   setosa    : 0
1st Qu.:5.600   1st Qu.:2.525   1st Qu.:4.00   1st Qu.:1.200   versicolor:50
Median :5.900   Median :2.800   Median :4.35   Median :1.300   virginica : 0
Mean   :5.936   Mean   :2.770   Mean   :4.26   Mean   :1.326
3rd Qu.:6.300   3rd Qu.:3.000   3rd Qu.:4.60   3rd Qu.:1.500
Max.   :7.000   Max.   :3.400   Max.   :5.10   Max.   :1.800
> boxplot(iris[iris$Species=="versicolor",1:4])

```

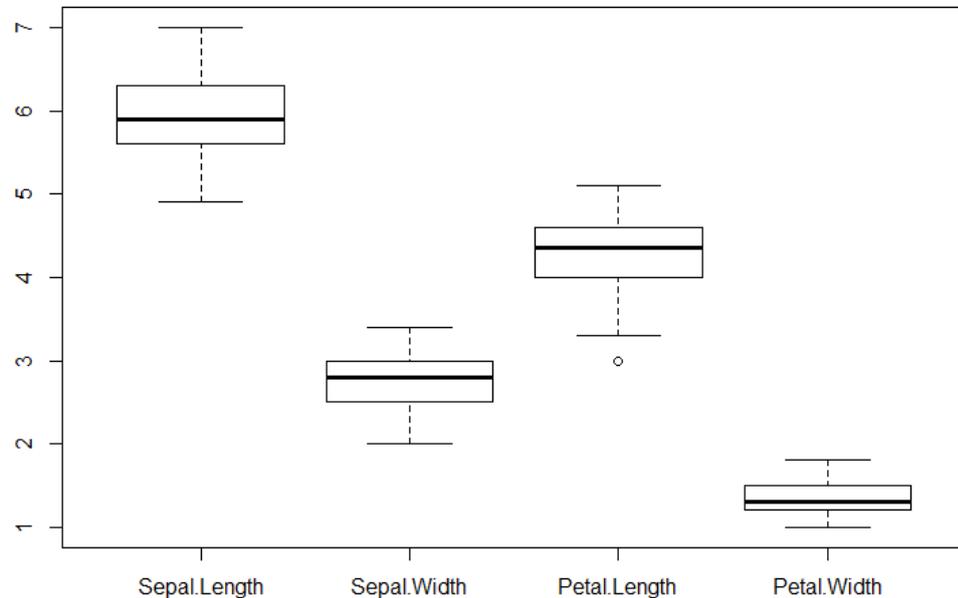


Рисунок 3.3 – Диаграмма размахов элементов выборки данных Фишера класса versicolor

```

> summary(iris[iris$Species=="virginica",]) ## класс virginica
  Sepal.Length   Sepal.width   Petal.Length   Petal.width   Species
Min.   :4.900   Min.   :2.200   Min.   :4.500   Min.   :1.400   setosa    : 0
1st Qu.:6.225   1st Qu.:2.800   1st Qu.:5.100   1st Qu.:1.800   versicolor:0
Median :6.500   Median :3.000   Median :5.550   Median :2.000   virginica :50
Mean   :6.588   Mean   :2.974   Mean   :5.552   Mean   :2.026
3rd Qu.:6.900   3rd Qu.:3.175   3rd Qu.:5.875   3rd Qu.:2.300
Max.   :7.900   Max.   :3.800   Max.   :6.900   Max.   :2.500
> boxplot(iris[iris$Species=="virginica",1:4])

```

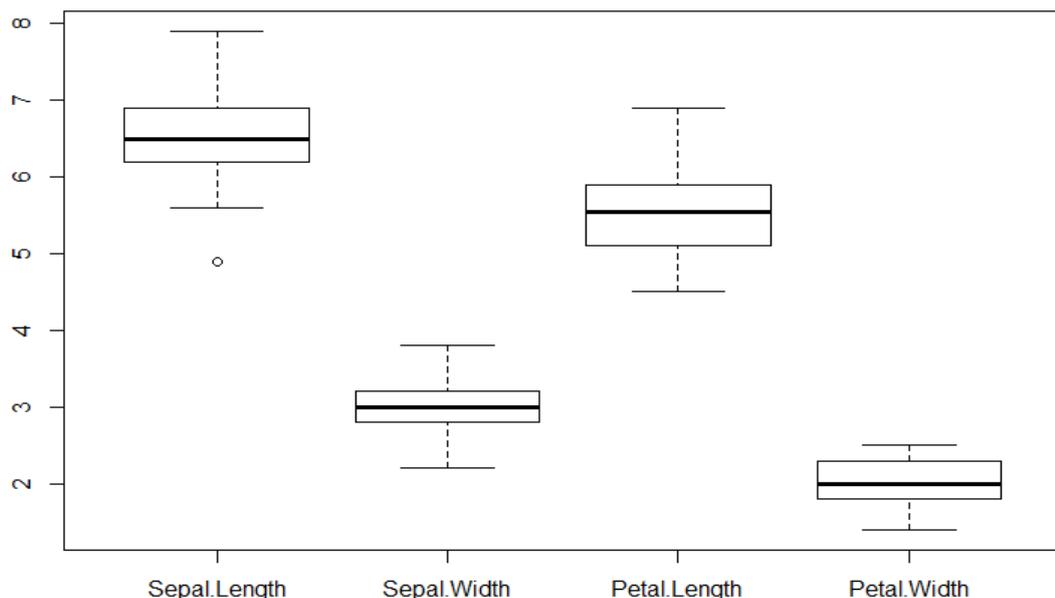


Рисунок 3.4 – Диаграмма размахов элементов выборки данных Фишера класса virginica

Из диаграмм размахов видно, что элементы класса *setosa* принципиально отличаются от элементов классов *versicolor* и *virginica*, что подтверждает известный факт, что класс *setosa* пространственно отделен от двух других классов, а классы *versicolor* и *virginica* частично перекрываются друг с другом.

Это же видно из диаграмм рассеяния (рис. 3.5):

```
> plot(iris[,1:4], col=c(rep("black",50), rep("blue",50),
rep("red",50)), pch=16, cex=0.8)
```

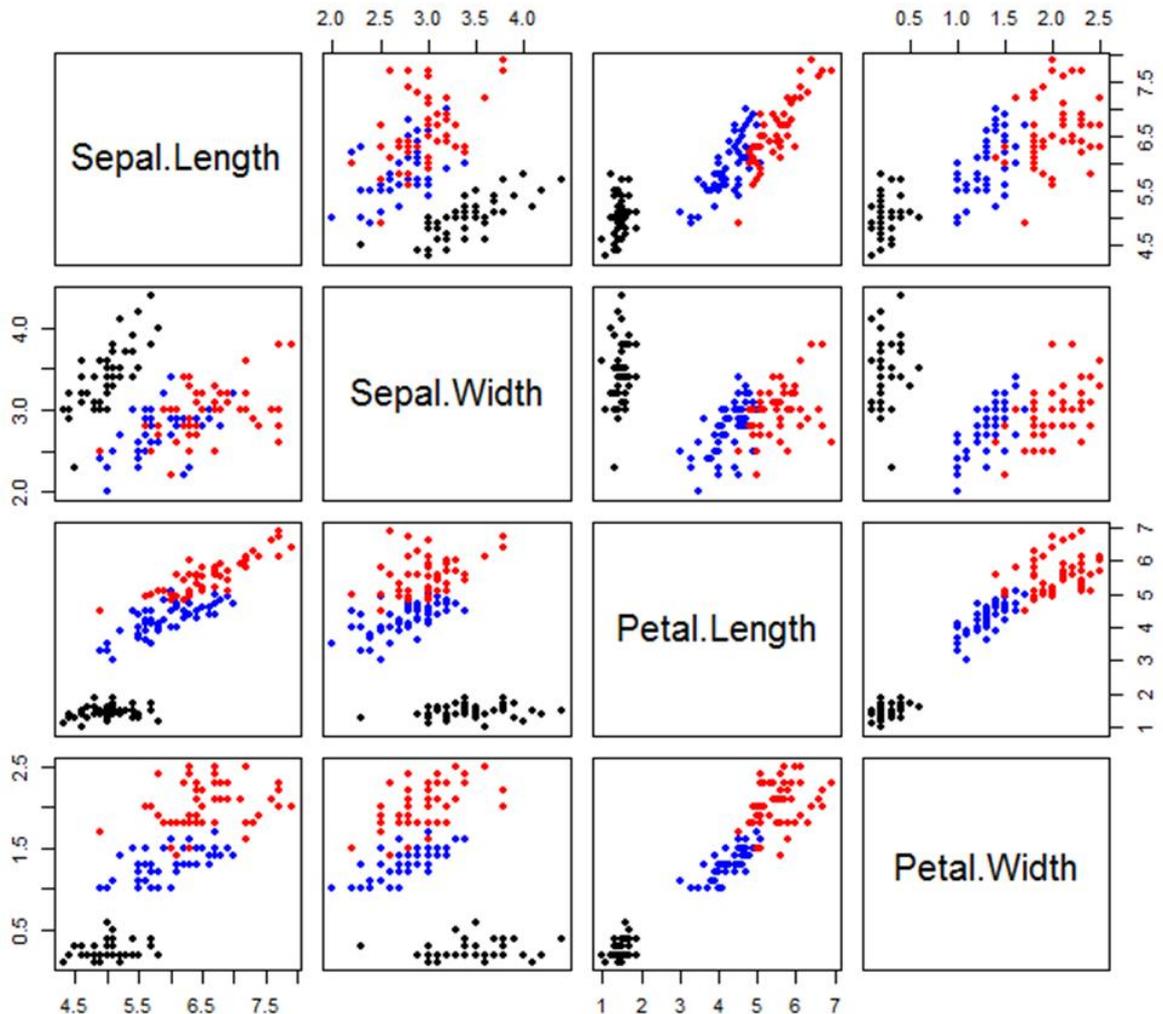


Рисунок 3.5 – Диаграммы рассеяния для всех пар числовых переменных выборки данных Фишера

На диаграмме рассеяния каждому элементу данных соответствует точка, координаты которой (в декартовой системе координат) равны значениям двух переменных.

В случае многомерных данных для полной визуализации данных на плоскости необходимо построить N диаграмм рассеяния, по одной на каждую пару переменных.

$$N = \frac{m*(m-1)}{2},$$

где m – число переменных или размерность.

Диаграммы рассеяния часто используются для демонстрации наличия или отсутствия корреляции между переменными.

Матрица корреляции числовых переменных выборки данных iris показана на рисунке 3.6.

```
> cor(iris[,1:4])
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.0000000	-0.1175698	0.8717538	0.8179411
Sepal.Width	-0.1175698	1.0000000	-0.4284401	-0.3661259
Petal.Length	0.8717538	-0.4284401	1.0000000	0.9628654
Petal.Width	0.8179411	-0.3661259	0.9628654	1.0000000

Рисунок 3.6 - Матрица корреляции числовых переменных выборки данных iris

Следует обратить внимание на сильную корреляцию переменных Petal.Length, Petal.Width и Sepal.Length, что также хорошо видно на диаграммах рассеяния (рис. 3.5).

Разрабатываемое интерактивное приложение Shiny позволяет пользователю выбрать из списка переменные x , y и задать число кластеров с использованием соответствующих виджетов. Затем делается запрос серверу Shiny на выполнение приложения. Сервер принимает значения виджетов, выполняет кластеризацию с помощью алгоритма ram и возвращает результат кластеризации в виде двух графиков. Алгоритм ram является улучшенной версией алгоритма K-means - одного из первых алгоритмов кластеризации, предложенного более пятидесяти лет назад и до сих пор широко используемого на практике.

Эскиз html-страницы приложения приведен на рисунке 3.7.

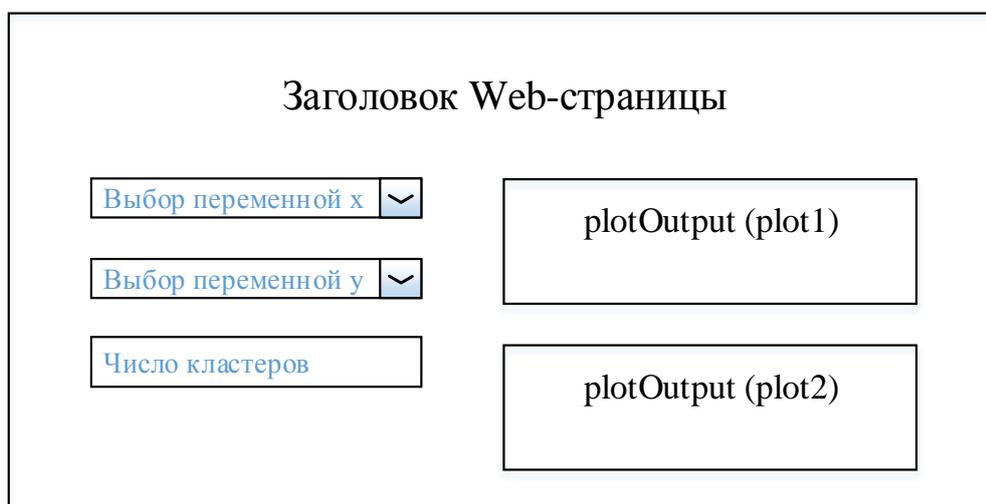


Рисунок 3.7 – Эскиз html-страницы интерактивного приложения кластеризации данных

Страница состоит из заголовка; левой панели, на которой размещаются три виджета: два виджета `selectInput` для задания переменных x и y , отображаемых на графиках, и виджет `numericInput` для задания числа кластеров, на которое алгоритм делит исходную выборку данных; правой панели, на которой выводятся результаты кластеризации в виде двух графиков.

Интерактивность проявляется в том, что изменение значения любого из виджетов приводит к мгновенному изменению графиков, аналогично тому, как любое изменение значение ячейки электронной таблицы приводит к мгновенному пересчету всех зависимых ячеек.

Пользователь может выбирать на левой панели координаты x и y графиков из списка `Petal.Length`, `Petal.Width`, `Sepal.Length`, `Sepal.Width` и `Species`, а также произвольно задавать необходимое число кластеров.

Пользовательский интерфейс созданного приложения показан на рисунке 3.8.

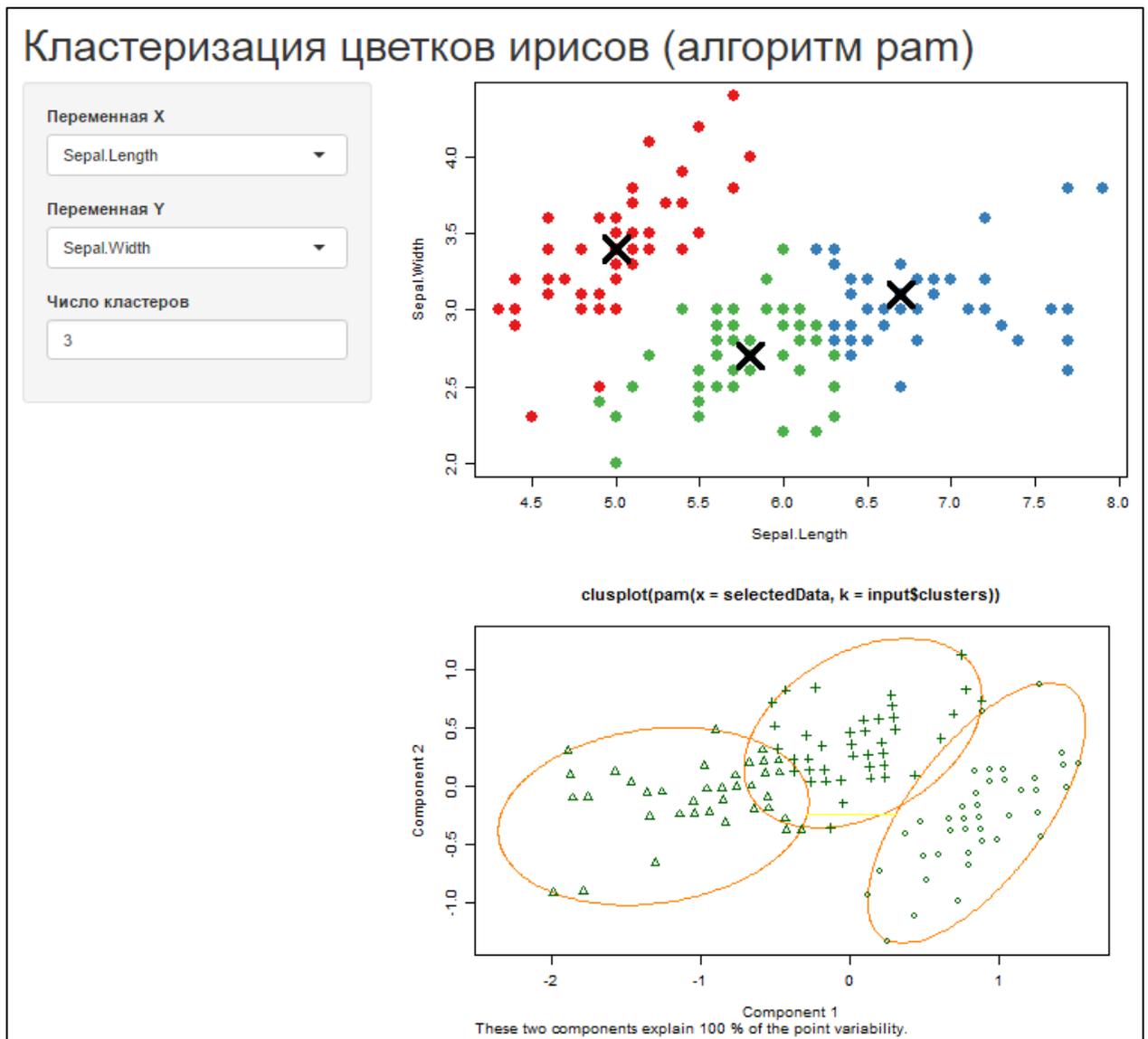


Рисунок 3.8 – Пользовательский интерфейс приложения кластеризации для числа кластеров $k=3$

Отображение данных в окне браузера соответствует эскизу html-страницы (рис. 3.7) и обеспечивается применением раскладки sidebarPanel.

На верхнем графике центры кластеров показаны крестиками, а принадлежащие им элементы данных выделены определенным цветом. Нижний график содержит такую же информацию, но границы кластеров представляют собой эллипсы наименьшей площади, охватывающие элементы каждого кластера. Кроме того, координатами точек на графиках являются две главные компоненты, которые для цветков ирисов близки к координатам Petal.Length и

Petal.Width, поэтому изменения виджетов `selectInput` не влияет на вид второго графика, он чувствителен только к изменению числа кластеров.

3.2 Логический анализ и программный код интерактивного web-приложения кластеризации данных

На рисунке 3.9 показана диаграмма последовательностей web-приложения кластеризации.

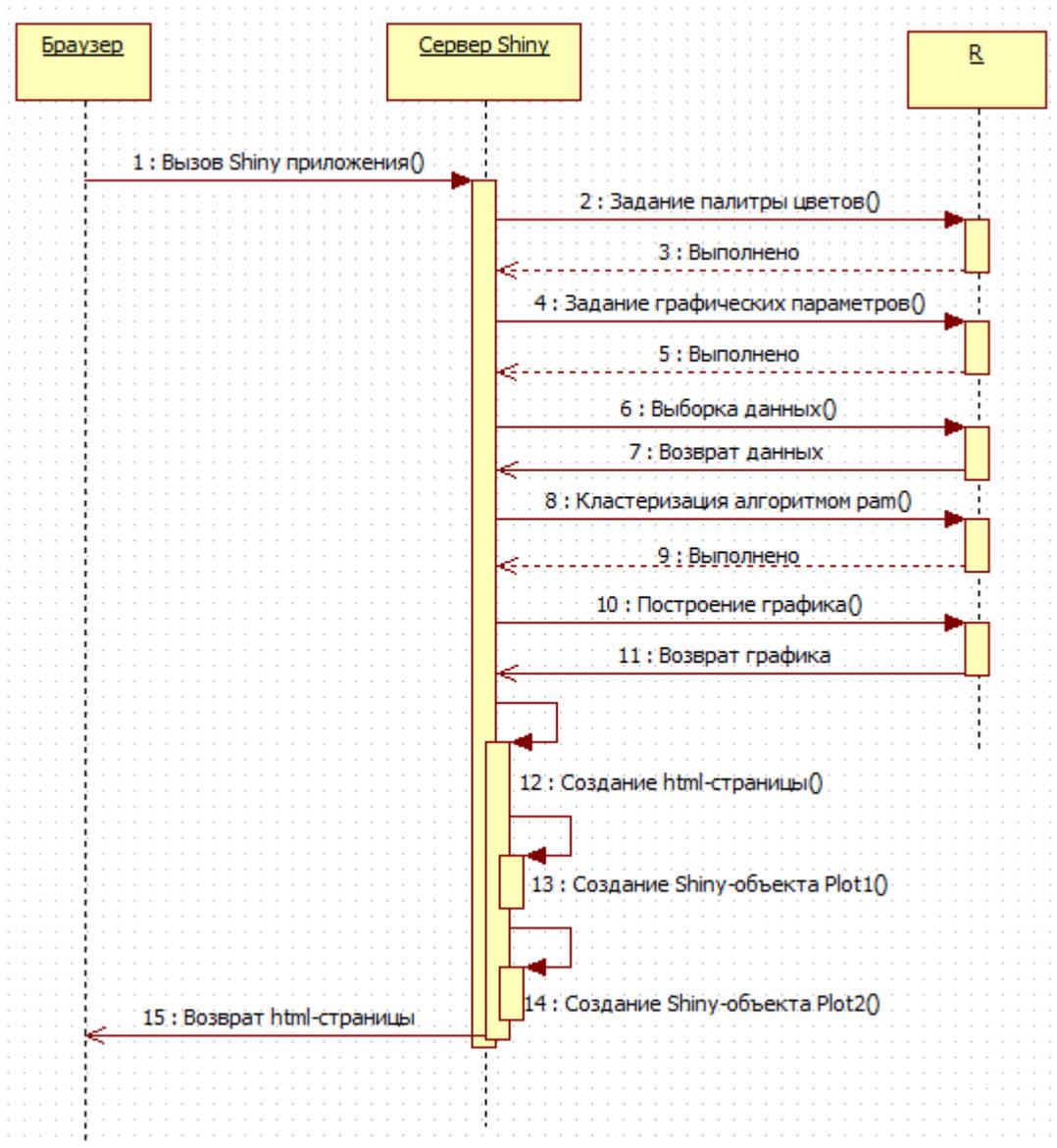


Рисунок 3.9 – Диаграмма последовательностей web-приложения кластеризации

Взаимодействие между браузером пользователя, сервером Shiny и программой R осуществляется следующим образом:

1. Пользователь делает в окне браузера запрос на выполнение сценариев `ui.R` и `server.R` web-приложения.

2. Shiny-сервер принимает значения параметров виджетов и делает ряд запросов к программе R на выполнение команд сценариев: определить палитру цветов для графиков, задать графические параметры, выбрать данные с использованием значений виджетов, кластеризовать данные алгоритмом `ram`, построить графики.

3. Shiny-сервер формирует html-страницу, базируясь на полученной информации от программы R. В частности, создаются объекты `plot1` и `plot2` для построения графиков в окне браузера пользователя.

4. Shiny-сервер отправляет созданные страницы браузеру пользователя.

Код R клиентского и серверного сценариев представлен на листинге 3.1.

Листинг 3.1. Фрагмент кода R клиентского и серверного сценариев

```
ui.R
pageWithSidebar(
  headerPanel('Кластеризация цветков ирисов (алгоритм ram)'),
  sidebarPanel(
    selectInput('xcol', 'Переменная X', names(iris)),
    selectInput('ycol', 'Переменная Y', names(iris),
selected=names(iris)[[2]]),
    numericInput('clusters', 'Число кластеров', 3, min = 1, max =
9)
  ),
  mainPanel(
    plotOutput('plot1'),
    plotOutput('plot2')
  )
)

server.R
library(cluster)

function(input, output) {
  output$plot1 <- renderPlot({
    palette(c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3",
"#FF7F00", "#FFFF33", "#A65628", "#F781BF",
"#999999"))
    par(mar = c(5.1, 4.1, 0, 1))
    selectedData <- iris[, c(input$xcol, input$ycol)]
```

```

clusters <- pam(selectedData, input$clusters)
plot(selectedData,
      col = clusters$cluster,
      pch = 20, cex = 3)
points(clusters$medoids, pch = 4, cex = 4, lwd = 4)
})
output$plot2 <- renderPlot({
  selectedData <- iris[, c(input$xcol, input$ycol)]
  clusters1 <- pam(selectedData, input$clusters)
  clusplot(clusters1)
})
}

```

HTML-код web-страницы приложения представлен в Приложении А.

3.3 Анализ выборки данных mtcars, пользовательский интерфейс интерактивного web-приложения Shiny

Для разработки данного приложения Shiny выбрана встроенная в базовую инсталляцию R выборка данных mtcars из американского журнала Motor Trend за 1974 год [44, 48]. Выборка содержит информацию о дизайне и технических характеристиках (число цилиндров, объем и мощность двигателя, расход топлива и т.д.) для 32 марок автомобилей и представлена в таблице 3.2.

Таблица 3.2 – Фрагмент выборки данных mtcars

```

> mtcars

```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
.....
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Описательная статистика выборки данных mtcars представлена на рисунке 3.10.

```

> summary(mtcars[,1:5])
      mpg          cyl          disp          hp          drat
Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0   Min.   :2.760
1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080
Median :19.20   Median :6.000   Median :196.3   Median :123.0   Median :3.695
Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7   Mean   :3.597
3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920
Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0   Max.   :4.930

> summary(mtcars[,6:10])
      wt          qsec          vs          am          gear
Min.   :1.513   Min.   :14.50   Min.   :0.0000   Min.   :0.0000   Min.   :3.000
1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000
Median :3.325   Median :17.71   Median :0.0000   Median :0.0000   Median :4.000
Mean   :3.217   Mean   :17.85   Mean   :0.4375   Mean   :0.4062   Mean   :3.688
3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000
Max.   :5.424   Max.   :22.90   Max.   :1.0000   Max.   :1.0000   Max.   :5.000

```

Рисунок 3.10 - Описательная статистика выборки данных mtcars

Разрабатываемое интерактивное web-приложение Shiny выводит в окно браузера первые элементы выборки данных и ее описательную статистику. Приложение позволяет пользователю выбрать число цилиндров `cyl`, трансмиссию `am` или число скоростей `gear` и интерактивно выводит зависимость расхода топлива `mpg` от выбранной переменной в виде диаграммы размаха.

Эскиз html-страницы приложения приведен на рис. 3.11.

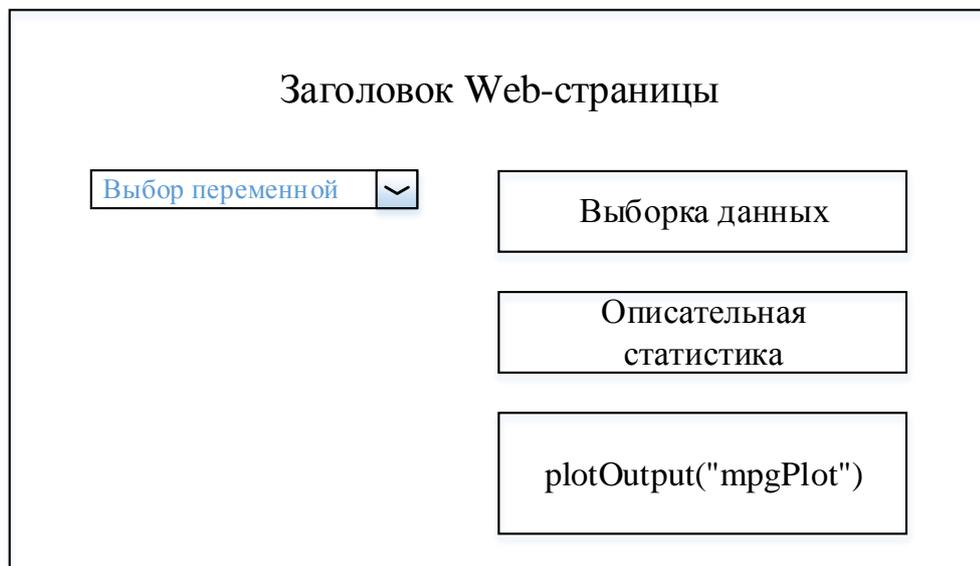


Рисунок 3.11 – Эскиз html-страницы интерактивного приложения изучения характеристик автомобилей

Страница состоит из заголовка; левой панели, на которой размещается виджет `selectInput` для выбора значения переменной; правой панели, на которой представлена часть выборки, описательная статистика, выводятся результаты в виде графика. При изменении значения переменной происходит мгновенное изменение графика.

Пользовательский интерфейс модели представлен на рисунке 3.12.

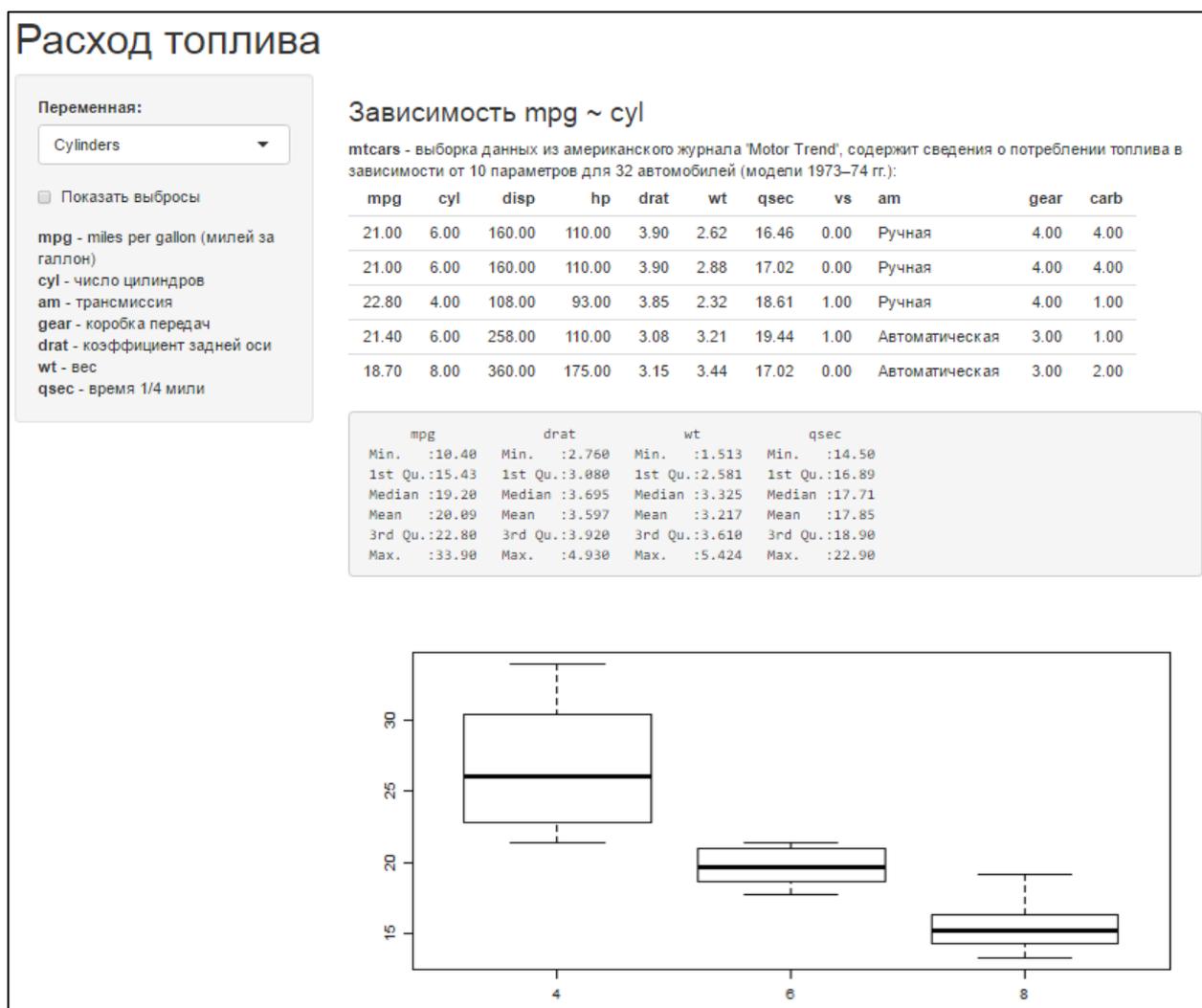


Рисунок 3.12 - Пользовательский интерфейс приложения характеристик автомобилей

Окно приложения содержит слева боковую панель с элементами для управления моделью. В основной части окна представлены результаты выполнения сценария. Представление данных в приложении обеспечивается

применением раскладки sidebarPanel. Для выбора значения входной переменной на левой панели используется виджет selectInput.

3.4 Логический анализ и программный код интерактивного web-приложения mtcars

На рисунке 3.13 показана диаграмма последовательностей интерактивного web-приложения mtcars.

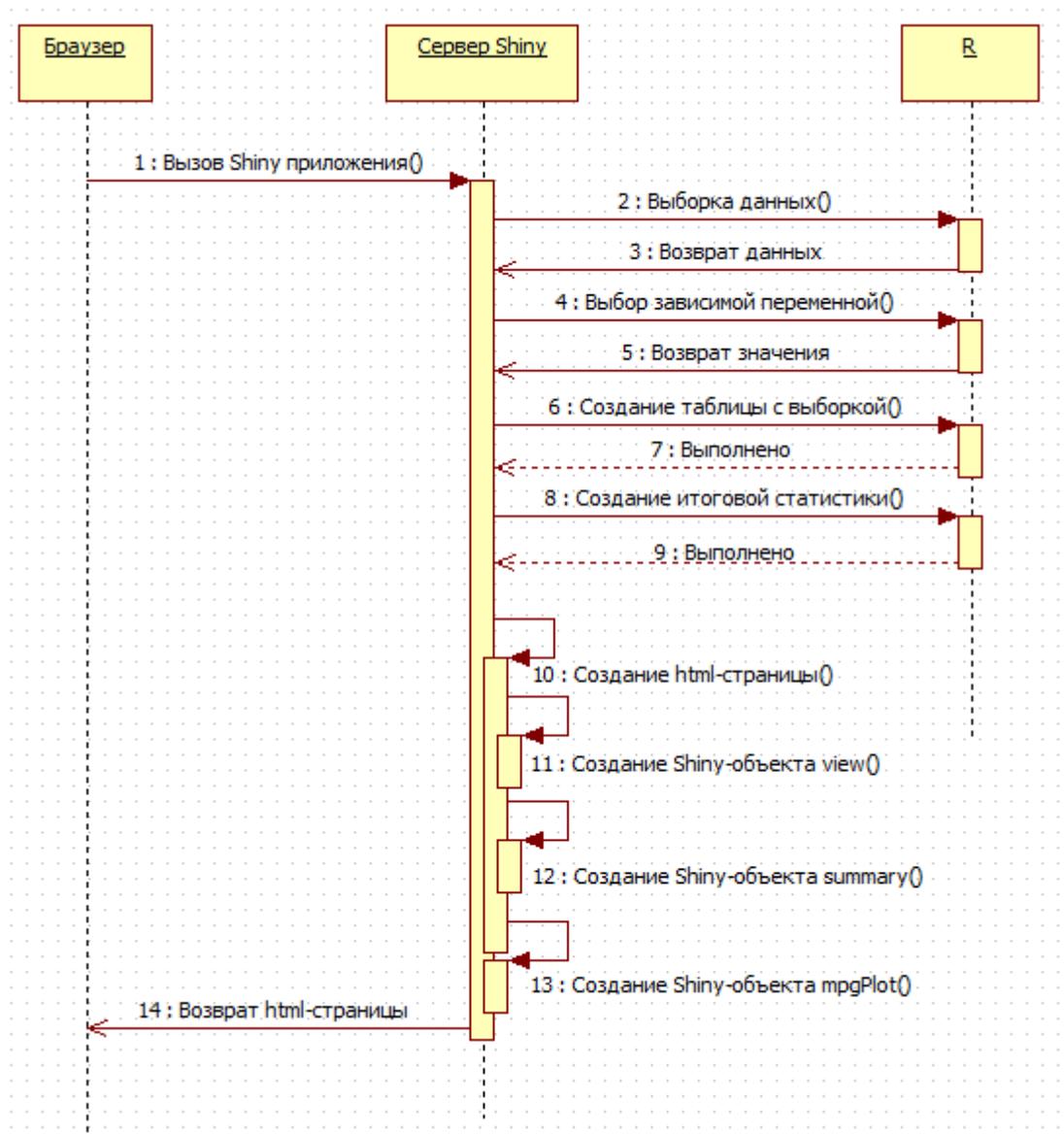


Рисунок 3.13 – Диаграмма последовательностей web-приложения характеристик автомобилей

Взаимодействие между браузером пользователя, сервером Shiny и программой R осуществляется следующим образом:

1. Пользователь делает в окне браузера запрос на выполнение сценариев `ui.R` и `server.R` web-приложения.

2. Shiny-сервер принимает значение параметра виджета и делает ряд запросов к программе R на выполнение команд сценариев: выбрать данные для построения таблицы с выборкой, выполнить итоговую статистику данных, выбрать данные с использованием значения виджета, построить график.

3. Shiny-сервер формирует html-страницу, базируясь на полученной информации от программы R. В частности, создаются объекты `view`, `summary` и `mpgPlot` для построения таблиц и графика в окне браузера пользователя.

4. Shiny-сервер отправляет созданные страницы браузеру пользователя.

Код клиентского приложения и серверного сценария представлен на листинге 3.2.

Листинг 3.2. Фрагмент кода R клиентского и серверного сценариев

`ui.R`

```
library (shiny)
shinyUI (pageWithSidebar(
  headerPanel ("Расход топлива"),
  sidebarPanel (width=3, # ----- sidebarPanel -----
    selectInput ("variable", "Переменная:",
      list ("Cylinders" = "cyl", "Transmission" = "am", "Gears"
= "gear")),
    checkboxInput ("outliers", "Показать выбросы", FALSE),
    HTML ("mpg - miles per gallon (милей за галлон)<br>
      <b> cyl </b> - число цилиндров <br>
      <b> am </b> - трансмиссия <br>
      <b> gear </b> - коробка передач <br>
      <b> drat </b> - коэффициент задней оси<br>
      <b> wt </b> - вес <br>
      <b> qsec </b> - время 1/4 мили <br>
    ") # - /HTML
  ), # --- /sidebarPanel
```

```

mainPanel( # ----- mainPanel -----
  h3 ("Зависимость ", textOutput("caption", inline = TRUE)),
  HTML ("mtcars - выборка данных из американского журнала
'Motor Trend', содержит
  сведения о потреблении топлива в зависимости от 10
параметров для 32 автомобилей (модели 1973-74 гг.):"),
  tableOutput ("view"),
  verbatimTextOutput ("summary"),
  plotOutput ("mpgPlot") # - вывод графика
) # - /mainPanel
)) # - /shinyUI

```

server.R

```

library (shiny)

mpgData <- mtcars # - исходная выборка данных
mpgData$am <- factor(mpgData$am, labels = c("Автоматическая",
"Ручная"))

shinyServer (function(input, output) {
  formulaText <- reactive ({
    paste ("mpg ~", input$variable)
  })
  output$caption <- renderText ({
    formulaText()
  })
  output$mpgPlot <- renderPlot ({
    boxplot (as.formula(formulaText()),
            data = mpgData,
            outline = input$outliers)
  })
  output$view <- renderTable({
    head (mpgData, n = 5)
  })
  output$summary <- renderPrint({
    dataset <-mpgData[,c("mpg","drat","wt","qsec")]
    summary (dataset)
  })
})

```

```
}) # --- shinyServer
```

HTML-код web-приложения представлен в Приложении Б.

Разработанная модель может также использоваться при изучении зависимостей в однотипных задачах.

Выводы по третьей главе

Таким образом, показано, что пакет Shiny из программы анализа данных R может успешно применяться для разработки интерактивных элементов учебных комплексов. Разработанные интерактивные компоненты учебных комплексов позволяют проводить модельные эксперименты с варьируемыми значениями входных переменных и могут найти эффективное применение в учебном процессе.

Глава 4 ТЕСТИРОВАНИЕ И АНАЛИЗ ЭФФЕКТИВНОСТИ ИНТЕРАКТИВНЫХ WEB-ПРИЛОЖЕНИЙ

4.1 Тестирование и анализ эффективности интерактивного web-приложения кластеризации

Данный раздел содержит результаты тестирования интерактивного web-приложения кластеризации.

В приведенных экспериментах на ирисах Фишера решалась задача кластеризации, т.е. на вход алгоритма были поданы исходные данные без указания классов, к которым они принадлежат. Задача кластеризации ирисов состоит в том, чтобы найти центры кластеров и отнести образцы данных к тому или иному кластеру.

Ниже представлены описательная статистика и результаты кластеризации в зависимости от входных параметров и числа кластеров (рис. 4.2 – 4.4).

```
> summary(iris[,2,3])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 2.000  2.800   3.000   3.057  3.300   4.400
```

Рисунок 4.1 - Описательная статистика для второго и третьего столбцов выборки данных iris

На рисунке 4.2 представлены результаты кластеризации для переменных Sepal.Length и Sepal.Width при числе кластеров $k=3$. Центры кластеров показаны крестиками, а принадлежащие им элементы данных выделены определенным цветом.

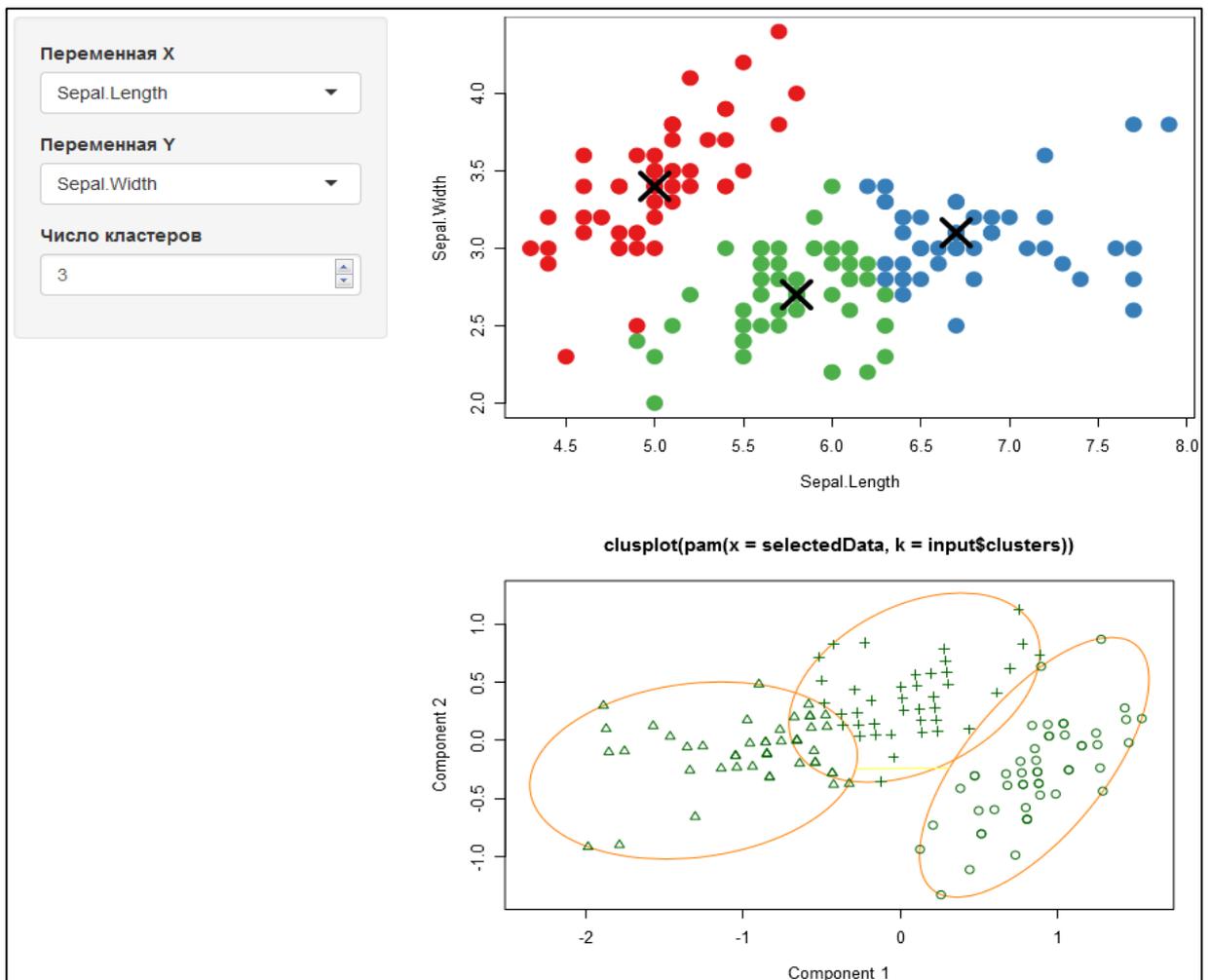


Рисунок 4.2 – Результаты кластеризации для переменных Sepal.Length и Sepal.Width, число кластеров $k=3$

```
> summary(iris[,4,5])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.100  0.300   1.300   1.199   1.800   2.500
```

На рисунках 4.3, 4.4 представлены результаты кластеризации для переменных Petal.Length и Petal.Width, при числе кластеров $k=3$ и $k=8$.

На графиках (рис. 4.3) видно, что один кластер пространственно отделен от двух других. При делении выборки на 8 кластеров (рис. 4.4) видно, что элементы класса *setosa* разбиваются на два кластера, а элементы классов *versicolor* и *virginica* делятся на шесть кластеров.

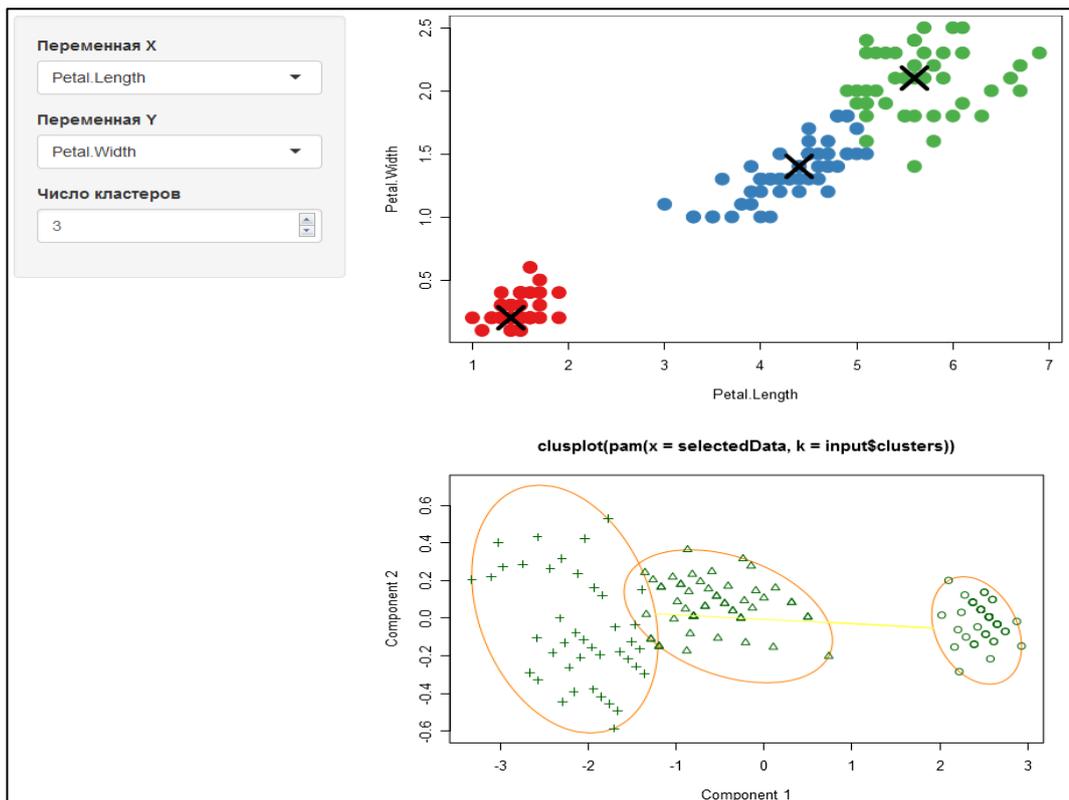


Рисунок 4.3 – Результаты кластеризации для переменных Petal.Length и Petal.Width, число кластеров k=3

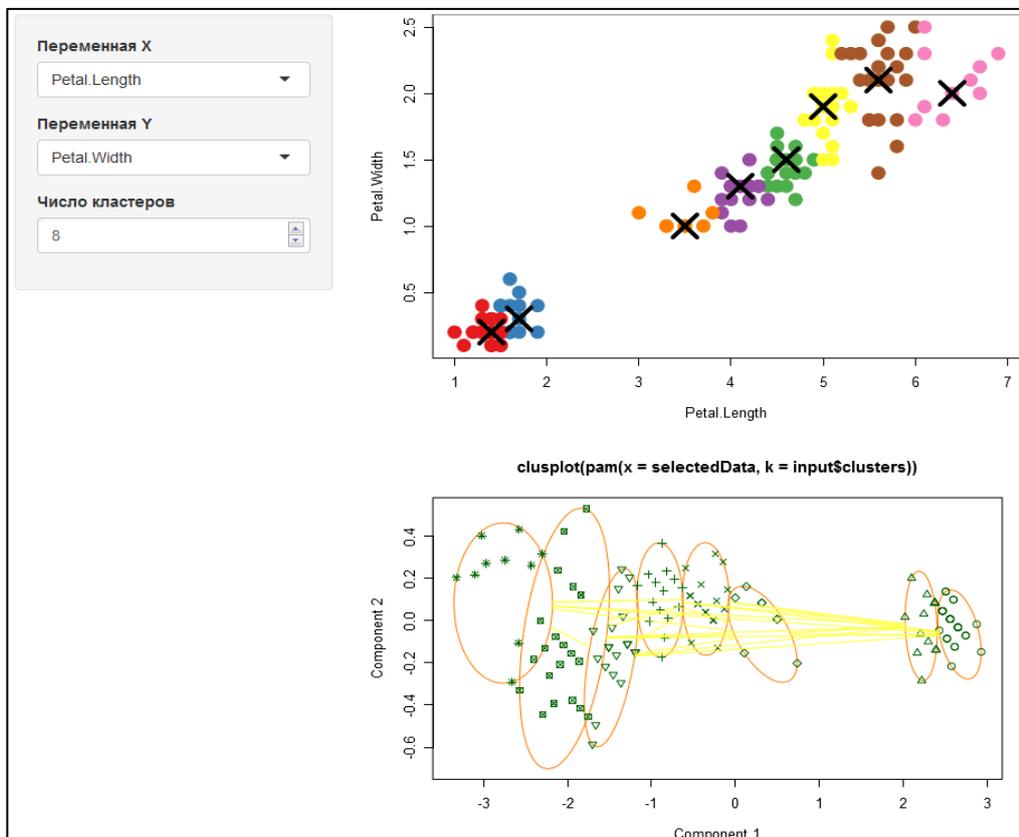


Рисунок 4.4 – Результаты кластеризации для переменных Petal.Length и Petal.Width, число кластеров k=8

Созданное приложение можно использовать в учебном процессе по дисциплинам анализа данных для иллюстрации работы алгоритмов кластеризации в реальном масштабе времени.

Что касается временной эффективности алгоритма, то время реакции сервера на действия пользователя практически мгновенное и, в основном, будет определяться пропускной способностью канала связи с Интернетом.

4.2 Тестирование и анализ эффективности интерактивного web-приложения mtcars

Данный раздел содержит результаты тестирования интерактивного приложения mtcars. Проверяется расход топлива в зависимости от трех характеристик семейств автомобилей. Ниже представлены описательная статистика и диаграммы размаха для числа миль на галлон топлива в зависимости от количества цилиндров, типа трансмиссии и числа скоростей у автомобиля (рис. 4.5 – 4.7).

```
> summary(mtcars[,2])  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
 4.000  4.000   6.000   6.188  8.000   8.000
```

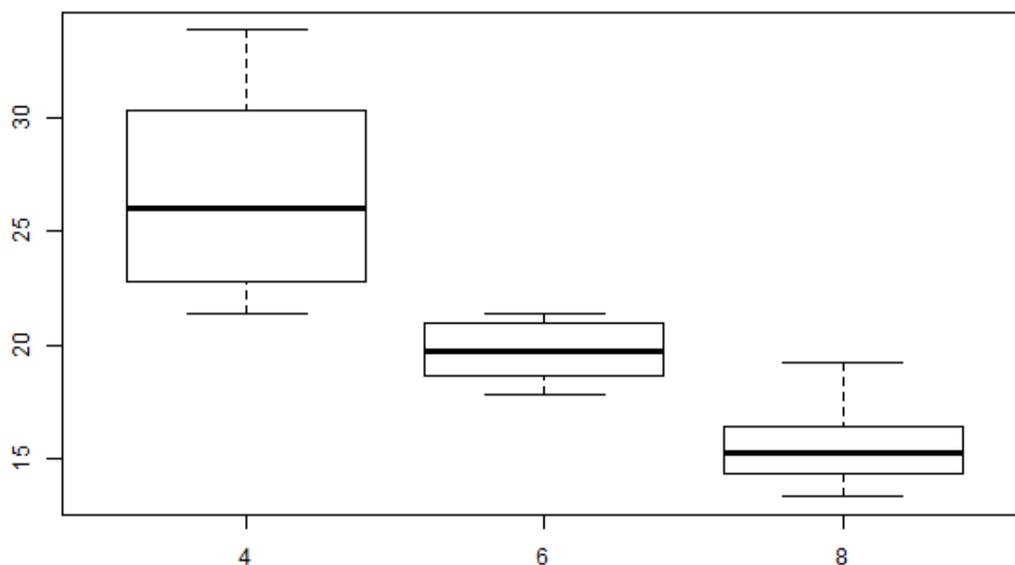


Рисунок 4.5 – Итоговая статистика и зависимость числа миль на галлон топлива от числа цилиндров автомобиля

Видно, что минимальный расход топлива наблюдается для 4-цилиндровых автомобилей, максимальный – для 8-цилиндровых.

```
> summary(mtcars[,9])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000	0.0000	0.0000	0.4062	1.0000	1.0000

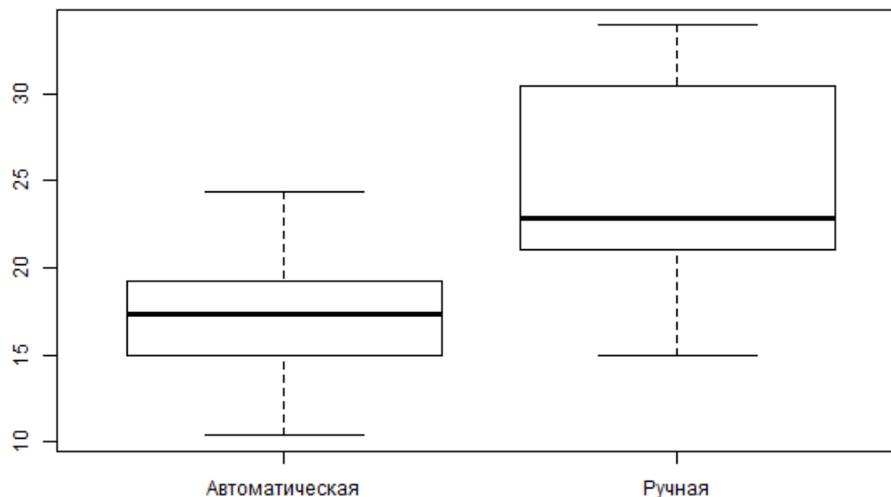


Рисунок 4.6 – Итоговая статистика и зависимость числа миль на галлон топлива от типа трансмиссии автомобиля

Из рисунка видно, что ручная трансмиссия в среднем на 35% экономичнее автоматической.

```
> summary(mtcars[,10])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.000	3.000	4.000	3.688	4.000	5.000

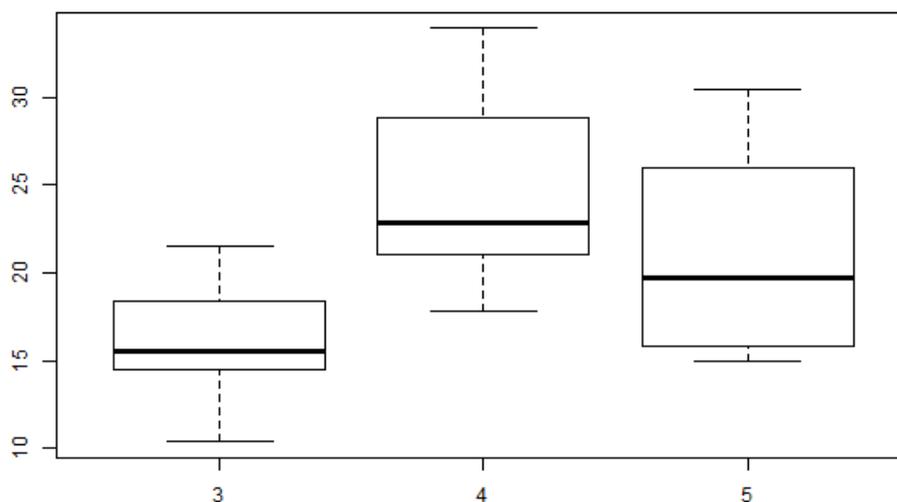


Рисунок 4.7 – Итоговая статистика и зависимость числа миль на галлон топлива от количества скоростей автомобиля

В данном случае самыми экономными являются четырехскоростные автомобили.

Что касается временной эффективности алгоритма, то время реакции сервера на действия пользователя практически мгновенное и, в основном, будет определяться пропускной способностью канала связи с Интернетом.

4.3 Апробация и оценка эффективности интерактивных учебных web-приложений

Для доказательства эффективности использования в учебном процессе разработанных интерактивных учебных web-приложений необходимо провести эксперимент.

Целью эксперимента являлась апробация разработанных интерактивных web-приложений для доказательства эффективности их использования в учебном процессе.

На основании цели эксперимента были определены следующие задачи:

1. Выбор экспериментальных групп.
2. Осуществление практического внедрения и реализации интерактивных учебных web-приложений.
3. Анализ результатов работы с использованием разработанных интерактивных учебных web-приложений.

В качестве критериев оценки хорошо построенного учебного процесса можно выделить следующие:

- посещаемость учащимися занятий;
- оценка учащимися преподаваемого материала;
- эффективность обучения учащихся.

Использование интерактивных web-приложений в учебном процессе предполагает прохождение тестирования по завершению изучения материалов курса. В связи с этим был проведен анализ того, влияет ли внедрение интерактивных web-приложений в учебный процесс, и позволяет ли это повысить показатели успеваемости обучающихся.

В двух учебных группах проводилось анкетирование студентов для определения на сколько понравился курс обучения с использованием интерактивных учебных web-приложений. Результаты анкетирования показали, что курс понравился 92% учащихся для экспериментальной группы 2 (с применением интерактивных web-технологий) и на 74% для экспериментальной группы 1 (традиционный подход, т.е. без применения интерактивных учебных web-приложений).

Множество возможных значений, которые могут наблюдаться при реализации эксперимента, образуют выборочное пространство, или, генеральную совокупность [21]. В результате эксперимента на некоторой выборке получены результаты, являющиеся основанием для суждения о генеральной совокупности.

Оценка параметров генеральной совокупности, выполненная на основании экспериментальных данных, всегда будет сопровождаться некоторой погрешностью из-за воздействия случайных вероятностных причин. Подразумевается, что не всякая статистика может служить оценкой неизвестного параметра распределения [21]. Поскольку результаты опытов случайны, любая статистика представляет собой случайную величину. В связи с этим оценки подобного рода должны быть рассмотрены как предположительные, а не как окончательные утверждения. Такие предположения о свойствах генеральной совокупности называются статистическими гипотезами. Статистической гипотезой называется утверждение, высказанное относительно распределения генеральной совокупности или некоторых его параметров. Обычно такую гипотезу обозначают как H_0 – нулевая (предложенная) гипотеза. Противоположное утверждение – отрицание гипотезы H_0 – называется конкурирующей (или альтернативной) гипотезой и обозначается как H_1 . Таким образом, в каждой конкретной ситуации можно сформулировать целое семейство различных гипотез, а для нулевой гипотезы – множество альтернативных гипотез.

Для проверки эффективности применения интерактивных технологий в учебном процессе была сформирована «нулевая гипотеза» об отсутствии отличий между учебными комплексами до и после внедрения интерактивных учебных моделей (H_0) и «альтернативная гипотеза» о наличии различий (H_1).

Критерий проверки статистической гипотезы представляет собой процедуру выработки решения о том, принять или опровергнуть данную гипотезу. Та часть выборочного пространства, которая приводит к отклонению гипотезы, называется областью принятия гипотезы, или критической областью. Граничные точки критической области называются критическими значениями. Уровень значимости гипотезы представляет собой вероятность того, что при выполнении нулевой гипотезы результаты проверок попадут в критическую область. Отклонение нулевой гипотезы в случае ее истинности называется ошибкой первого рода. Принятие нулевой гипотезы в случае, если она верна, называется ошибкой второго рода.

За низший уровень зачастую принимается значение – $P = 0,05$, за средний уровень – $P = 0,01$, а за высший уровень значимости – $P = 0,001$. Значения 0,01 и 0,05 называются стандартными уровнями статистической значимости.

В качестве эксперимента был проведен анализ статистических данных результатов работ студентов двух групп по дисциплине «Информатика 2», 30 студента которого представили экспериментальные группы исследования – определена группа испытуемых «Группа 1» из 15 человек, в которой проводились учебные занятия с использованием традиционного подхода без применения интерактивных технологий и группа испытуемых «Группа 2», также состоящая из 15 учащихся, которые проходили обучение с использованием интерактивных учебных web-приложений.

Критериями оценивания были выбраны следующие показатели:

- средний результат выполнения работ студентом без использования интерактивных web-приложений в учебном процессе;
- средний результат выполнения работ студентом, с использованием интерактивных web-приложений в учебном процессе.

Полученные статистические данные отражены в таблице 4.1.

Таблица 4.1– Статистика обучения двух групп испытуемых

№ студента	Группа 1	Группа 2
1	12	16
2	15	17
3	10	16
4	12	19
5	15	18
6	12	14
7	11	18
8	14	16
9	15	18
10	12	17
11	14	16
12	16	18
13	15	15
14	12	16
15	14	17

Данные по правильности ответов двух групп испытуемых отображены на гистограмме и представлены на рисунке 4.8. На приведенной гистограмме можно заметить, что для большинства испытуемых средний стал больше с использованием интерактивных web-приложений в учебном процессе, чем без использования.

Исходя из этого, можно сделать вывод об эффективности внедрения интерактивных web-приложений в учебный процесс, так как основной целью внедрения являлось улучшение показателей успеваемости студентов.

Для подтверждения эффективности внедрения в учебный процесс интерактивных web-приложений необходимо провести анализ полученных статистических данных. В качестве инструмента проведения статистического анализа выбрана надстройка «Пакет анализа», входящая в программу MS Excel, и предоставляющая собой набор средств для выполнения статистического анализа. В качестве метода анализа использован двухвыборочный t-тест с различными дисперсиями.

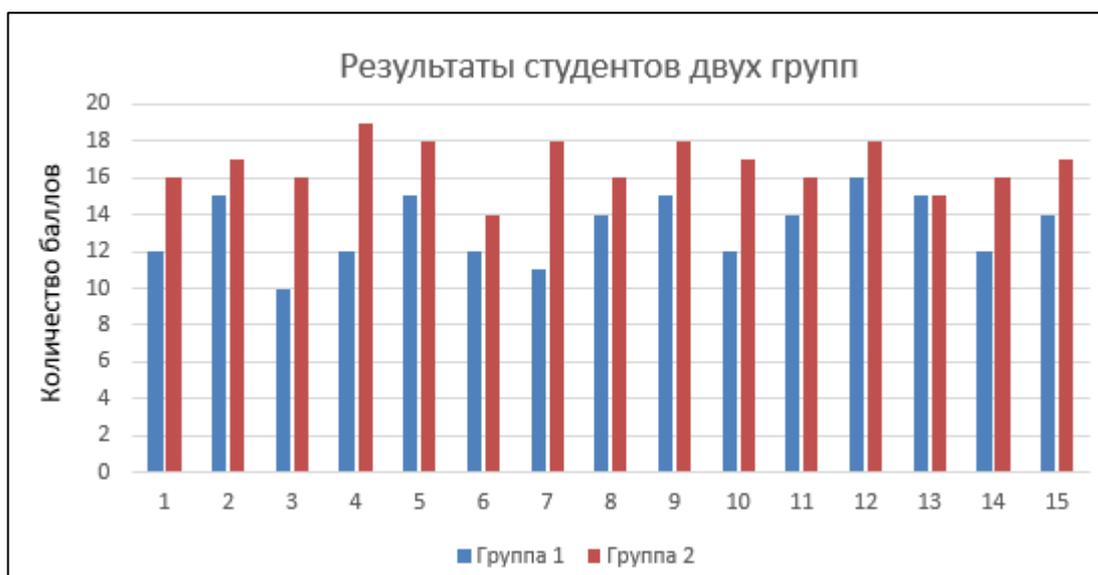


Рисунок 4.8 – Результаты студентов двух экспериментальных групп (группа 1 – без применения интерактивных web-приложений, группа 2 – с применением интерактивных web-приложений)

Двухвыборочный t-тест осуществляет проверку на равенство средних значений генеральной совокупности в каждой выборке [13]. Наиболее частые случаи применения t-теста связаны с проверкой равенства средних значений в двух выборках. Существует три вида данного теста, которые характеризуются следующими условиями:

- дисперсии распределения равны;
- дисперсии совокупности не равны;
- представление выборок до и после эксперимента по одному и тому же признаку.

Двухвыборочный t-тест с разными дисперсиями - инструмент анализа данных, который используется для проверки гипотезы о равенстве математических ожиданий для двух выборок. Не требует предположения о равенстве дисперсий генеральных совокупностей. Данный критерий также называется t-тестом или тестом Стьюдента для неравных дисперсий.

Для определения тестовой величины t используется следующая формула.

$$t = \frac{\bar{x} - \bar{y} - \Delta_0}{\frac{s_1^2}{m} + \frac{s_2^2}{n}}$$

где \bar{x} и \bar{y} – средние двух выборок X и Y,

$\Delta_0 = \bar{x} - \bar{y}$ – разность между соответствующими значениями переменной X и Y.

Для тестирования одной и той же генеральной совокупности, необходимо использовать парный тест, который используется для проверки гипотезы о равенстве математических ожиданий для двухмерной выборки данных.

Для данного исследования наиболее подходящим является «Двухвыборочный t-тест с различными дисперсиями», так как используется одна совокупность и имеются данные с проведением эксперимента и без него.

Основными значениями для данного вида анализа являются:

1. Среднее. Представляет собой выборочные средние для каждой выборки.

2. Дисперсия – несмещенные выборочные оценки дисперсий выборок.

3. Наблюдения. Представляют собой объемы выборок, то есть количество испытуемых.

4. Гипотетическая средняя разность – число, равное предполагаемой разности средних, которое задано в диалоговом окне.

5. df – число степеней свободы.

6. t-статистика – значение критериальной статистики.

7. P (T <= t) одностороннее. Данная величина представляет собой вероятность того, что наблюдаемое значение t-статистики будет отрицательным, если предположить, что средние генеральной совокупности равны, при $t < 0$. При $t \geq 0$ «P (T <=t) одностороннее» делает возможным наблюдение значения t-статистики, которое будет положительным.

8. «t критическое одностороннее» представляет пороговое значение, так что вероятность наблюдения значения t-статистики большего или равного «t критическое одностороннее».

9. «P (T ≤ t) двустороннее» показывает вероятность наблюдения значения t-статистики по абсолютному значению большего, чем t.

10. «t критическое двустороннее» показывает пороговое значение, так что значение вероятности наблюдения значения t-статистики по абсолютному значению большего, чем «t критическое двустороннее».

Результаты расчета двухвыборочного t-теста с различными дисперсиями представлены на рисунке 4.9.

Двухвыборочный t-тест с различными дисперсиями		
	Группа 1	Группа 2
Среднее	13,26666667	16,73333333
Дисперсия	3,20952381	1,780952381
Наблюдения	15	15
Гипотетическая разность средних	0	
df	26	
t-статистика	-6,010169499	
P(T≤t) одностороннее	1,19914E-06	
t критическое одностороннее	1,70561792	
P(T≤t) двухстороннее	2,39827E-06	
t критическое двухстороннее	2,055529439	

Рисунок 4.9 – Результат расчета двухвыборочного t-теста с различными дисперсиями

Полученные расчеты (рис. 4.9) показывают, что среднее первой совокупности меньше среднего второй, тем самым подтверждая общее увеличение уровня успеваемости студентов.

Также, были определены минимальный и максимальный пороги допустимого значения t-статистики, равные соответственно значениям 1.7 и 2.05. Согласно таблице критических значений рассматриваемого критерия, для вероятности 0.05 критическим значением коэффициента t-Стьюдента является значение 2.2.

Полученное значение t-статистики, равное значению 6.01, превышает максимальный порог в 2.05. В связи с этим можно сделать вывод, что различия в среднем уровне успеваемости студентов без применения интерактивных web-

приложений в учебный процесс и с применением достоверны. Таким образом, использование в учебном процессе интерактивных web-приложений оказывает влияние на успеваемость студентов, то есть по результатам t – теста нулевая гипотеза отклоняется и принимается альтернативная гипотеза.

На основании результатов апробации можно судить о том, что гипотеза, представленная на защиту магистерской диссертации, успешно подтверждена.

ЗАКЛЮЧЕНИЕ

В рамках диссертационной работы получены следующие основные результаты:

1. На основе исследования научной литературы определены требования к технологии разработки интерактивных учебных web-компонентов.

2. Исследованы возможности программной среды R совместно с пакетом расширения Shiny для разработки интерактивных учебных web-компонентов.

3. Сформулированы рекомендации по созданию интерактивных компонентов в среде R.

4. Разработаны и протестированы интерактивные учебные web-приложения, реализованные на основе классических задач.

5. Выработаны практические рекомендации и порядок работы с разработанными учебными web-приложениями.

6. Проведен педагогический эксперимент, в ходе которого были получены данные, подтверждающие правильность выбранной гипотезы.

Применение разработанных интерактивных web-компонентов в лабораторном практикуме формирует у учащихся умения выполнения модельных экспериментальных исследований, отрабатываются навыки проведения измерений и обработки информации.

Таким образом, предложенную технологию разработки интерактивных web-приложений можно использовать в любых сетевых учебных комплексах естественно-научного профиля для разработки интерактивных лабораторных работ, тренингов, моделей систем и процессов, исследовательских проектов, тестов.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Андреев, А.В. Практика электронного обучения с использованием Moodle / А.В. Андреев, С.В. Андреева, И.Б. Доценко. – Таганрог: ТТИ ЮФУ, 2008. – 146 с.
2. Асоев, Б.С. Роль дистанционного обучения в образовательном процессе // Актуальные проблемы общества в современном научном пространстве: сборник статей студентов, аспирантов, молодых ученых и преподавателей. – Уфа: Аэтерна, 2016. – С. 187-188
3. Батаев, А. В. Обзор рынка систем дистанционного обучения в России и мире // Молодой ученый, 2015, №17. - С. 433-436.
4. Баженов, Р. И., Корнилов А. П., Лопатин Д. К. Проектирование web-ориентированной информационной системы университета на основе клиент-серверных технологий // Актуальные проблемы гуманитарных и естественных наук. 2014. №4 - С. 68-71
5. Баженов, Р. И., Лопатин Д. К. О применении современных технологий в разработке интеллектуальных систем // Журнал научных публикаций аспирантов и докторантов. 2014. № 3 (93). С. 263-264.
6. Башмаков, А.И., Башмаков И.А. Разработка компьютерных учебников и обучающих систем. — М.: Информационно-издательский дом «Филинь». 2003.-616с.
7. Бершадская, Е.А. Метод интеллект-карт как инструмент визуализации когнитивных процессов учащихся / Инструментальная дидактика и дидактический дизайн: теория, технология и практика многофункциональной визуализации знаний: материалы Первой Всероссийской научно-практической конференции, Москва Уфа: БГПУ имени М. Акмулы, 2013. –290с.
8. Венэблз, У.Н., Смит Д.М. Введение в R. Заметки по R: среда программирования для анализа данных и графики. Вер. 3.1.0. Москва, 2014. 109 с.

9. Вислобоков, Н.Ю. Технологии организации интерактивного процесса обучения // Информатика и образование. 2011. - N 6. - С. 111-114.

10. Галкина, Л.А. Андрагогический подход к разработке конкурентоспособных сетевых учебно-методических комплексов для оказания образовательных услуг по профессиональной переподготовке специалистов экономического профиля // Вестник ассоциации вузов туризма и сервиса. 2011. - №4. – с.53-58.

11. Джеймс, Г., Уиттон Д., Хасты Т., Тибширани Р. Введение в статистическое обучение с примерами на языке R. Пер. С.Э. Мاستицкого. М.: ДМК Пресс, 2016. 450 с.

12. Доррер, А.Г., Иванилова Т.Н. Моделирование интерактивного адаптивного обучающего курса // Современные проблемы науки и образования. 2007. – № 5. – С. 52-59

13. Ермолаев, О.Ю. Математическая статистика для психологов / О.Ю. Ермолаев. - М.: МПСИ: Флинта. 2002. – С. 207-215.

14. Интерактивные системы Scilab, Matlab, Mathcad [Электронный ресурс]: учеб. пособие / И. Е. Плещинская [и др.]; Казанский нац. исслед. технол. ун-т. - Казань : КНИТУ, 2014. - 195 с. : ил.

15. Каменева, Т. Н. Педагогические технологии в электронном образовательном пространстве: традиции и инновации // Образовательные технологии и общество, 2013. – Т. 16. – №. 1. – С. 609-626.

16. Кит Джереми. HTML5 для веб-дизайнеров. Серия «Актуальные книги для тех, кто создает сайты», книга 6, 2012

17. Киреев, В.И. Численные методы в примерах и задачах: Учеб. пособие / В.И. Киреев, А.В. Пантелеев. – 2-е изд. стер. – М.: Высшая школа, 2006. – 480 с.: ил.

18. Куклев, В.А. Мобильное обучение: от теории к практике / Высшее образование в России, 2010, №7.

19. Левкин, Г. Г., Глухих В. Р., Базилевич С. В. Организация дистанционного обучения в профессиональной переподготовке: Аспекты

использования социальных сетей, электронных учебников и учебного сайта в дополнительном образовании. – «Scientific magazine» Kontsep, 2012

20. Лисицына, Л.С., Першин А.А. К вопросу создания Fab lab для подготовки разработчиков пользовательских веб-интерфейсов на основе HTML и CSS // Дистанционное и виртуальное обучение. 2014. №1. С. 32-38.

21. Минько, А.А. Статистика в бизнесе. Руководство менеджера и финансиста / А.А. Минько. – М.: Эксмо, 2008. – 504 с.

22. Миркин, Б. Г. Методы кластер-анализа для поддержки принятия решений: обзор: препринт WP7/2011/03 / Б. Г. Миркин; Национальный исследовательский университет «Высшая школа экономики». – М.: Изд. дом Национального исследовательского университета «Высшая школа экономики», 2011.

23. Михеева, Е. В. Информационные технологии в профессиональной деятельности. – «Издательство «Прспект», 2013.

24. Орчаков, О.А., Калмыков А.А. Проектирование дистанционных курсов. -М.: МНЭПУ,2002.-52с.

25. Осин, А.В Открытые образовательные модульные мультимедиа системы. – М.: Издательский сервис, 2010. – 328 с.

26. Панфилова, А. П. Инновационные педагогические технологии: активное обучение : учеб. пособие для студ. высш. учеб. заведений / А. П. Панфилова. - Гриф УМО. - Москва : Академия, 2009. - 191, [1] с. - (Высшее профессиональное образование. Педагогические специальности). - Библиогр.: с. 187-189.

27. Пасько, А. Моделирование и визуализация на основе функционального представления в многоуровневой образовательной практике / А. Пасько, В. Аджиев, Е. Маликова, В. Пилюгин. – Научная визуализация, т.4, №1, 2012. – с.1-11

28. Педагогические технологии: вопросы теории и практики внедрения: справ. для студентов : [учеб. пособие] / авт.-сост. А. В. Винеvская ; под общ.

ред. И. А. Стеценко. - Ростов-на-Дону : Феникс, 2014. - 254 с. - (Библиотека студента). - Библиогр.: с. 233-254.

29. Полат, Е.С. Теория и практика дистанционного обучения. Учебное пособие. М.: Академия, 2004, 416 с.

30. Сетевая образовательная среда: электронные ресурсы: учебно-методическое пособие / под ред. Т.Н. Носковой. – СПб.: Изд-во РГПУ им. А.И. Герцена, 2015. – 114с.

31. Соловов, А.В. Электронное обучение: проблематика, дидактика, технология. – Самара: «Новая техника», 2006. – 464 с.

32. Стебеньева, Т.В., Ларина Т.С., Юрятина Н.Н. Новая технология разработки электронных образовательных ресурсов как фактор информатизации сферы образования // Новые педагогические технологии. – 2014. – № 17. – С. 159–165.

33. Усмонов, М.С. Технология создания интерактивных электронных учебных курсов и ее структура // Молодой ученый, 2014. - №5. - С. 560-563.

34. Хабибуллин, Р. Опыт разработки учебных компьютерных моделей по физике в среде Adobe Flash для раздела «Электромагнитные колебания и волны» курса физики средней школы // Вестник Пермского государственного гуманитарно-педагогического университета. Серия: Информационные компьютерные технологии в образовании, 2015. - №11. - С. 94-100

35. Хортон, У., Хортон К. Электронное обучение: инструменты технологии / Пер. с англ. М.: КУДИЦ-ОБРАЗ, 2009. 640 с.

36. Шарков, Ф.И. Интерактивные электронные коммуникации: (возникновение "Четвертой волны") : учеб. пособие [для вузов] / Ф. И. Шарков. - 2-е изд. - Москва: Дашков и К°, 2010. - 259 с. : ил. - Библиогр.: с. 254-255.

Электронные ресурсы

37. Бирюкова, С. Анализ последовательностей в R: TraMineR. НИУ ВШЭ. 2014. [Электронный ресурс]. Режим доступа: URL: <https://www.hse.ru>

38. Введение в HTML5 / Миллз Крис [и др.]. Электрон. текстовые данные. М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. -

133с.- [Электронный ресурс]. Режим доступа:
<http://www.iprbookshop.ru/52143.html>.

39. ГиперМетод. Конструктор электронных курсов eAuthor. [Электронный ресурс]: Режим доступа: URL: <http://hypermethod.ru/product/2>.

40. Инькова, Н.А. Сетевой учебно-методический комплекс для начинающих разработчиков электронных учебных курсов / Н.А. Инькова, В.Е. Подольский. – [Электронный ресурс]. Режим доступа: <http://ito.edu.ru/2003/II/4/II-4-1564.html>

41. Ларина, Э.С. Создание интерактивных приложений в Adobe Flash [Электронный ресурс]/ Ларина Э.С.— Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 191 с.— Режим доступа: <http://www.iprbookshop.ru/39568.html>.

42. Полат, Е.С. Организация ДО в РФ. [Электронный ресурс]: Код доступа. URL. http://vio.uchim.info/Vio_20/cd_site/articles/art_1_20.htm

43. Редактор электронных курсов CourseLab. [Электронный ресурс]: Режим доступа: URL: <http://www.courselab.ru>.

44. Роберт, И, Кабаков. R в действии. Анализ и визуализация данных в программе R. [Электронный ресурс] / И. Роберт, Кабаков. — Электрон. дан. — М. : ДМК Пресс, 2014. — 588 с. — Режим доступа: <http://e.lanbook.com/book/58703>

45. Савельев, А.О. Проектирование и разработка веб-приложений на основе технологий Microsoft [Электронный ресурс]/ Савельев А.О., Алексеев А.А.— Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2010. — 240 с.— Режим доступа: <http://www.iprbookshop.ru/16729.html>.

46. Ульченко, Е.Н. Разработка интерактивных мультимедийных ресурсов при помощи социальных сервисов сети интернет [Электронный ресурс]: материалы научных исследований/ Ульченко Е.Н.— Электрон. текстовые данные. — Волгоград: Волгоградский государственный социально-

педагогический университет, «Перемена», 2012.- 64 с.- Режим доступа: <http://www.iprbookshop.ru/21457.html>.

47. Шипунов, А.Б. Наглядная статистика. Используем R!. [Электронный ресурс] / А.Б. Шипунов, Е.М. Балдин, П.А. Волкова, А.И. Коробейников. — Электрон. дан. — М. : ДМК Пресс, 2014. — 298 с. — Режим доступа: <http://e.lanbook.com/book/50572>

Литература на иностранном языке

48. Официальный сайт RStudio – Режим доступа: <http://shiny.rstudio.com/>

49. Data Visualization Tools for Big Data [Электронный ресурс]. – Режим доступа <https://blog.profitbricks.com/39-data-visualization-tools-for-big-data/>

50. Dean Attali. Building Shiny apps - an interactive tutorial, 2015. [Электронный ресурс]: Режим доступа: <http://deanattali.com/blog/building-shiny-apps-tutorial/>

51. F. Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, 34:1–47, 2002. [Электронный ресурс]: Режим доступа: <https://scholar.google.com/citations?user=WZBcZV4AAAAJ&hl=ru>

52. Robert I. Kabacoff. R in Action: Data analysis and graphics with R / Shelter Island, 2014

53. Zev Ross, Andrew Clark. R Powered Web Applications with Shiny, 2016. [Электронный ресурс]: Режим доступа: <https://bookdown.org/aclark/zevRossShinyBook>.

54. Garrett Golemund. The Shiny Cheat sheet, 2014. [Электронный ресурс]: Режим доступа: <https://shiny.rstudio.com/articles/cheatsheet.html>

55. Allaire, JJ, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, Aron Atkins, Rob Hyndman, and Ruben Arslan. Rmarkdown: Dynamic Documents for R., 2017.

56. Cheng Joe. MiniUI: Shiny Ui Widgets for Small Screens, 2016. [Электронный ресурс]: Режим доступа: <https://github.com/rstudio/miniUI>

57. e-Learning Software - Easygenerator URL. [Электронный ресурс]: Режим доступа: <https://www.easygenerator.com>.

ПРИЛОЖЕНИЕ А

HTML-код приложения кластеризации Ирисы Фишера

```
<!DOCTYPE html>
<html>
<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/> <script
type="application/shiny-singletons"></script> <script type="application/html-
dependencies">json2[2014.02.04];jquery[1.12.4];babel-
polyfill[6.7.2];shiny[1.0.0];jqueryui[1.12.1];showdown[0.3.1];highlight.js[6.2];selecti
ze[0.11.2];bootstrap[3.3.7];font-awesome[4.7.0]</script><script src="shared/json2-
min.js"></script>
<script src="shared/jquery.min.js"></script>
<script src="shared/babel-polyfill.min.js"></script>
<link href="shared/shiny.css" rel="stylesheet" />
<script src="shared/shiny.min.js"></script>
<script src="shared/jqueryui/jquery-ui.min.js"></script>
<script src="shared/showdown/compressed/showdown.js"></script>
<script src="shared/highlight/highlight.pack.js"></script>
<link href="shared/selectize/css/selectize.bootstrap3.css" rel="stylesheet" />
<!--[if lt IE 9]>
<script src="shared/selectize/js/es5-shim.min.js"></script>
<![endif]-->
<script src="shared/selectize/js/selectize.min.js"></script>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link href="shared/bootstrap/css/bootstrap.min.css" rel="stylesheet" />
<script src="shared/bootstrap/js/bootstrap.min.js"></script>
<script src="shared/bootstrap/shim/html5shiv.min.js"></script>
<script src="shared/bootstrap/shim/respond.min.js"></script>
```

```

<link href="shared/font-awesome/css/font-awesome.min.css" rel="stylesheet" />
<script src="shared/shiny-showcase.js"></script>
  <link rel="stylesheet" type="text/css" href="shared/highlight/rstudio.css"/>
  <link rel="stylesheet" type="text/css" href="shared/shiny-showcase.css"/>

  <title>Кластеризация цветков ирисов (алгоритм ram)</title>

</head>

<body>
  <table id="showcase-app-code">
    <tr>
      <td id="showcase-app-container" class="showcase-app-container-expanded">
        <div class="container-fluid">
          <div class="row">
            <div class="col-sm-12">
              <h1>Кластеризация цветков ирисов (алгоритм ram)</h1>
            </div>
          </div>
          <div class="row">
            <div class="col-sm-4">
              <form class="well">
                <div class="form-group shiny-input-container">
                  <label class="control-label" for="xcol">Переменная X</label>
                  <div>
                    <select id="xcol"><option value="Sepal.Length"
selected>Sepal.Length</option>
<option value="Sepal.Width">Sepal.Width</option>
<option value="Petal.Length">Petal.Length</option>
<option value="Petal.Width">Petal.Width</option>

```

```

<option value="Species">Species</option></select>
      <script type="application/json" data-for="xcol" data-
nonempty="">{ }</script>
    </div>
  </div>
  <div class="form-group shiny-input-container">
    <label class="control-label" for="ycol">Переменная Y</label>
    <div>
      <select id="ycol"><option
value="Sepal.Length">Sepal.Length</option>
<option value="Sepal.Width" selected>Sepal.Width</option>
<option value="Petal.Length">Petal.Length</option>
<option value="Petal.Width">Petal.Width</option>
<option value="Species">Species</option></select>
      <script type="application/json" data-for="ycol" data-
nonempty="">{ }</script>
    </div>
  </div>
  <div class="form-group shiny-input-container">
    <label for="clusters">Число кластеров</label>
    <input id="clusters" type="number" class="form-control" value="3"
min="1" max="9"/>
  </div>
</form>
</div>
<div class="col-sm-8">
  <div id="plot1" class="shiny-plot-output" style="width: 100% ; height:
400px"></div>
  <div id="plot2" class="shiny-plot-output" style="width: 100% ; height:
400px"></div>

```

```

    </div>
  </div>
</div>
<td id="showcase-sxs-code" class="showcase-sxs-code-collapsed"></td>
</td>
</tr>
</table>
<div class="container-fluid shiny-code-container well" id="showcase-well">
  <div class="row">
    <div id="showcase-app-metadata" class="col-sm-4">
      <h4 class="text-muted shiny-showcase-apptitle">
        Pam example
      <small>
        <br/>
        by
        <a href="http://www.rstudio.com/" class="shiny-showcase-appauthor">Joe
Cheng &lt;joe@rstudio.com&gt;</a>

      </small>
    </h4>
  </div>
  <div id="showcase-code-inline" class="col-sm-8">
    <div id="showcase-code-tabs">
      <a id="showcase-code-position-toggle" class="btn btn-default btn-sm"
onclick="toggleCodePosition()">
        <i class="fa fa-level-up"></i>
        show with app
      </a>
      <ul class="nav nav-tabs">
        <li class="active">

```

```

    <a href="#server_R_code" data-toggle="tab">server.R</a>
  </li>
  <li class="">
    <a href="#ui_R_code" data-toggle="tab">ui.R</a>
  </li>
</ul>
<div class="tab-content" id="showcase-code-content">
  <div class="tab-pane active" id="server_R_code">
    <pre class="shiny-code"><code class="language-r">library(cluster)

```

```

function(input, output) {
  output$plot1 &lt;- renderPlot({
    palette(c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3",
              "#FF7F00", "#FFFF33", "#A65628", "#F781BF", "#999999"))

    par(mar = c(5.1, 4.1, 0, 1))
    selectedData &lt;- iris[, c(input$xcol, input$ycol)]
    clusters &lt;- pam(selectedData, input$clusters)
    plot(selectedData,
          col = clusters$cluster,
          pch = 20, cex = 3)
    points(clusters$medoids, pch = 4, cex = 4, lwd = 4)
  })
  output$plot2 &lt;- renderPlot({
    selectedData &lt;- iris[, c(input$xcol, input$ycol)]
    clusters1 &lt;- pam(selectedData, input$clusters)
    clusplot(clusters1)
  })
}

```

```

</code></pre>
  </div>
  <div class="tab-pane" id="ui_R_code">
    <pre class="shiny-code"><code class="language-r">pageWithSidebar(
headerPanel('Кластеризация цветков ирисов (алгоритм pam)'),
sidebarPanel(
  selectInput('xcol', 'Переменная X', names(iris)),  selectInput('ycol', 'Переменная
Y', names(iris), selected=names(iris)[[2]]),
  numericInput('clusters', 'Число кластеров', 3, min = 1, max = 9)
),
mainPanel(
  plotOutput('plot1'),
  plotOutput('plot2')
)
)
</code></pre>
  </div>
  </div>
  <small class="showcase-code-license text-muted">
    Code license:
    <a href="http://www.r-project.org/Licenses/MIT">MIT</a>
  </small>
  </div>
  </div>
  </div>
  </div>
  </div>
</body>

</html>

```

ПРИЛОЖЕНИЕ Б

HTML-код приложения mtcars

```
<!DOCTYPE html>
<html>
<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/> <script
type="application/shiny-singletons"></script> <script type="application/html-
dependencies">json2[2014.02.04];jquery[1.12.4];babel-
polyfill[6.7.2];shiny[1.0.0];selectize[0.11.2];bootstrap[3.3.7]</script><script
src="shared/json2-min.js"></script>
<script src="shared/jquery.min.js"></script>
<script src="shared/babel-polyfill.min.js"></script>
<link href="shared/shiny.css" rel="stylesheet" />
<script src="shared/shiny.min.js"></script>
<link href="shared/selectize/css/selectize.bootstrap3.css" rel="stylesheet" />
<!--[if lt IE 9]>
<script src="shared/selectize/js/es5-shim.min.js"></script>
<![endif]-->
<script src="shared/selectize/js/selectize.min.js"></script>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link href="shared/bootstrap/css/bootstrap.min.css" rel="stylesheet" />
<script src="shared/bootstrap/js/bootstrap.min.js"></script>
<script src="shared/bootstrap/shim/html5shiv.min.js"></script>
<script src="shared/bootstrap/shim/respond.min.js"></script> <title>Расход
топлива</title>

</head>

<body>
```

```

<div class="container-fluid">
  <div class="row">
    <div class="col-sm-12">
      <h1>Расход топлива</h1>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-3">
      <form class="well">
        <div class="form-group shiny-input-container">
          <label class="control-label" for="variable">Переменная:</label>
          <div>
            <select id="variable"><option value="cyl" selected>Cylinders</option>
<option value="am">Transmission</option>
<option value="gear">Gears</option></select>
            <script type="application/json" data-for="variable" data-
nonempty="">{ }</script>
          </div>
        </div>
      </div>
    <div class="form-group shiny-input-container">
      <div class="checkbox">
        <label>
          <input id="outliers" type="checkbox"/>
          <span>Показать выбросы</span>
        </label>
      </div>
    </div>
    <b>mpg</b> - miles per gallon (милей за галлон)<br>
    <b>cyl</b> - число цилиндров<br>
    <b>am</b> - трансмиссия<br>

```

`gear` - коробка передач

`drat` - коэффициент задней оси

`wt` - вес

`qsec` - время 1/4 мили

`</form>`

`</div>`

`<div class="col-sm-8">`

`<h3>`

Зависимость

``

`</h3>`

`mtcars` - выборка данных из американского журнала 'Motor Trend',

содержит

сведения о потреблении топлива в зависимости от 10 параметров для 32 автомобилей

(модели 1973–74 гг.):

`<div id="view" class="shiny-html-output"></div>`

`<pre id="summary" class="shiny-text-output noplaceholder"></pre>`

`<div id="mpgPlot" class="shiny-plot-output" style="width: 100% ; height: 400px"></div>`

`</div>`

`</div>`

`</div>`

`</body>`

`</html>`