



УНИВЕРСИТЕТ ИТМО

Г.М. ОРЛОВ, О.А. ИГНАТЬЕВА,
А.Г. ВАСИЦ, Б.А. НИЗОМУТДИНОВ

СОВРЕМЕННЫЕ МЕТОДЫ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ

УПРАВЛЕНИЕ ГОСУДАРСТВЕННЫМИ ИНФОРМАЦИОННЫМИ СИСТЕМАМИ

Санкт-Петербург 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**Г. М. Орлов, О. А. Игнатьева, А. Г. Васин,
Б. А. Низомутдинов**

СОВРЕМЕННЫЕ МЕТОДЫ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлениям подготовки 09.04.03 Прикладная информатика
и 41.06.01 Политические науки и регионоведение
в качестве учебного пособия для реализации основных профессиональных
образовательных программ высшего образования магистратуры и аспирантуры

 УНИВЕРСИТЕТ ИТМО

**Санкт-Петербург
2021**

Орлов Г. М., Игнатъева О. А., Васин А. Г., Низомутдинов Б. А. **Современные методы обработки и анализа данных.** – СПб.: Университет ИТМО, 2021. – 147 с.

Рецензент:

Ковальчук Сергей Валерьевич, кандидат технических наук, Национальный центр когнитивных разработок, старший научный сотрудник, доцент факультета цифровых трансформаций Университета ИТМО.

Учебное пособие посвящено современным методам обработки данных и предназначено для магистрантов, обучающихся на образовательной программе «Умный город и урбанистика» по специализации «Управление государственными информационными системами» в рамках дисциплины «Управление на основе данных: методы и технологии» (направление подготовки 09.04.03 «Прикладная информатика»). Пособие также рекомендовано аспирантам (направление подготовки 41.06.01 «Политические науки и регионоведение»), применяющих в своей работе современные методы обработки данных. Учебное пособие состоит из двух разделов. В первом разделе рассматриваются методы работы со средними выборками данных посредством языка программирования R, программного обеспечения для проведения сетевого анализа Rајек, также рассматриваются технологии парсинга. Второй раздел посвящен формированию у студентов навыков работы с большими данными. Доступ к файлам, указанным в пособии, обеспечивается на информационном портале Центра технологий электронного правительства Института дизайна и урбанистики Университета ИТМО (регистрация обучающихся — <https://news.egov.itmo.ru/for-students>).



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2021

© Г.М. Орлов, О.А. Игнатъева, А.Г. Васин,
Б.А. Низомутдинов, 2021

Содержание

ВВЕДЕНИЕ	7
1. ОСНОВЫ РАБОТЫ С ДАННЫМИ	9
1.1. ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСНОВАНИЯ РАБОТЫ С ДАННЫМИ	9
ДАННЫЕ КАК ОБЪЕКТ ИЗУЧЕНИЯ.....	9
МЕТОДЫ РАБОТЫ С ДАННЫМИ	13
УРОВНИ И ВИДЫ СОЦИАЛЬНЫХ ИССЛЕДОВАНИЙ.....	17
РАЗРАБОТКА ПРОГРАММЫ ЭМПИРИЧЕСКОГО ИССЛЕДОВАНИЯ.....	22
ОСНОВНЫЕ ОШИБКИ ПРИ СБОРЕ ДАННЫХ	25
ПРАКТИЧЕСКОЕ ЗАДАНИЕ	26
1.2 СТАТИСТИКА С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ R	27
ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ R.....	27
ОПИСАТЕЛЬНАЯ СТАТИСТИКА С ИСПОЛЬЗОВАНИЕМ R.....	32
ПОСТРОЕНИЕ МОДЕЛИ МНОЖЕСТВЕННОЙ РЕГРЕССИИ НА ЯЗЫКЕ R.....	34
ПРАКТИЧЕСКОЕ ЗАДАНИЕ	43
1.3. СЕТЕВОЙ АНАЛИЗ С ИСПОЛЬЗОВАНИЕМ RAJЕК	44
ПОСТРОЕНИЕ СЕТИ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ RAJЕК	44
ИСПОЛЬЗОВАНИЕ КЛАССИФИКАЦИЙ ДЛЯ УПОРЯДОЧИВАНИЯ ДАННЫХ.....	49
ОПРЕДЕЛЕНИЕ ПЛОТНЫХ УЧАСТКОВ СЕТИ ПРИ ПОМОЩИ RAJЕК	51
РАСЧЕТЫ ЦЕНТРАЛЬНОСТЕЙ ВЕРШИН И ЦЕНТРАЛИЗАЦИИ СЕТИ	53
ПРАКТИЧЕСКОЕ ЗАДАНИЕ	55
1.4. СБОР ТЕКСТОВЫХ ДАННЫХ	56
ВИДЫ ДАННЫХ, ГЕНЕРИРУЕМЫХ ПОЛЬЗОВАТЕЛЯМИ, ПОДХОДЫ К ИХ ПОЛУЧЕНИЮ И СИСТЕМАТИЗАЦИИ	56
СБОР ДАННЫХ ЧЕРЕЗ API.....	56

ПРИМЕР РАБОТЫ С API ДЛЯ ВЫГРУЗКИ ДАННЫХ ИЗ ЭЛЕКТРОННОЙ БИБЛИОТЕКИ ELIBRARY.RU	57
ПРИМЕР РАБОТЫ С API ДЛЯ ВЫГРУЗКИ ДАННЫХ TWITTER	60
СБОР ДАННЫХ БЕЗ ИСПОЛЬЗОВАНИЯ API	61
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПАРСИНГА САЙТОВ	64
ПРИМЕР. ПАРСИНГ ПУБЛИКАЦИЙ С ИНФОРМАЦИОННОГО ПОРТАЛА.....	68
ПАРСИНГ СОЦИАЛЬНЫХ СЕТЕЙ.....	74
ОГРАНИЧЕНИЕ НА СКАЧИВАНИЕ ИНФОРМАЦИИ.....	74
2. ОСОБЕННОСТИ РАБОТЫ С «БОЛЬШИМИ ДАННЫМИ»	76
2.1 ОСНОВНЫЕ ПОДХОДЫ К ПОСТАНОВКЕ ЗАДАЧИ ПРИ СОЗДАНИИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИХ СИСТЕМ В ЗАДАЧАХ ОБРАБОТКИ ДАННЫХ В СФЕРЕ ЗДРАВООХРАНЕНИЯ	76
БАЗОВЫЕ ИНСТРУМЕНТЫ СОСТАВЛЕНИЯ ТРЕБОВАНИЙ ЗАКАЗЧИКА К ИАС	78
<i>Пользовательская история</i>	78
<i>Сценарий использования</i>	80
<i>Скриншоты</i>	85
СПЕЦИФИЧНЫЕ ИНСТРУМЕНТЫ ДЛЯ ИАС И ВІ.....	85
<i>Примеры</i>	86
АНАЛИЗ ИСТОЧНИКА ДАННЫХ	86
СОЗДАНИЕ ТЕРМИНОЛОГИЧЕСКОЙ БАЗЫ АНАЛИТИЧЕСКОГО РЕШЕНИЯ И ИЗУЧЕНИЕ НОРМАТИВНО-ПРАВОВОЙ БАЗЫ	88
<i>Примеры информационно-логических моделей ИАС</i>	90
2.2 ПРОЕКТИРОВАНИЕ ВИТРИН ДАННЫХ АНАЛИТИЧЕСКОГО РЕШЕНИЯ В КОЛОНОЧНЫХ СУБД	92
БАЗОВЫЕ ВИДЫ АНАЛИТИЧЕСКИХ РЕШЕНИЙ	92
ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ПЛОСКИХ ВИТРИН БОЛЬШИХ ДАННЫХ.....	94
ETL - ПРОЦЕСС ИЗВЛЕЧЕНИЯ, ПРЕОБРАЗОВАНИЯ И ЗАГРУЗКИ ДАННЫХ	95

ДОКУМЕНТИРОВАНИЕ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОГО РЕШЕНИЯ	95
<i>Примеры проектов витрин данных</i>	96
<i>Практическое задание</i>	96

2.3 ИСПОЛЬЗОВАНИЕ ВИТРИН ДАННЫХ ДЛЯ НАСТРОЙКИ И ВИЗУАЛИЗАЦИИ ОТЧЁТОВ.....98

СУБД SLICKHOUSE. КЛИЕНТ DBeaver для обращения к БД.	
СОСТАВЛЕНИЕ SQL ЗАПРОСОВ К ВИТРИНЕ ДАННЫХ	98
<i>Начало работы с DBeaver</i>	98
<i>Выполнение SQL запросов</i>	103
БАЗОВЫЕ SQL КОНСТРУКЦИИ В ТЕРМИНАХ КОЛОНОЧНОЙ БД SLICKHOUSE	105
<i>Синтаксис конструкции SELECT</i>	105
<i>COUNT, UNIQ и другие агрегатные функции</i>	106
<i>JOIN. Конструкция для объединения таблиц данных</i>	106
<i>UNION. Конструкция для дополнения таблицы данными другого запроса</i>	107
<i>Практические задания</i>	107
<i>Пример</i>	108
СОСТАВЛЕНИЕ SQL ЗАПРОСОВ К ВИТРИНЕ ДАННЫХ. ПРОДВИНУТЫЕ ВЫБОРКИ ДАННЫХ	110
<i>Запросы над непересекающимися параметрами</i>	110
<i>Использование подзапросов в фильтрах</i>	114
ВИЗУАЛИЗАЦИЯ ДАННЫХ	116
<i>Примеры</i>	118
<i>Практическое задание</i>	120
ПРИМЕР. ПОКАЗАТЕЛИ НАЦИОНАЛЬНОГО ПРОЕКТА СОЗДАНИЕ ЕДИНОГО ЦИФРОВОГО КОНТУРА В СФЕРЕ ЗДРАВООХРАНЕНИЯ	121

2.4 ПРОВЕРКА ИАС НА ТРЕБУЕМУЮ ФУНКЦИОНАЛЬНОСТЬ И КАЧЕСТВО ДАННЫХ	124
Виды тестирования	124
Распространённые ошибки/замечания к юнит тесткейсам	126
<i>Практическое задание</i>	127
<i>Комплексный пример создания и тестирования витрины данных учёта движения коечного фонда</i>	127
2.5 СВОДНЫЕ ПАНЕЛИ РУКОВОДИТЕЛЕЙ КАК ИНСТРУМЕНТ УПРАВЛЕНИЯ НА ОСНОВЕ ДАННЫХ СИСТЕМОЙ ЗДРАВООХРАНЕНИЯ	129
2.6 ПОДХОДЫ К ОЦЕНКЕ ЦИФРОВОЙ ЗРЕЛОСТИ СИСТЕМ СОЦИАЛЬНОЙ СФЕРЫ. ИНДЕКСЫ СИМИС И СЕЗАМ	134
Обзор подходов к оценке уровня информатизации социальной сферы	134
Предлагаемый подход к оценке цифровой зрелости	136
Оценка цифровой зрелости сервиса записи на прием к врачу	136
<i>Пример расчета СЕЗАМ по Санкт-Петербургу</i>	137
ПРИЛОЖЕНИЕ	139
Перечень примеров на внешних носителях	139
Перечень таблиц	139
Перечень рисунков	139
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	143

Введение

Пособие подготовлено междисциплинарным коллективом и ориентировано на освоение современных методов обработки и анализа данных в сфере управления государственных информационных систем и процессов, связанных с развитием электронного правительства и цифровизацией всех отраслей. В качестве предметного поля для демонстрации специфики применения методов обработки и интерпретации информации используются кейсы в здравоохранении и социальной сфере.

Любое научное исследование проводится в рамках определенной методологии, которая предполагает не только грамотное составление программы исследования, но и выбор теоретической концепции для интерпретации данных. По мнению Э. Г. Юдина можно выделить четыре уровня методологии: методический, конкретно-научный, общенаучный и философский. К сожалению, очень часто при обучении студентов преподаватель останавливается на методическом уровне, т.е. в рамках подготовки программы исследования. В рамках пособия мы осветим вопросы научных парадигм социальных и политических наук, поднимемся на общенаучный уровень и поговорим о философских принципах научного познания. Таким образом, мы представим системный взгляд на подготовку к качественному проведению междисциплинарного исследования.

Междисциплинарность определяет и ориентацию пособия на два направления подготовки – прикладная информатика и политология, а также две аудитории – студентов магистратуры и аспирантов. Учебное пособие предназначено для магистрантов, обучающихся на образовательной программе «Умный город и урбанистика» по специализации «Управление государственными информационными системами» в рамках дисциплины «Управление на основе данных: методы и технологии» (направление подготовки 09.04.03 «Прикладная информатика») а также для аспирантов-политологов (направление подготовки 41.06.01 «Политические науки и регионоведение»), осваивающих современные методы обработки и анализа данных в рамках дисциплины «Политические институты, процессы и технологии». Актуальность данной темы для политологов подтверждается необходимостью развития навыков и компетенций, необходимых для проведения современных политологических исследований - проектирования и осуществления комплексных междисциплинарных научных исследований в сфере политических наук, в том числе с применением компьютерных методов извлечения и анализа больших данных, сетевого анализа, использования различных программных средств и систем.

В первом разделе данного пособия будут рассмотрены статистические методы анализа количественных и номинальных данных с использованием языка программирования R с точки зрения его возможностей в области визуализации данных, построения регрессионных моделей для разного типа предикторов. Также в данном разделе учебного пособия представлен метод сетевого анализа данных,

возникший в рамках социальных наук. Например, в политологии данный метод является отражением новейшей конкретно-научной парадигмы сетевого анализа. Сетевой анализ позволяет выстраивать сети как формальных, так и неформальных связей между различными акторами с учетом их близости друг к другу и разного рода ролей в сети. Программное обеспечение *Paјek*, представленное в данном пособии, как одна из программ, позволяющая проводить сетевой анализа, дает возможность визуализировать данные в виде сети и производить расчет разных количественных зависимостей между акторами.

Для анализа явлений и процессов, для поиска связей и закономерностей требуется большой объем открытой информации. Чтобы работать с такими данными, на первом шаге их важно собрать. В своей работе исследователь должен понимать основные подходы к сбору открытых данных, а также какие существуют программные алгоритмы для решения поставленной задачи, в первую очередь речь идет о парсинге. Поэтому важно овладеть основами автоматизированного сбора информации из сети. Даже если самостоятельно не получится автоматизировано собрать информацию из сети, обладая базовыми знаниями, можно подготовить качественное техническое задание для программиста. Дополнительно понимание принципов парсинга позволит заранее спланировать исследование: уже на первом шаге своей работы можно искать источники, которые оптимально подходят для парсинга.

Во втором разделе настоящего пособия будут рассмотрены особенности работы с большими данными, формирующимися в промышленных, многопользовательских информационно-аналитических системах (далее - ИАС), основные этапы и инструменты аналитика для проектирования.

Будут разобраны базовые понятия для проектирования ИАС, рассмотрена работа с заказчиком аналитических продуктов с использованием таких инструментов, как опрос, составление пользовательских историй, сценариев использования, раскрыта важность создания терминологической базы и изучения нормативно-правовой основы, а также рассмотрены вопросы проектирования и использования витрин данных аналитического решения в колоночных СУБД, разобраны основы SQL запросов с примерами на свободно-распространяемом ПО, проверки ИАС на требуемую функциональность и качество данных.

В разделе будут рассмотрены инструменты управления на основе данных на примере в сфере здравоохранения и подходы к оценке цифровой зрелости информационных систем социальной сферы.

В материал раздела включены практические задания, шаблоны и образцы документов на примере задач обработки данных в сфере здравоохранения.

Доступ к файлам, указанным в пособии, обеспечивается на информационном портале Центра технологий электронного правительства Института дизайна и урбанистики Университета ИТМО (регистрация обучающихся — <https://news.egov.itmo.ru/for-students>).

1. Основы работы с данными

1.1. Теоретико-методологические основания работы с данными

Данные как объект изучения

Статистика была первой научной отраслью, которая занималась обработкой социальных данных. Данные для нее собирались в основном путем анкетирования. Разрабатывались разные способы формирования репрезентативной выборки, которая бы корректно отражала весь массив данных.

Составление репрезентативной выборки – это отдельный раздел в методологии научного исследования [1, С. 29-35]. Только вероятностные выборки, составленные с учетом требований теории вероятности, могут служить основанием для экстраполяции результатов исследования на всю генеральную совокупность [2, Р. 210-228]. Составление вероятностных выборок требует наличия списка элементов, входящих в генеральную совокупность, что не всегда возможно. В связи с этим мы начнем с описания выборок, которые сформированы без учета требований теории вероятности, и поговорим об их ограничениях.

Первым типом формирования выборки без учета требований теории вероятности является выборка по случаю. Например, мы можем сформировать такую выборку, останавливая прохожих у входа на рынок в обеденное время. Такая выборка оправдана только в том случае, если мы хотим выяснить характеристики людей, посещающих рынок в данный момент времени, или если у нас нет возможности составить другую, более репрезентативную выборку. Обычно к такому способу формирования выборки прибегают преподаватели, которые опрашивают студентов, посещающих поточные лекции. Такая выборка может позволить протестировать инструментарий, но такой метод формирования выборки нельзя использовать с целью обобщения выводов на генеральную совокупность.

Вторым типом выборки, не основанной на теории вероятности, служит целевая (оценочная) выборка. Элементы для такой выборки отбираются на основании суждения о том, какие из них более соответствуют целям исследования. Данная выборка может быть использована, когда мы хотим изучить небольшое подмножество элементов более крупной совокупности, которые легко идентифицировать, но сложно перечислить. Иногда целевая выборка может быть использована для анализа девиантных кейсов, чтобы лучше понять типичные кейсы. Данная выборка подходит для тестирования инструментария или разведывательного исследования. Экстраполяцию выводов на генеральную совокупность лучше не делать.

Третьим типом «невероятностной» выборки является выборка по типу «снежного кома». Выборка по типу «снежного кома» может быть использована для

разведывательного исследования, когда нет возможности перечислить элементы, входящие в генеральную совокупность, и сложно получить доступ к представителям данной социальной группы, например, при исследовании проблемы бездомности [3, С. 48-51].

Четвертым типом выборки, в основе которой нет положений теории вероятностей, является квотная выборка. Как и в случае с вероятностной выборкой, квотная выборка отвечает требованиям репрезентативности. Для формирования данной выборки используется таблица, в каждой из ячеек которой прописана пропорция населения, соответствующая определенной возрастной группе, полу или классу. Квотная выборка формируется в соответствии с требованиями репрезентативности и повторяет то же процентное соотношение внутри выборки, как и в генеральной совокупности. Однако при использовании квотной выборки могут возникнуть такие проблемы, как своевременное обновление данных в ячейках, что может быть трудозатратным, и могут быть отклонения в характеристиках респондентов, связанные с поведением интервьюеров. Впервые квотная выборка была применена институтом исследования общественного мнения Гэллапа в США и была значительным шагом на пути к более точному предсказанию результатов выборов Президента. Так, благодаря данному типу выборки, Гэллап смог точно предсказать результаты выборов Президента США в 1941 г.

Научно обоснованные выборки, результаты анализа которых можно распространять на генеральную совокупность, должны исходить из положений теории вероятностей, которая позволяет придать репрезентативный характер исследуемой выборки [4, С. 41-46]. Всего существует четыре типа вероятностных выборок: простая случайная выборка, систематическая выборка, стратифицированная выборка и многоступенчатая кластерная выборка [5, С. 197-220]. В основе этих выборок лежит необходимость использования нумерованного списка для элементов генеральной совокупности. Так, для простой случайной выборки нам необходимо использовать таблицу случайных чисел. Например, у нас есть генеральная совокупность размером одна тысяча учащихся определенного вуза. Нам нужно сделать выборку, состоящую из ста человек. Мы нумеруем все элементы генеральной совокупности и затем на основе таблицы случайных чисел отбираем сто элементов для выборки.

В случае с систематической выборкой мы отбираем каждую k -ую единицу в списке и начинаем прохождение этого списка со случайного элемента. Например, если наша генеральная совокупность составляет одну тысячу человек, а выборка должна быть сто человек, то выборочный интервал составит десять. Мы будем отбирать из списка каждый десятый элемент. Данный дизайн менее трудоемкий, чем использование простой случайно выборки, и дает схожие результаты. Но при использовании систематической выборки мы можем столкнуться с проблемой цикличности данных, если данные в списке организованы в табличном виде,

например, перечисление военнослужащих может быть сформировано в виде отрядов по десять человек, начиная с сержантов. Тогда у нас в выборке могут оказаться одни сержанты, и выборка будет чрезмерно направленной. Таким образом, нужно внимательно смотреть за дизайном организации данных в списке.

Стратифицированная выборка может расцениваться как модификация для трех других вероятностных выборок. Она позволяет достигнуть большей репрезентативности данных. Стратифицированная выборка – это группировка единиц, составляющих генеральную совокупность в однородные группы (страты) перед осуществлением отбора данных. Есть два способа осуществления стратифицированного отбора данных. Во-первых, мы можем распределить данные на две группы, например, по полу студентов, и затем на основе таблицы случайных чисел отобрать нужные элементы из этих двух групп. Во-вторых, мы можем распределить данные по группам, записать эти группы друг под другом и затем сделать из этих групп систематическую выборку.

И, наконец, мы можем сформировать многоступенчатую кластерную выборку, которая представляет собой движение от больших групп к индивидам, но без использования большого списка элементов. Это имеет смысл, когда нам нужно проанализировать жителей какого-нибудь города. Для этого, например, мы составляем список жилых домов, затем случайным образом отбираем из них выборку из ста домов. После этого мы составляем список домохозяйств по фамилиям хозяев, отбираем сто домохозяйств, составляем список членов домохозяйств. После этого мы из списка случайным образом отбираем членов семей для опроса [6, С. 23-24].

Таким образом, проведение социального исследования должно подчиняться строгим критериям научности как на этапе формирования программы исследования и отбора дизайна для выборки данных из генеральной совокупности, так и на этапе проведения исследования с последующей интерпретацией данных на основе выбранной в начале исследования теоретической рамки.

За несколько последних десятилетий такие физические объекты, как бумажные книги, видеокассеты и музыкальные диски, традиционные фотографии на фотопленке, настольные игры, талоны на запись к врачу, результаты инструментальных медицинских исследований, перешли из физической или электронной аналоговой формы в единую цифровую форму. Этот процесс позволил отделить понятие самих данных от их физических носителей. Теперь мы можем рассматривать в каждом из ранее различных физических объектов сами данные, изучать и анализировать едиными методами и средствами обработки с целью принятия управленческих решений или работы окружающих нас автоматизированных и автоматических систем. С этой потребностью обработки и анализа цифровых данных связано самое активное развитие дисциплины работы с данными, рост интереса к этой теме, количества специалистов, изучающих и применяющих эти методы, множества публикаций.

Социальные сети, мобильная связь, автоматизированные системы в нефтегазовой сфере, здравоохранении и других отраслях экономики генерируют огромные массивы информации, которые могут быть сохранены и обработаны с помощью современных компьютерных средств и сетей передачи информации.

Компания Gartner (США) классифицирует “большие данные” следующим образом:

1. Операционные данные. Сведения, получаемые в процессе регулярной деятельности организаций. Различные логи и транзакционные данные. Кассовые чеки, записи к врачу, случаи медицинского обслуживания, структурированные электронные документы и другие. Данные ERP, CRM, централизованных систем.

2. «Темные» данные. Информация, которая не накапливается специально. К примеру, электронные письма, переписка в мессенджерах.

3. Коммерческие данные. К примеру, данные маркетинговых исследований, кредитные рейтинги.

4. Официальные данные. Правительственные данные, отчёты, официальная статистика и другие.

5. Данные социальных сетей.

В распространённом мнении к большим данным относят такие объёмы информации, обработать которые на одном компьютере невозможно или затруднительно. Однако к этому простому, эмпирическому определению можно задать много вопросов. К примеру, следующие:

- Обработать за сколько времени? Секунда, минута, час, день неделя? Обработать онлайн или из некоторого среза данных?

- Что такое один компьютер? В современных центрах обработки данных редко выделяются физически-конкретные процессоры и жёсткие диски. Чаще всего – виртуальные машины.

- Обработать какие данные? Первичные, неагрегированные данные или предобработанные? Текстовые, числовые, изображения, структурированные или неструктурированные?

Ответ на подобные вопросы сформулировал Дуг Лэйни, работая в Meta Group (выкуплена компанией Gartner) – базовые характеристики больших данных определяются через три V: объём информации (volume), скорость обработки данных (velocity) и многообразие (variety). Впоследствии характеристики были расширены другими важными составляющими, к примеру, ценность информации.

По-другому “большие данные” можно определить как те, с которыми не справляются традиционные системы управления базами данных (СУБД). Сравнение традиционных баз данных (БД) и технологий Big data приведено в таблице, см. Таблица 1.

Таблица 1. Сравнение характеристик традиционных баз данных и технологий Big data

Характеристика	Традиционные БД	Big data
Объём информации	Мб, Гб	Тб, Пб, Эб
Гранулированность данных	Структурированные	Структурированные, неструктурированные, частично структурированные
Способ хранения	Централизованный	Смешанный или распределённый
Модель хранения и обработки данных	Преимущественно вертикальная	Горизонтальная модель

Термин Big data (большие данные) не следует путать с терминами Data science (наука о данных), Machine learning (машинное обучение). Последние два относятся к глубокому изучению данных, поиску корреляций и выстраиванию машинных моделей обработки данных. Первый – непосредственно к большим данным, к подходам и инструментам по их хранению, обработке и анализу.

Для решения задач обработки многообразия данных значительного объёма были выработаны новые специализированные подходы и продукты, например:

- NoSQL (с англ. «Not only SQL» - не только SQL) БД.
- MongoDB – документно-ориентированная БД, использует подход к построению нетрадиционных, не обязательно реляционных и строго структурированных БД.
- Параллельная обработка данных с возможностью горизонтального масштабирования. Hadoop – ПО с открытым кодом компании Apache Software Foundation, созданное в парадигме MapReduce, когда исполняемые задачи разбиваются на множество элементарных задач, исполняющиеся на сети распределённых ЭВМ, кластеров, после чего результат исполнения сводится воедино.
- Clickhouse – свободно-распространяемое, горизонтально масштабируемое ПО компании Yandex.

Методы работы с данными

При работе с данными обычно выделяют методы сбора и методы анализа данных. Данное учебное пособие посвящено рассмотрению обоих видов. Эмпирическое исследование, в противоположность теоретическому, представляет собой совокупность последовательных методологических, методических и организационных процедур, направленных на получение достоверных данных об изучаемом явлении для их последующего практического применения.

Первой классификацией методов исследования можно назвать их деление на теоретические и эмпирические. К теоретическим методам исследования относятся сравнительно-исторический, структурно-функциональный, герменевтический и системный. К эмпирическим методам исследования относят методы наблюдения, анализа документов, методы анкетирования и интервьюирования. Эмпирические методы, в свою очередь, подлежат классификации. Иногда их делят на опросные и неопросные методы. При этом к опросным методам относятся метод анкетирования, метод интервьюирования, метод фокус-групп, а к неопросным методам – методы анализа документов, наблюдение. Данная классификация относится к традиционным методам сбора данных с их последующей обработкой статистическими методами. Также эмпирические методы исследования могут быть количественными и качественными. К количественным методам в социологии относятся метод контент-анализа документов, парсинг, метод опроса и прямого наблюдения. К качественным методам сбора информации относят глубинные интервью, фокус-группы и кейс-стади.

Использование методов наблюдения позволяет получить факты (свидетельства) о состоянии объективной реальности. Метод наблюдения является одним из самых эффективных методов сбора информации, поскольку предполагает вовлечение ученого в сбор информации. В свою очередь, методы наблюдения можно подразделить на формализованные и неформализованные, включенные и невключенные виды наблюдения.

Необходимо отметить, что наблюдение является методом сбора информации посредством непосредственной регистрации исследователем событий и условий в поле. В отличие от обыденного наблюдения научное наблюдение является целенаправленным восприятием для достижения научных целей и задач. Объективность в плане возможности контроля путем повторного наблюдения или с использованием других методов исследования является отличительной чертой научного наблюдения.

Научные наблюдения бывают следующих видов: формализованное, и неформализованное, включенное и невключенное.

Формализованное наблюдение структурируется жесткой программой и предполагает разработку и использование инструментария (карточек и протоколов). Неформализованное наблюдение может ограничиться общим планом. Необходимо отметить, что неформализованное наблюдение не позволяет собрать данные о тенденциях и закономерностях изучаемых процессов. Это всегда основа для проведения более тщательного формализованного наблюдения или эксперимента.

Включенность наблюдателя в процесс наблюдения предполагает выделение двух типов наблюдения: включенного и невключенного. При проведении невключенного наблюдения исследователь отмечает факты пассивно, наблюдая за явлением со стороны. Включенное наблюдение предполагает погружение в

ситуацию и часто называется наблюдением «в маске», так как исследователь скрывает не только сам факт проведения исследования, но и свою роль.

Основным источником информации для исследователя служит текст, который размещается как на электронных носителях, так и на бумаге, и в интернете. Документ является материальным носителем, содержащим информацию в установленной форме и по установленным правилам, необходимую для научно-исследовательских и практических целей.

Для логического метода анализа документов характерна некоторая субъективность восприятия данных. Для того, чтобы сделать качественный анализ объективным, необходимо использовать логические приемы (индукция, дедукция, классификация, систематизация, анализ, синтез, сравнительный анализ, аналогия).

Этапы анализа документа:

1. Выбор документа в соответствии с целями и задачами исследования.
2. Анализ контекста создания документа.
3. Логический анализ документа

При интуитивном анализе документов используются следующие логические процедуры анализ документов:

- индукция – метод рассуждения от частного к общему;
- дедукция – метод рассуждения от общего к частному;
- классификация – дифференциация идей текста по релевантному признаку;
- систематизация – объединение идей текста вокруг ключевой идеи по определенному принципу;
- анализ (разложение, расчленение, разборка) — метод исследования, характеризующийся выделением и изучением отдельных частей объектов исследования;
- синтез – процесс соединения текстовых данных в единое целое и обычно используется вместе с анализом.

Контент-анализ является количественным методом анализа документов, осуществляющем перевод текстовых данных в количественные показатели с последующей их статистической обработкой и интерпретацией. Сущность контент-анализа состоит в подсчете интересующих нас ключевых слов с выяснением скрытого смысла текста.

Тексты в контент-анализе — это книги, статьи, рекламные тексты, речи выступлений, тексты на веб-страницах.

Необходимо отметить, что контент-анализ используется только как дополнительный метод при проведении значительного исследования, которое базируется на использовании программы эмпирического исследования.

Контент-анализ предполагает, что исследователь определяет объект, категории и единицы анализа, выбирает статистический метод и начинает сбор данных. Контент-анализ позволяет увидеть в документе его скрытый смысл.

Этапы контент-анализа:

1. выделяют единицы анализа и переводят их в числовой формат;
2. проводят подсчет частотных распределений методами статистического анализа;
3. исследователь проводит интерпретацию полученных результатов.

Объект контент-анализа — это тексты, публикуемые в книгах, газетах, на веб-страницах, также это ответы на открытые вопросы анкет. Единицы анализа — это темы, образы, метафоры, примеры, проблемы, аналогии, каламбуры, надписи на стенах, фамилии политиков, названия партий и т.д. Выбор единицы анализа зависит от исследовательской программы, объекта, предмета исследования и цели. Главная задача контент-анализа заключается в установлении единиц анализа. При этом единицы анализа должны подчиняться требованиям единообразия, т. е. легко и однозначно идентифицироваться в тексте. Слова, выбранные для подсчета, должны быть интересны для последующей политологической, социологической, социальной интерпретации.

Метод анкетирования является еще одним распространенным количественным методом сбора данных. При составлении анкеты действуют не только логические, но психологические и эстетические законы. Составление анкеты — это искусство, которое опирается на определенные правила, но полностью не подчинено им. Стержень анкеты — это композиция расположения вопросов. Композиция — это «составление», «связывание». Структура — это совокупность устойчивых связей объекта, обеспечивающих его целостность и тождественность самому себе.

Анкета уникальна, что объясняется рядом причин:

- 1) создается под конкретное исследование;
- 2) отражает индивидуальное мировоззрение исследователя;
- 3) адресована для изучения именно этого объекта исследования;
- 4) при составлении новой анкеты возможна преемственность отдельных вопросов, блоков вопросов, но не анкеты в целом.

Структура анкеты состоит из следующих частей:

- введение (цель исследования, правила заполнения анкеты, личная выгода для респондента от участия в опросе, обещание анонимности или конфиденциальности);
- реквизитная часть (название анкеты, дата, время и место проведения опроса, фамилия интервьюера);
- информативная часть (содержательные вопросы с нарастанием степени сложности);
- классификационная часть (паспортичка);
- заключительная часть (благодарность за участие).

Части 1, 2, 5 — обслуживающий, вспомогательный аппарат анкеты, части 3 и 4 — основная, базисная часть анкеты. «Эффект эха» означает, что вопросы расположены в анкете так, что ответ на один вопрос предполагает схожие ответы на другие вопросы. При составлении анкеты нужно соблюдать следующие требования: термины должны быть понятны респонденту; базисная часть (информативная часть) должна содержать 20-30 вопросов; формулировка вопросов должна соответствовать исследовательской задаче; анкета должна соответствовать способностям респондента, не унижать его достоинства.

Программная логика вопроса не должна совпадать с логикой построения анкеты. Например, в соответствии с программной логикой выделяем и опрашиваем заинтересованных лиц, затем незаинтересованных лиц в данном товаре. В логике построения анкеты должно быть сначала так: опрашиваем всех, затем только участников конфликта, затем незаинтересованных лиц, а в заключение всех респондентов. Для получения достоверной информации и избегания «эффекта эха» сначала задаются вопросы о частных сторонах явления, а затем обобщающий вопрос. Вопросы должны идти от простых к сложному. Сначала задается общий вопрос для завязки диалога. Уточняется мнение респондента по каким-то вопросам. Далее идут более сложные вопросы, относящиеся к событиям, фактам. 2-3 самых сложных вопроса, требующих размышления и работы памяти, задаются «на пике». Затем все заканчивается «паспортичкой» (социально-демографической информацией). Вопросы должны идти от общего к конкретному (туннельный подход). Есть еще секционный подход (анкеты-омнибусы) - переключение между разными темами путем специальных вопросов.

Источниками информации для разработки анкеты служат чужие анкеты, литература, интуиция, предыдущий опыт и СМИ. Переменная исследования: объект, предмет (процесс, явление), интересующие исследователя. Переменная анкеты: вопрос анкеты (инструмент операционализации). Анкетный вопрос представляет совокупность знаний, включающую то, что нам известно, и то, что мы хотим узнать. Выделяют три функции анкетных вопросов:

- индикаторная функция – направлена на получение искомой информации;
- коммуникативная функция – означает связь между исследователем и респондентом;
- инструментальная функция – характеризует измерительные возможности вопроса.

В данном учебном пособии пойдет речь о сборе текстовых данных посредством парсинга, работе с большими данными, а также обработки данных с использованием языка программирования R и сетевого анализа посредством RajeK.

Уровни и виды социальных исследований

Когда мы планируем исследование, мы должны находиться на позициях определенной научной парадигмы и выбрать в ее рамках соответствующую

концепцию. Парадигма включает набор подходов и методов изучения реальности, которые определяют идеи о предмете исследования, а также устанавливают особые правила взаимодействия ученых друг с другом в рамках этих идей. Понятие парадигмы, как специфической формы научной работы было предложено впервые Т. Куном в работе «Структура научных революций» [7, С.120-132]. Необходимо отметить, что научное сообщество делится на группы, которые работают в рамках определенной парадигмы. Мы можем говорить о научных парадигмах в рамках определенных наук. Так, в рамках социологии можно выделить позитивизм, структурализм, структурный функционализм и социальный конструктивизм. В политической науке ключевыми парадигмами является институционализм, новый институционализм, структурный функционализм и аксиологический подход. Выбор той или иной парадигмы задает ракурс анализа и интерпретации данных, получаемых в ходе исследования. Однако и на уровне парадигмы существуют разные теории, и для того, чтобы корректно провести исследование и интерпретировать результаты, необходимо корректно определить не только парадигму, но и теорию. Дело в том, что каждая парадигма состоит из набора теорий и концепций. Например, неоинституциональная парадигма в политологии представлена теорией общественного выбора, теорией игр и сетевым анализом. Таким образом, когда мы готовимся проводить исследование, нам необходимо определить не только парадигму, но также определить теорию или теоретическую рамку для концептуализации проблемы исследования и ее дальнейшей интерпретации.

Научное познание может осуществляться как на уровне логики с последующим подтверждением гипотезы эмпирическим материалом (дедуктивное обоснование), так и на эмпирическом уровне, когда мы начинаем со сбора эмпирических фактов и заканчиваем значимым выводом. Примером такой индуктивной теории является «обоснованная теория» Б. Глейзера и А. Страуса [8, С. 143-153]. Деление теорий на микро-, мезо- и макроуровни можно продемонстрировать на примере лестницы абстракций Дж. Сартори [9, С. 66-77]. Суть данной концепции заключается в том, что, находясь на нижнем уровне этой лестницы, мы получаем знание, богатое деталями, но нам сложно сделать обобщение таких предметов на нижнем уровне абстракции, на среднем уровне мы можем уже распределить эти предметы по классам, на верхнем уровне у нас получаются генерализации, когда мы уже можем обобщить классы одним понятием. Движение по лестнице абстракций происходит с уменьшением детализации от нижнего к верхнему уровню знания. Но важно помнить, что не все знание мы получаем из эмпирики, есть знание, которое получается на верхнем уровне из логических рассуждений. Очень важно, чтобы при движении наверх контекстуальность знания уменьшалась, иначе мы можем получить обобщение детализированных описаний, что приведет к путанице.

Ученые, проводящие исследования, в рамках социальных наук могут быть классифицированы на две группы. Первые занимаются приращением знания ради знания, и мы можем назвать их фундаментальными учеными. Вторая группа состоит из ученых-прикладников, аналитиков, в чью задачу входит выявление проблемных сфер, их изучение с последующей формулировкой вариантов решения.

В данном учебном пособии мы будем говорить как об особенностях проведения социального исследования, так и об особенностях его разновидности – анализа публичной политики [10, С. 255-277]. Но необходимо отметить, что социальное исследование представляет собой объяснение социальных фактов через политические, психологические и экономические факты. Если строгое социологическое исследование зиждется на требовании свободы от оценочных суждений с точки зрения «плохо» или «хорошо», то некоторые виды социальных исследований в рамках политологии предполагают необходимость оценочного суждения.

Системный подход в политологии представляет совокупность методологических процедур, предложенных американским политологом В. Данном [11, Р. 1-22]. В качестве таких методологических процедур выделяют мониторинг с совокупностью методов, используемых для проведения обычного социального исследования (опрос, наблюдение, кейс-стади, статистический анализ). Затем оценивание политики и политических программ, которое позволяет нам сделать вывод о степени достижения проводимой политикой намеченных в программе целей. Далее это прогнозирование, которое дает возможность сформировать представление об ожидаемых результатах политики. И, наконец, это рекомендации, которые позволяют нам сделать вывод касательно предпочитаемых вариантов проведения политики [11, Р. 6-10]. Необходимо отметить, что мониторинг и оценивание политики относятся к ретроспективному анализу (*ex post*). Они проводятся социологами и политологами и касаются результатов реализации политических программ, поэтому данные процедуры не позволяют исправить то, что уже создано. Прогнозирование и рекомендации относятся к перспективному анализу (*ex ante*). Его проводят экономисты и специалисты в области теории принятия решений. Данный вид анализа политики проводится для того, чтобы предугадать варианты ее реализации и по возможности избежать ошибок при решении той или иной политической проблемы.

Мониторинг является центральной методологической процедурой в анализе публичной политики. Существуют четыре методологических подхода при анализе публичной политики: анализ социальных систем, социальный эксперимент, социальный аудит, синтез исследования и практики. Все они требуют разного типа контроля и информации [11, Р. 276-302].

Таблица 2. Основные ограничения четырех методологических подходов в мониторинге политики (Источник: W.Dunn, 2004. p. 284)

Подход	Тип контроля	Тип информации
Анализ социальных систем	Количественный	Доступная или новая
Социальный эксперимент	Прямые манипуляции и количественный тип	Новая
Социальный аудит	Количественный и / или качественный	Новая
Синтез исследования и практики	Количественный и / или качественный	Доступная

Анализ социальных систем основан на использовании социальных показателей. Социальные показатели представляют собой статистику, которая позволяет измерять социальные условия и их изменения за время реализации политической программы. Социальные показатели являются выражением смысла переменных, которые характеризуют индивида, событие, объект, имея разные числовые значения.

Существует два способа определения переменных: конструктивное определение и операционное определение. Конструктивное определение представляет собой словесное определение из словаря с использованием синонимов. Например, мы можем определить «образовательные возможности», как свободу в определении образовательной среды в соответствии с чьими-либо способностями. Конструктивное определение переменной указывает на приблизительное отношение данной переменной к реальности.

Однако мы можем понять политические действия и их результаты только косвенно, используя операционные определения и показатели переменных. Операционные определения придают значение переменной посредством уточнения операций, необходимых для ее измерения. Например, с точки зрения операционного определения мы можем определить «образовательные возможности» как количество детей из семей с доходом менее трехсот тысяч рублей в год, посещающих колледжи и университеты согласно отчетам.

При этом показатели переменных – это непосредственно наблюдаемые характеристики, которые заменяют косвенно обозреваемые или не обозреваемые характеристики переменных. Например, количество подростков, ежегодно поступающих в университет на бюджетные места, количество наркозависимых или количество выбросов серы в воздух. Количество делает восприятие переменной обозреваемой, в то время как качество мы не можем измерить (удовлетворенность работой, качеством жизни, экономическим прогрессом). Необходимо отметить, что для определения одной и той же переменной мы можем использовать разные показатели, что порождает проблему их интерпретации.

Поскольку отношение между переменными и показателями является неоднозначными, желательно использовать множественные показатели (индексы) для характеристики действий и результатов политики. Индекс представляет собой

комбинацию двух и более индикаторов, которые позволяют лучше измерить продукты реализации политики. Например, существует индекс загрязнения воздуха, индекс потребления электроэнергии, индекс качества жизни, индекс покупательной способности, индекс потребительских цен. Индекс позволяет отслеживать изменения в результатах реализации политики во времени, показывая, насколько изменились их показатели по отношению к базовому периоду.

Индексы могут быть простыми и составными. Простые индексы состоят только из одного показателя. Например, количество преступлений на сто тысяч жителей региона. Составные индексы включают несколько показателей. Например, индекс потребительских цен строится на основе показателей стоимости четырехсот товаров и услуг в США. Индекс конструируется либо посредством агрегирования, либо посредством усреднения. Агрегированные индексы строятся путем суммирования значений показателей (потребительских цен) за определенный период. Усреднение требует вычисления изменения средних значений показателей за определенный период времени. Например, индекс потребительской способности является агрегированным индексом. Он измеряет реальный заработок в соответствующие периоды, исходя из уровня индекса потребительских цен.

Использование индексов для характеристики изучаемых переменных часто связано с проблемой неточности информации, зафиксированной в этих показателях, сложностью формирования выборки для расчета индексов, которая бы однозначно отражала интересы всех слоев общества. Также индексы не всегда характеризуют качественные изменения в их значениях.

Таким образом, анализ социальных систем представляет собой статистический анализ показателей и индексов, характеризующих макросостояние социальных общностей на уровне государства, региона или муниципалитета. Социальные показатели позволяют оценить воздействие осуществляемой политики на целевые группы. Однако этот метод часто называется случайной инновацией, так как мы можем оценить качество реализации политики только постфактум, что является не лучшим способом избежать дорогостоящих ошибок, заложенных в политической программе.

Таким образом, как видно из приведенного материала, анализ публичной политики является более комплексным, чем простое социальное исследование. Системный подход предполагает выделение таких методологических процедур, как мониторинг, оценивание политики, прогнозирование и рекомендации. В свою очередь, мониторинг ближе всего находится к традиционному социальному исследованию, включая такие методологические подходы, как анализ социальных систем, социальных эксперимент, социальный аудит и синтез исследований и практики.

Проведение исследования предполагает выполнение определенной последовательности шагов, ведущих от общей идеи исследования (исследовательского вопроса) к эффективно сконструированным переменным, которые можно измерить на практике. Измерение означает аккуратное наблюдение реального мира с целью описания объектов и событий в терминах атрибутов (характеристик), которые составляют переменные. Необходимо отметить, что большинство переменных не существуют в реальности. Они создаются исследователями, поэтому они редко имеют однозначное, непротиворечивое значение. Например, переменная «религиозность» может быть измерена количеством посещения церкви в неделю или принадлежностью к соответствующей конфессии.

Существуют следующие критерии для измерения качества переменных: точность, надежность и валидность. Надежность означает, что при использовании данного метода по отношению к тому же объекту мы получим похожие результаты исследования. Валидность означает степень, до которой эмпирическое измерение переменной адекватно отражает реальное значение изучаемого концепта. Переменные могут быть номинальными, т.е. обозначаемые словами, порядковыми с возможностью ранжирования на большее или меньшее, метрическими, т.е. предполагающими изменение переменных на равных интервалах.

Разработка программы эмпирического исследования

Проведение научного исследования предполагает не только определение проблемы и постановку исследовательского вопроса, но и разработку полноценной программы исследования. Однако до разработки данного документа необходимо прежде всего четко определиться с проблемой исследования, определить теоретическую рамку и создать набор переменных, которые будут измеряться на практике или оцениваться на примере других теоретических источников, например, как при использовании кейс-стади [8, С.132-172].

Определение теоретической рамки тесно связано с выбором научной парадигмы. В социальном исследовании существуют несколько наиболее распространенных парадигм, которые определяют основные подходы к структурированию и интерпретации проблемы, а также правила работы ученых в данной парадигме. Мы поговорим в рамках нашего учебного пособия о позитивизме, конфликтной парадигме (марксизме), структурном функционализме и социальном конструктивизме.

Представителем раннего позитивизма является О. Конт, который первым заявил и обосновал, что общество нужно изучать научными методами. До Конта именно религиозные парадигмы объясняли различия между обществами. Конт первым пришел к выводу, что общественное устройство должно изучаться логически и рационально. Таким образом, позитивизм представляет собой научное направление в методологии, которое полагает в качестве основного источника

знания об обществе те знания, которые получены на основе эмпирического исследования, а не путем философских рассуждений.

Марксизм («конфликтная парадигма» в западной научной традиции – прим. автора О.А. Игнатъевой) рассматривает социальное поведение как конфликтный процесс, заключающийся в стремлении одних доминировать над другими. Формации сменяют друг друга, но конфликт между основными классами каждой формации сохраняется. Процесс общественного развития объясняется на основе действия трех законов диалектики: переход количества в качество, единства и борьбы противоположностей, а также отрицания отрицания.

Парадигма структурного функционализма берет свое начало в работах Г. Спенсера, Б. Малиновского и А. Редклиффа-Брауна, однако наибольшего расцвета она достигает в работах Т. Парсонса и его ученика Р. Мертона. Данная парадигма представляет макросоциальный подход к анализу социальных и политических проблем. Общество в нем представляет собой систему, состоящую из политической, экономической, социальной и культурной подсистем. Каждая подсистема выполняет определенную функцию, необходимую для слаженного функционирования данной системы. В свою очередь, выявленные подсистемы также состоят из институтов, выполняющих определенную роль, позволяющую сохранять устойчивость системы. Для Т. Парсонса конфликт является чуждым явлением для общества и является признаком его болезни. Однако Р. Мертон, продолживший разработку данной парадигмы, выявляет возможность возникновения дисфункций социальной системы, таких как бюрократизм и коррупция.

Парадигма социального конструктивизма является микросоциальным подходом к анализу социальных явлений. Ведущими представителями данной парадигмы являются П. Бергер и Т. Лукман. Суть данной парадигмы заключается в том, что социальные отношения начинают формироваться на уровне взаимодействия индивидов. Эти самые первичные взаимодействия, например, при формировании новой семьи становятся привычными практиками, которые для второго и третьего поколения передаются как традиция и не подвергаются сомнению. Таким образом, устойчивые социальные институты становятся таковыми на основе первичного взаимодействия индивидов, т.е. взаимодействие является источником возникновения институтов.

Далее нам необходимо определиться с проблемой и провести концептуализацию и операционализацию выбранных для ее описания научных понятий. Концептуализация понятия связывает дизайн исследования или результаты исследования с теорией, выбранной для обоснования данного исследования. Операционализация выбранного понятия предполагает создание системы переменных и индикаторов для описания и измерения концепта.

Традиционная модель науки предполагает прохождение следующих этапов при проведении исследования [12, С. 72-117]. Так, исследователь сначала должен

определился с теоретической рамкой. При этом возможно использовать несколько теоретических рамок в зависимости от количества переменных, которые будут использованы при построении модели исследования. На основании теории исследователь формулирует гипотезу, которая устанавливает каузальные отношения между двумя переменными и более. Например, гипотезу можно сформулировать следующим образом, студенты, изучающие экономику, склонны уделять больше времени математике, чем студенты, изучающие политологию. Это утверждение нужно проверить. Для того, чтобы протестировать гипотезу, нам необходимо замерить все переменные, которые в нее входят. Заключительным этапом в традиционной модели науки служит наблюдение того, как ведут себя индикаторы переменных в реальности. Мы не просто тестируем гипотезу путем ее верификации, но и должны ее фальсифицировать [13, С. 78].

Исследовательский дизайн (программа исследования) позволяет нам упорядочить наши усилия в ходе проведения научного исследования. С точки зрения Добренькова В.И. и Кравченко А.И. [14, С. 151], именно разработка программы исследования, а также анализ данных и их интерпретация занимают наибольшее количество времени у исследователя. В российской традиции программа исследования состоит из двух частей. Первая часть – теоретико-методологическая, вторая часть методическая. Первая часть с необходимостью включает в себя проблему исследования, объект, предмет, цель и задачи исследования. Также в этой части описывается концептуализация и операционализация базовых понятий исследования. Цель исследования означает модель ожидаемого результата (способ решения проблемы), который можно достичь при проведении данного исследования. Проблема исследования – это отражение проблемной ситуации, которая, в свою очередь, содержит противоречие между тем, что есть, и тем, что должно быть, или между знанием и незнанием. Задачи – это этапы на пути достижения цели, поэтому они не могут быть шире или одного объема с целью по содержанию. Объект исследования – это та часть объективной реальности, на которую направлен фокус внимания ученого. Например, объектом исследования могут выступать избирательная кампания 2020 г. в США. Предмет исследования – это одна из сторон (один из аспектов) объекта исследования, например, результаты избирательной кампании 2020 г. в США.

Методическая часть программы исследования включает описание выборки и правил ее формирования, совокупность методов сбора и анализа данных, обоснование логики их использования. Необходимо отметить, что в качественных и количественных исследованиях выборки формируются по-разному, о чем речь пойдет далее. В качестве методов сбора информации можно указать количественные методы сбора информации (опрос, наблюдение, социометрический опрос) и качественные методы (кейс-стади, интервью, фокус-группы) [15, С. 48-55]. К методам обработки информации можно отнести

статистические методы анализа, метод конденсации смыслов [16, С. 224-225] метод обоснованной теории и т.д. [17, С. 29-38].

Необходимо также отметить, что программа исследования должна сопровождаться графиком выполнения работ, так как задержка в выполнении того или иного этапа проведения исследования чревата несвоевременным выполнением работы. Сам процесс исследования предполагает реализацию следующих этапов: определение проблемы исследования, создание программы (дизайна) исследования, сбор данных (полевая стадия), анализ данных и написание отчета, научно-исследовательской работы или научной статьи.

Основные ошибки при сборе данных

Познавая мир, мы ориентируемся, с одной стороны, на наш опыт, а с другой стороны, на знание о нем других, которые передаются из поколения в поколение. Знание, которое мы получаем от других, преобладает над тем, которое мы можем получить из собственного опыта, так как мы не можем самостоятельно охватить все многообразие окружающего мира самостоятельно. Следовательно, мы опираемся на информацию, которая содержится в традиции семьи и социума, а также на экспертные знания. Часто таким экспертом может выступать медийное лицо, известный журналист, актер, политик. Но на сколько мы можем доверять знанию такого лица, например, в области самолетостроения? Соответственно мы можем говорить о повседневном знании, которое не отвечает строгим критериям научности, и, собственно, о научном знании, которое обладает системностью, объективностью и точностью.

В философии существует отрасль под названием эпистемология, которая занимается теорией познания, устанавливает его принципы. В свою очередь, методология представляет собой подраздел эпистемологии, направленный на поиск способов получения истинного, научного знания, разработку процедур научного исследования.

Какие ошибки в ходе проведения научного исследования уводят нас от получения научно-достоверного знания? Американский социолог Е. Вабби выделяет четыре обычные ошибки при сборе данных – это неаккуратное наблюдение, чрезмерное обобщение, выборочное наблюдение и нелогичное обоснование [2, Р. 6-8].

С неаккуратным наблюдением мы сталкиваемся тогда, когда пытаемся вспомнить о каком-то событии пост-фактум, когда мы не используем программу наблюдения с указанием объектов наблюдения, единиц наблюдения, временных промежутков проведения этого наблюдения, т.е. тогда, когда вместо научного наблюдения мы ограничиваемся повседневным. Например, при анализе отношения к вашему кандидату, которого вы продвигали в рамках политической кампании, вы пытаетесь вспомнить, что говорили представители гражданского

населения на встрече с ним. Вы не сможете сделать это точно, так как на момент встречи у вас не было четко поставленной цели.

Следующая ошибка – это чрезмерное обобщение. Данная ошибка имеет место в наших исследованиях тогда, когда мы анализируем небольшую совокупность единиц наблюдения и затем пытаемся распространить полученные знания на всю генеральную совокупность или на весь объект наблюдения. Такая ошибка часто встречается тогда, когда мы проводим кейс-стади. Например, перед нами стоит задача проанализировать процесс приватизации на примере приватизации одного из вагоноремонтных заводов в Великобритании в эпоху М. Тэтчер. После того как мы проанализировали этот кейс, мы пытаемся сделать значимые выводы обо всем процессе приватизации и не только в Британии, но и в других странах. Это и есть ошибка чрезмерного обобщения.

Также мы можем столкнуться с ошибкой селективного (выборочного) наблюдения. Например, когда-то мы проанализировали большой массив данных и пришли к выводу, что в США наибольшее количество преступлений совершается людьми из низкого страта. Если последующее исследование покажет, что убийства совершаются также и представителями средних классов, мы будем просто игнорировать эту информацию для того, чтобы она не помешала нашей прекрасно разработанной и подтвержденной в рамках одного исследования гипотезы.

И, наконец, речь пойдет об ошибке нелогичного обоснования. Она встречается, например, в широко известной поговорке «Исключение подтверждает правило». Однако данное утверждение само по себе нелогично, поскольку это лишь умозаключение не имеет под собой эмпирического обоснования. Еще одним примером нелогичного обоснования является «ошибка игрока». Так, например, игрок в покер предпочитает оставаться в игре до последнего в надежде, что за неудачей последует удача.

Таким образом, нам необходимо очень осторожно следить за возможностью появления таких ошибок в ходе проведения исследования, так как они могут стать причиной получения недостоверного знания, в результате которого наша статья или более значимая работа может быть отклонена эпистемическим сообществом.

Практическое задание

Составить программу эмпирического исследования, определив научную парадигму и используя процедуры концептуализации и операционализации.

1.2 Статистика с использованием языка программирования R

Введение в программирование на языке R

Использование статистических методов стало проще для ученых и студентов с появлением статистического пакета SPSS [18]. Однако, несмотря на дружелюбный интерфейс данного программного обеспечения, оно имеет один существенный минус, который ограничивает возможности его широкомасштабного использования – это стоимость SPSS. Только крупные учебные заведения и исследовательские организации могут позволить себе установить лицензионную версию данного пакета.

Язык программирования R позволяет устранить данный существенный недостаток SPSS, так его использование не предполагает значительных издержек для исследователя. Язык программирования R представляет собой целую вселенную, позволяющую обрабатывать количественные данные как в статистике, так и в сетевом анализе и парсинге. Однако, в отличие от SPSS, для того, чтобы провести статистический анализ данных, недостаточно просто нажать на «кнопку», и машина все сама посчитает. Нет, здесь необходимо владеть знаниями функций на языке R для осуществления статистических расчетов. Язык программирования необходимо учить так же, как и иностранный язык, посредством повседневной практики и освоения все новых и новых разделов статистики. Необходимо отметить, что вокруг языка программирования R сложились сообщества его разработчиков и пользователей, которые позволяют решать возникающие проблемы при использовании тех или иных функций для статистического анализа на языке R. Речь идет о сообществах www.habr.com, Community Stack Overflow, RStudio Community [19].

Для начала работы на языке программирования R необходимо установить две программы: R и RStudio. Для этого нужно воспользоваться ссылками www.cran.r-project.org и www.rstudio.com/products/rstudio/download, которые позволяют загрузить лицензионные программные продукты бесплатно. Статистические вычисления производятся в программе RStudio, которая согласовано работает с программой R. При открытии скрипта в RStudio перед нами появляется окно, состоящее из четырех разделов: 1) окно для скрипта; 2) окно консоли; 3) объекты, созданные в среде «История операций»; 4) файлы, рисунки, установочные пакеты и помощь [20] – Рис. 1.

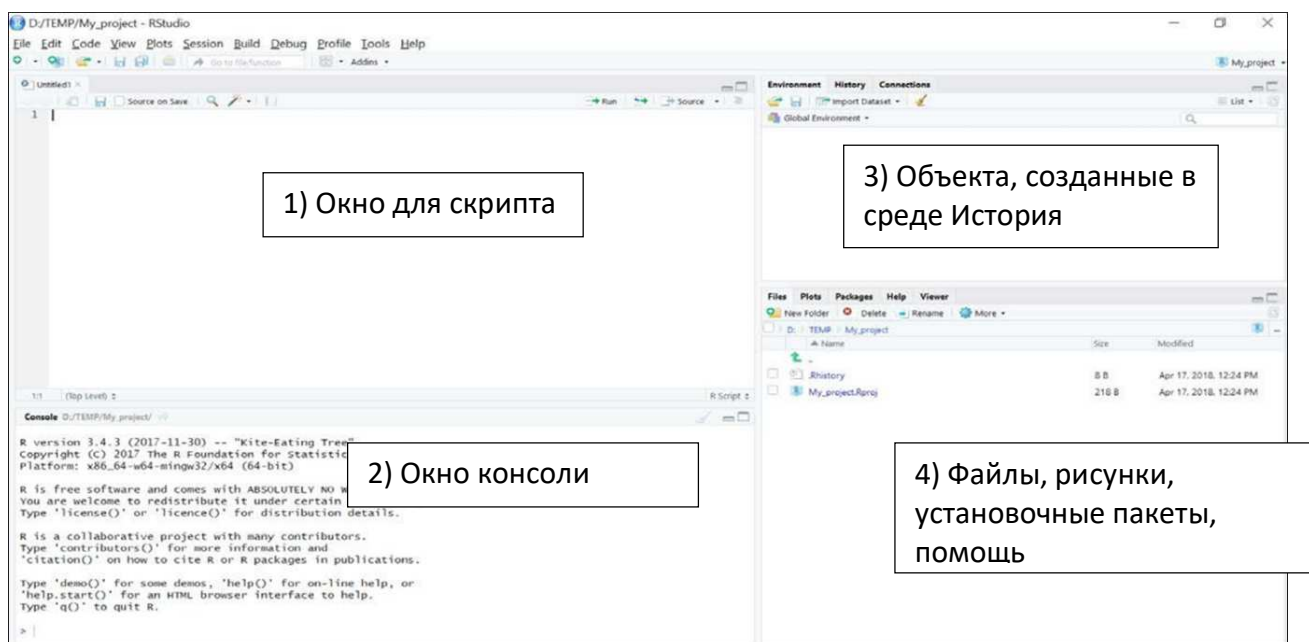


Рисунок 1 Окно для работы в RStudio

Для написания функций, позволяющих проводить статистические расчеты, необходимо использовать окно для скрипта. Результаты вычислений отражаются в окне консоли [21, С. 35-43]. Документы, созданные в RStudio, сохраняются с расширением `.RData`. Однако в RStudio можно также работать и с документами (базами данных), имеющих расширение `csv`, `xls` и `sav`. Для того, чтобы открыть документ в формате `.RData`, достаточно использовать команду `File -> Open File`. Для работы с файлами в другом формате нужно прописать директорию. Важно учесть, что на современных компьютерах есть только системный диск `C`, куда сохранять файлы с базами данных нежелательно. Для этого лучше использовать флеш-карту, которая будет идентифицироваться компьютером как диск `d`. Однако выбор диска для написания директории – это еще полпроблемы. Основной вопрос: где брать базы данных для обучения или проведения самостоятельных научных исследований? Некоторые базы есть в специализированных пакетах R, например, в пакете `MASS` [22]. Также данные можно брать на портале открытых данных, например, на сайте портала открытых данных правительства г. Санкт-Петербург: <http://data.gov.spb.ru/>. На порталах открытых данных базы данных в основном доступны в `csv` и `xls` формате.

Доступ к файлам, указанным в пособии, обеспечивается на информационном портале Центра технологий электронного правительства Института дизайна и урбанистики Университета ИТМО (регистрация обучающихся — <https://news.egov.itmo.ru/for-students>).

Для открытия базы данных в формате `xls` необходимо активировать пакет `readxl`. Для активации пакета мы используем функцию `library()` и укажем в скобках данный пакет. Таким образом, активация пакета выглядит так: `library(readxl)`. После этого нам необходимо запустить данную функцию, используя комбинацию

клавиш CTRL+ENTER. Сохраним на диске d с портала открытых данных правительства г. Санкт-Петербург файл Post offices_SPb.xls. Вложим содержимое директории в переменную post при помощи знака присвоения «<->». Получим переменную post<-read_xls('e:/Post offices_SPb.xls',sheet=1). Проверим результат при помощи функции чтения верхних строк базы данных head(), т.е. head(post, n=10).

```
library(readxl)
post<-read_xls('e:/Post offices_SPb.xls',sheet=1)
head(post, n=10)

## # A tibble: 10 x 8
##   name      district address nearest_subway_~ mode  closed_for_lunch
##   <chr> <chr>      <chr> <chr>          <chr> <chr>          <chr>
## 1 Санк~ Адмирал~ "г. Са~ ст.м.Невский пр~ "кру~ <NA>          088; 314~
## 2 Санк~ Адмирал~ "г. Са~ ст.м.Технологич~ "пон~ 13.00-14.00  251-00-14
## 3 Санк~ Адмирал~ "г. Са~ <NA>          "пон~ 13.00-14.00  714-59-14
## 4 Санк~ Адмирал~ "г. Са~ ст.м.Технологич~ "пон~ 13.00-14.00  316-14-20
## 5 Санк~ Адмирал~ "г. Са~ ст.м. Балтийска~ "пон~ 13.00-14.00  251-21-7~
## 6 Санк~ Адмирал~ "г. Са~ ст.м.Сенная пло~ "пон~ 14.00-15.00  310-72-37
## 7 Санк~ Адмирал~ "г. Са~ ст.м.Сенная пло~ "пон~ 14.00-15.00  314-47-80
## 8 Санк~ Адмирал~ "г. Са~ ст.м.Невский пр~ "пон~ 13.00-14.00  314-93-27
## 9 Санк~ Адмирал~ "г. Са~ ст.м. Балтийская "пон~ 14.00-15.00  251-18-03
## 10 Санк~ Адмирал~ "г. Са~ ст.м.Невский пр~ "пон~ 12.00-13.00  570-34-71
## # ... with 1 more variable: class <chr>
```

Теперь откроем файл в формате csv. Это можно сделать двумя способами. Во-первых, можно использовать функцию read.table(). И тогда данная операция будет выглядеть следующим образом: read.table('e:/Districts_SPb2.csv', sep=',', header=TRUE).

```
read.table('e:/Districts_SPb2.csv',sep=',',header=TRUE)

##   name                okato code_    district
## 1 Центральный        40298      31
## 2 Адмиралтейский    40262      32
## 3 Выборгский         40265      36
## 4 Красносельский    40279      16
## 5 Василеостровский  40263       6
## 6 Фрунзенский        40296      13
## 7 Московский         40284      14
## 8 Петродворцовый    40290      21
## 9 Калининский       40273      10
## 10 Пушкинский        40294      35
```

## 11 Кировский	40276	15
## 12 Курортный	40281	38
## 13 Красногвардейский	40278	11
## 14 Колпинский	40277	37
## 15 Невский	40285	12
## 16 Кронштадтский	40280	50
## 17 Приморский	40270	34
## 18 Петроградский	40288	7

Во-вторых, мы можем открыть данный тип файла, используя функцию `read.csv`. Тогда последовательность операций выглядит следующим образом: `districts<-read.csv('e:/Districts_SPb2.csv', header=TRUE)`. Проверим данные при помощи функции `head(districts, n=3)`.

```
districts<-read.csv('e:/Districts_SPb2.csv',header=TRUE)
head(districts, n=3)
```

##	name	okato	code_district
## 1	Центральный	40298	31
## 2	Адмиралтейский	40262	32
## 3	Выборгский	40265	36

Формат `csv` расшифровывается как «comma separated values», т.е., другими словами, это формат, в котором данные разделены запятыми. Этот тип данных относится к «опрятным данным». «Опрятные данные» означают, что в таких данных нет пропусков значений или объединения ячеек. Обычно они также пишутся в Excel, но их значения разделены запятыми. Такие данные наиболее удобны для чтения при использовании многих языков программирования.

RStudio также может читать SPSS файлы с расширением `.sav` [23]. Для этого нужно активировать пакет R: `foreign`, т.е. мы должны указать `library(foreign)` и нажать комбинацию клавиш CTRL-ENTER. Создаем переменную `birth_rate` для того, чтобы прочитать данные об индексе рождаемости в разных странах, хранящиеся в файле SPSS. Для того, чтобы прочитать данный файл в R, нам потребуется функция `read.spss()`. Запишем: `birth_rate<-read.spss('e:/birth_rate.sav', to.data.frame=TRUE)`. Получаем следующий результат, который можно посмотреть при помощи функции `head()` или `tail()`, позволяющих просматривать первые и последние строки.

```
library(foreign)
birth_rate<-read.spss('d:/birth_rate.sav',to.data.frame=TRUE)
head(birth_rate)
```

##	Country
## 1	UAE

```
## 2 DRC (Congo)
## 3 Czech Republic
## 4 Germany
## 5 Finland
## 6 Egypt
## Maternalmortalityrateper100000livebirths
## 1 6
## 2 693
## 3 4
## 4 6
## 5 3
## 6 33
## Numberofdeathsofnewbornsbeforetheageof28days
## 1 415
## 2 97832
## 3 175
## 4 1604
## 5 104
## 6 31796
## Adolescentbirthratenumbersofbirthsper1000womenaged1519years
## 1 28.0758
## 2 126.2696
## 3 10.4158
## 4 7.1462
## 5 7.0306
## 6 51.8828
```

Для извлечения данных и расчетов из программной среды RStudio мы будем использовать написание презентации на языке RMarkdown с сохранением ее в формате Word. Для этого выполним команду File>New File>RMarkdown, затем выберем из списка документ в формате Word. Написание презентаций на языке RMarkdown предполагает использование чанков, в которых прописывается и активируется код. Для вставки чанка нужно набрать комбинацию клавиш CTRL+ALT+I. Первый чанк должен остаться неизменным. Это настроечный код, который запускает функцию knitr. По умолчанию в нем прописан следующий текст:

```
'''{r setup, include=FALSE}
knitr::opts_chunk$set(echo=TRUE)
'''
```

В остальных чанках прописываются коды, которые позволяют делать статистические расчеты и визуализировать данные. Для того, чтобы содержимое в чанках отображалось и запускался код с соответствующей функцией, необходимо указать после r «echo=TRUE, eval=TRUE». Между чанками прописывается текст,

который отражает основное содержание исследования. После того как написание презентации закончено, на панели выбираем «клубок» Kint. Теперь коды в наших чанках активируются, и мы получим документ в формате Word.

Описательная статистика с использованием R

Обратимся к описательной статистике и рассчитаем медиану и квантили, а также среднее и стандартное отклонение для данных из датасета energy.xls. Для начала активируем пакет readxl при помощи функции library().

```
library(readxl)
energy_data<-read_xls('e:/energy.xls',sheet=1)
```

Далее рассчитаем медиану и квантили для переменной, содержащейся в данном датасете, т.е. для AlternativeNuclearEnergy. Построим график для медианы и квантилей в виде боксплота (ящика с усами), Рис. 2.

```
energy<-energy_data$AlternativeNuclearEnergy
median(energy)

## [1] 15.11033

quantile(energy)

##      0%      25%      50%      75%     100%
## 0.9332368 5.9623467 15.1103274 27.8468201 43.2411504
```

```
library(ggplot2)
data_energy<-data.frame(y=energy)
ggplot(data=data_energy)+geom_boxplot(aes(x='Медиана \ни
квантили',y=y))+labs(x='мощности',y='использование \пэнергии')
```

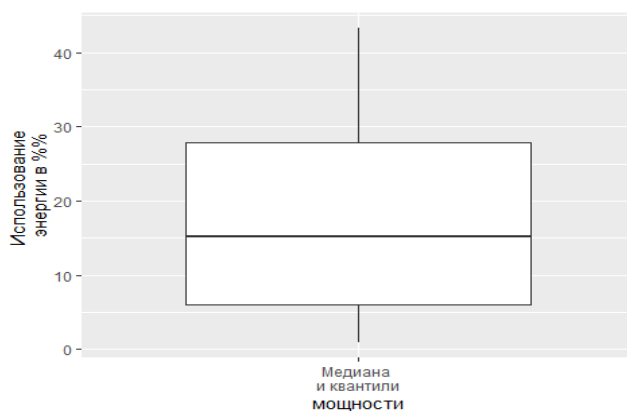


Рисунок 2 Визуализация медианного значения использования источников альтернативной энергии.

Далее рассчитаем медианное значение и значение квантилей для выбросов CO2 с использованием соответствующих функций языка R.

```
emission<-energy_data$CO2Emissions
median(emission)
## [1] 7.088592
quantile(emission)
##    0%      25%      50%      75%     100%
##  3.896324  5.580589  7.088592  9.283754 15.388950
```

Рассчитаем среднее и стандартное отклонение для обеих переменных.

```
mean(energy)
## [1] 17.80113
sd(energy)
## [1] 13.44134
mean(emission)
## [1] 7.971809
sd(emission)
## [1] 3.450858
```

Далее отразим на графике переменную использования альтернативных источников энергии, Рис. 3.

```
library(ggplot2)
data_energy<-data.frame(y=energy)
ggplot(data=data_energy)+geom_boxplot(aes(x='Медиана \ни
квантили',y=y))+stat_summary(geom='pointrange',fun.data=mean_sd1,fun.args=
list(mult=1),aes(x='Среднее\ни ст.отклонение',y=y))+labs(x='Использование
\nэнергии',y='Процент \ниспользуемой энергии')
```

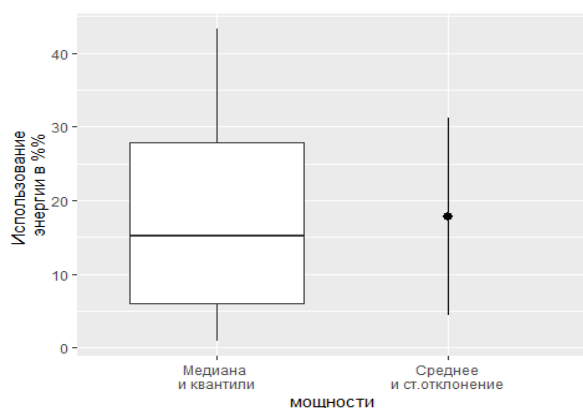


Рисунок 3 Сопоставление среднего и медианного значения количества используемой альтернативной энергии.

Теперь построим среднее и медианное значение для количества выбросов углекислого газа, выраженного в процентах, Рис. 4.

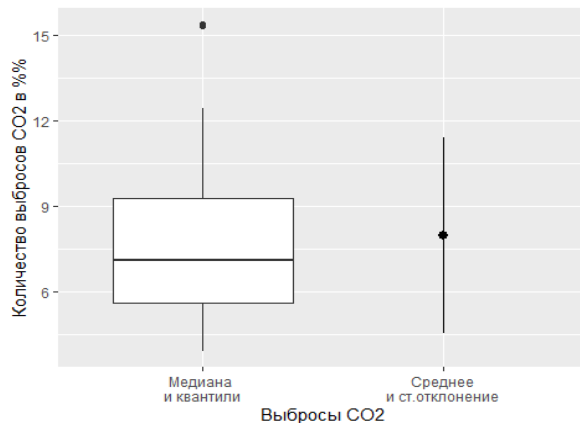


Рисунок 4 Сопоставление среднего и медианного значения количества выбросов CO₂.

Используя данный пример, также рассчитаем корреляцию Пирсона и ранговую корреляцию Спирмена для переменных «Альтернативные источники энергии» и «Выбросы CO₂».

```
cor(energy,emission,method="pearson")  
## [1] -0.4402592  
cor(energy,emission,method="spearman")  
## [1] -0.477193
```

Построение модели множественной регрессии на языке R

В качестве примера использования языка программирования R в статистическом анализе построим множественную линейную регрессионную модель с дискретными и непрерывными предикторами на основании данных о весе новорожденных у курящих и некурящих рожениц [24, С. 49-56]. Данный датасет расположен в пакете MASS, который мы должны активировать при помощи функции `library()`. Конструируемая модель носит ознакомительный характер с кодировками основных статистических функций в языке программирования R. С более сложными регрессионными моделями можно познакомиться в учебнике Дж. Фаравей [25].

```
library(MASS)  
newborn<-birthwt # Переименуем датасет  
#Проверим, из каких переменных состоит датасет  
str(newborn)  
  
## 'data.frame': 189 obs. of 10 variables:  
## $ low : int 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ age : int 19 33 20 21 18 21 22 17 29 26 ...
## $ lwt : int 182 155 105 108 107 124 118 103 123 113 ...
## $ race : int 2 3 1 1 1 3 1 3 1 1 ...
## $ smoke: int 0 0 1 1 1 0 0 0 1 1 ...
## $ ptl : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ht : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ui : int 1 0 0 1 1 0 0 0 0 0 ...
## $ ftv : int 0 3 1 2 0 0 1 1 1 0 ...
## $ bwt : int 2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

Преобразуем значения и тип переменных. Напоминаем, что переменная `Smoke` - дискретная, т.е. она должна стать классифицирующим фактором.

```
newborn$smoke[newborn$smoke==1]<- 'Smoke'
newborn$smoke[newborn$smoke==0]<- 'Dont smoke'
newborn$smoke<-factor(newborn$smoke)
```

Используя точечные диаграммы Кливленда, проверим данные на наличие выбросов. Сначала проверим непрерывную переменную `age` (возраст матери), Рис. 5.

```
library(ggplot2)
theme_set(theme_bw())
gg_point<-ggplot(newborn,aes(y=1:nrow(newborn)))+geom_point()
gg_point+aes(x=age)
```

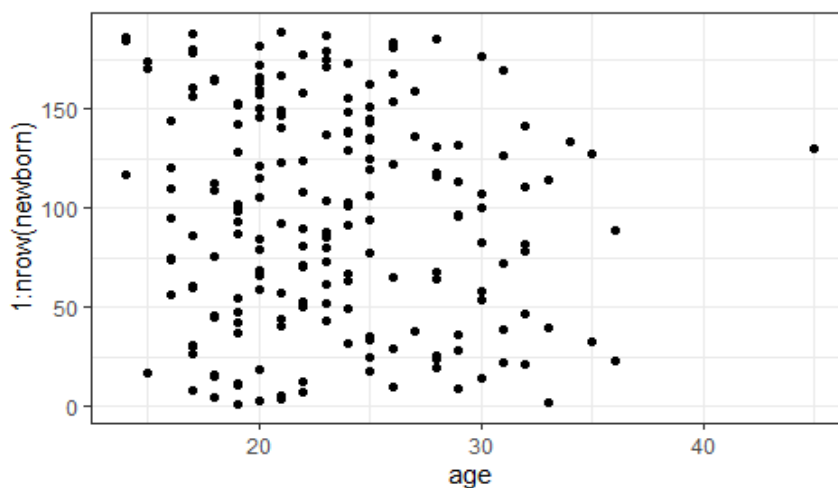


Рисунок 5 Диаграмма Кливленда для независимой переменной `age`

Данные по переменной `age` более-менее однородны.

Теперь исследуем зависимую переменную `bwt` (вес ребенка при рождении), Рис. 6.

```
gg_point+aes(x=bwt)
```

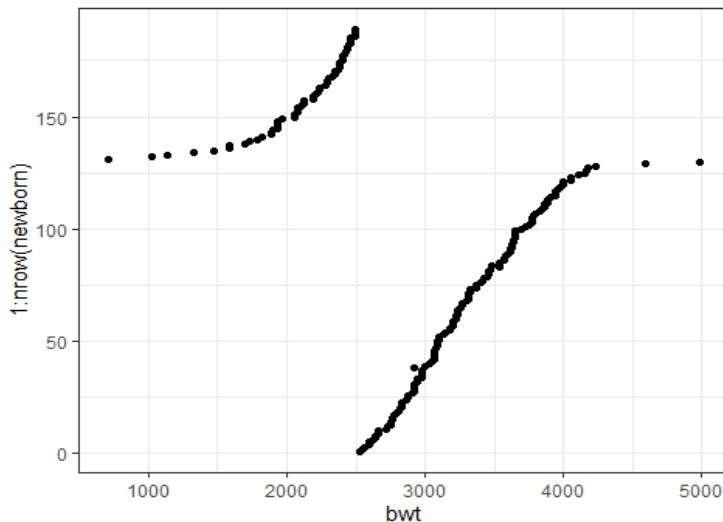


Рисунок 6 Диаграмма Кливленда для зависимой переменной *bwt*

Строим регрессионную модель в дискретными и непрерывными предикторами:

```
bwt=b0 + b1*age+ b2*Smoke + b3*age*Smoke + e
```

Строим модель с взаимодействием предикторов

```
Mod<-lm(bwt~age*smoke, data=newborn)
```

Далее проверим полученную модель на условия применимости модели линейной регрессии, среди которых можно выделить проверку на наличие линейной связи, нормальность распределения, мультиколлинеарность, на отсутствие гетероскедастичности остатков, на независимость наблюдений.

Проверим данные на нормальность распределения, используя квантильный график остатков, Рис. 7. Для этого активируем пакет *car*.

```
library(car)  
qqPlot(Mod, id=FALSE)
```

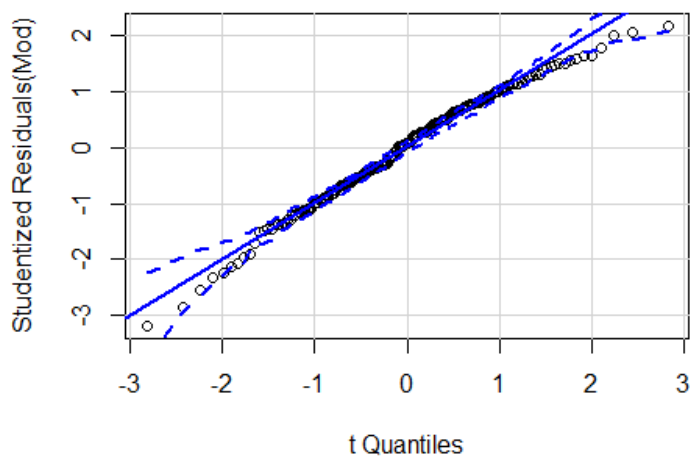


Рисунок 7 Квантильный график остатков

Данные подчинены нормальному распределению, так как значимых отклонений от линии на квантильном графике остатков нет.

Для проверки данных на коллинеарность построим регрессионную модель без взаимодействия предикторов, в которой предикторы будут соединены “+”. Для того, чтобы проверить наличие мультиколлинеарности у предикторов, будем использовать фактор инфляции дисперсии (Variance inflation factor, vif). Функция vif() расположена в пакете car.

```
Mod_vif<-lm(bwt~age+smoke,data=newborn)
library(car)

## Loading required package: carData

vif(Mod_vif)

##      age      smoke
## 1.00197 1.00197
```

Значения vif меньше двух, следовательно, коллинеарности предикторов нет.

Для анализа остатков рассчитаем дополнительный набор данных с использованием функции fortify() из пакета ggplot2. Пакет ggplot2 также используется для построения графиков в среде RStudio.

```
Mod_diag<-fortify(Mod)
head(Mod_diag,n=10)

##      bwt age  smoke      .hat   .sigma   .cooksd  .fitted   .resid
## 85 2523  19      0 0.014443758 710.5267 1.242008e-03 2932.954 -409.954161
```

```

## 86 2551 33      0 0.035590176 708.8245 1.128088e-02 3321.193 -770.193456
## 87 2557 20 Smoker 0.018179969 710.8940 6.854098e-04 2827.422 -270.422121
## 88 2594 21 Smoker 0.015549613 710.9999 3.671583e-04 2808.582 -214.581585
## 89 2600 18 Smoker 0.026666860 710.9027 9.831399e-04 2865.103 -265.103194
## 91 2622 21      0 0.010422673 710.6600 7.101780e-04 2988.417 -366.416917
## 92 2637 22      0 0.009292381 710.6240 6.763800e-04 3016.148 -379.148295
## 93 2637 17      0 0.020812178 710.9529 6.239089e-04 2877.491 -240.491404
## 94 2663 29 Smoker 0.033220950 711.1785 4.671731e-07 2657.857   5.142708
## 95 2665 26 Smoker 0.018528745 711.1691 2.330828e-05 2714.379  -49.378902
##      .stdresid
## 85 -0.582227688
## 86 -1.105776291
## 87 -0.384790670
## 88 -0.304925535
## 89 -0.378863239
## 91 -0.519336577
## 92 -0.537074606
## 93 -0.342661111
## 94  0.007374397
## 95 -0.070275007

```

Проверим данные на наличие влиятельных наблюдений при помощи графика расстояний Кука, Рис. 8. У влиятельных наблюдений расстояние Кука будет больше единицы.

```
ggplot(Mod_diag,aes(x=1:nrow(Mod_diag),y=.cooks)) + geom_bar(stat='identity')
```

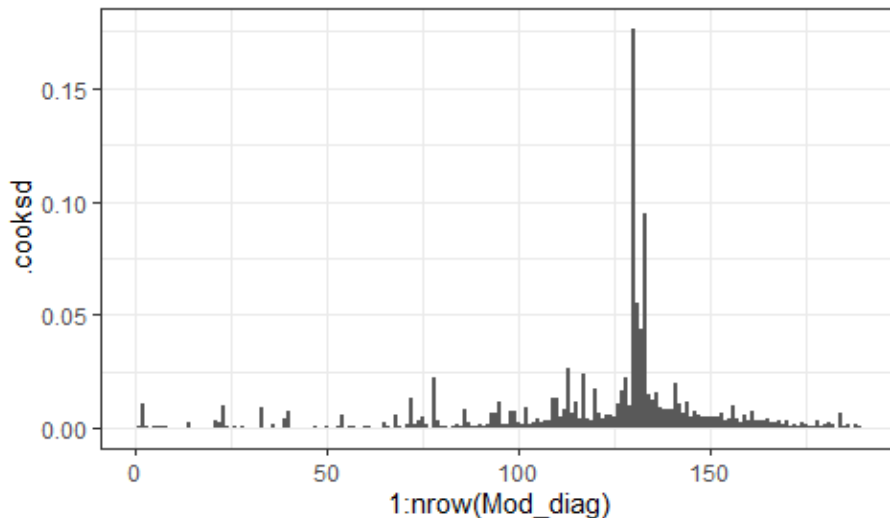


Рисунок 8 График расстояний Кука

Расстояния Кука у всех наблюдений меньше единицы, поэтому в данном датасете влиятельных наблюдений нет.

Построим график остатков в зависимости от предсказанных значений, Рис.9.

```

gg_residue<-
ggplot(data=Mod_diag,aes(x=.fitted,y=.stdresid))+geom_point()+geom_hline(y
intercept=0)+geom_smooth()
gg_residue

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```

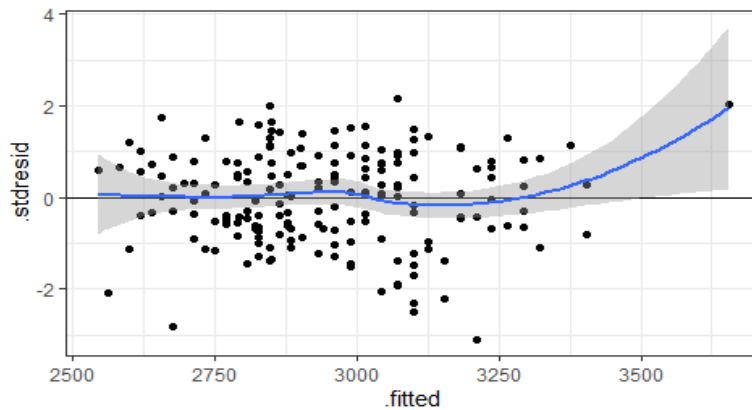


Рисунок 9 График остатков в зависимости от предсказанных значений зависимой переменной

Предсказанные значения распределены равномерно, поэтому гетероскедастичность отсутствует.

Построим графики остатков в зависимости от предикторов, используемых в данной модели, Рис. 10, Рис. 11.

```

gg_residue+aes(x=age)
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```

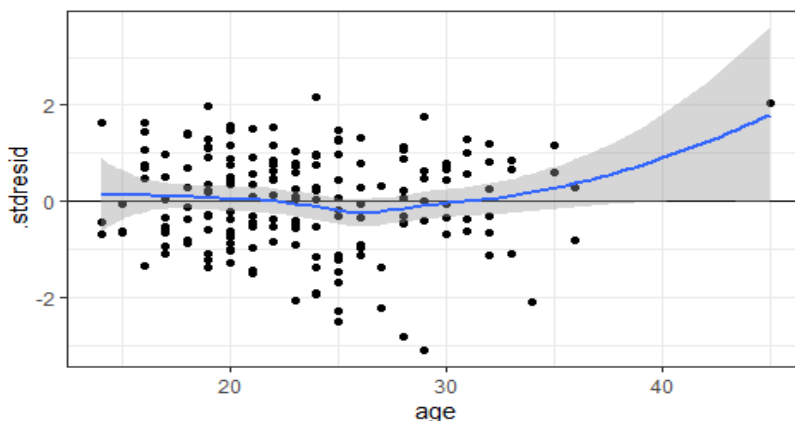


Рисунок 10 График остатков для непрерывного предиктора age

```
ggplot(Mod_diag, aes(x=smoke, y=.stdresid))+geom_boxplot()+geom_hline(yintercept=0)
```

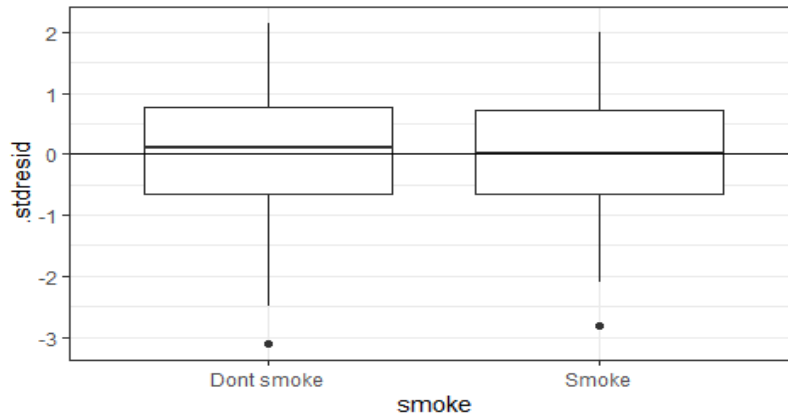


Рисунок 11 График остатков для дискретного предиктора smoke

В обоих случаях гетерогенность дисперсии не выявлена.

При помощи функции `summary()` посчитаем коэффициенты и их значимость, а также коэффициент детерминации.

```
summary(Mod)
```

```
##
## Call:
## lm(formula = bwt ~ age * smoke, data = newborn)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2189.27  -458.46   51.46   527.26  1521.39
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2406.06    292.19   8.235 3.18e-14 ***
## age             27.73     12.15   2.283  0.0236 *
## smokeSmoker    798.17    484.34   1.648  0.1011
## age:smokeSmoker -46.57     20.45  -2.278  0.0239 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 709.3 on 185 degrees of freedom
## Multiple R-squared:  0.06909,    Adjusted R-squared:  0.054
## F-statistic: 4.577 on 3 and 185 DF,  p-value: 0.004068
```

Попробуем упростить регрессионную модель, избавившись от взаимодействия предикторов. Проверим при помощи функции `drop1()`, можно ли избавиться от взаимодействия.

```
drop1(Mod,test='F')

## Single term deletions
##
## Model:
## bwt ~ age * smoke
##           Df Sum of Sq      RSS      AIC F value Pr(>F)
## <none>                93062605 2485.2
## age:smoke  1    2609683 95672288 2488.5  5.1878 0.02389 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Поскольку уровень значимости меньше 0,05, то при удалении взаимодействия из модели, модель значимо изменится. Следовательно, взаимодействие из данной модели удалять нельзя.

В результате у нас получается две регрессионные модели:

№ 1. Для базового уровня:

$bwt = 2406.06 + 27.73 * age$ (т.е. для некурящих рожениц)

№ 2. Для объектов, не относящихся к базовому уровню:

$bwt = 2406.06 + 27.73 * age + 798.17 + (-46.57) * age$ (т.е. для курящих рожениц)

Далее визуализируем полученную регрессионную модель. Для классификации данных по группам используем функцию `group_by()`. Для этого активируем пакет `dplyr`.

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##   recode

## The following object is masked from 'package:MASS':
##
##   select

## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

new_data<-
newborn%>%group_by(smoke)%>%do(data.frame(age=seq(min(.$age),max(.$age),le
ngth.out=100)))
Predictions<-predict(Mod,newdata=new_data,se.fit=TRUE) #Рассчитаем
предсказанные значения
new_data$fit<-Predictions$fit
new_data$se<-Predictions$se.fit #Рассчитаем стандартные ошибки
t_crit<-qt(0.975,df=nrow(newborn)-length(coef(Mod))) #Рассчитаем
критические значения для 95% доверительного интервала
#Строим доверительный интервал
new_data$lwr<-new_data$fit-t_crit*new_data$se
new_data$upr<-new_data$fit+t_crit*new_data$se
```

Визуализируем данные на основе рассчитанных значений, а также добавляем данные наблюдений в виде точек при помощи геом (geom_point).

```
Plot_smoke<-
ggplot(new_data,aes(x=age,y=fit))+geom_ribbon(alpha=0.2,aes(ymin=lwr,ymax=
upr,group=smoke))+geom_line(aes(colour=smoke))
Plot_final<-
Plot_smoke+geom_point(data=newborn,aes(x=age,y=bwt,colour=smoke))+scale_co
lour_discrete('Пристрастие к курению',labels=c('Не
курит','Курит'))+labs(x='Возраст роженицы', y='Вес ребенка')
Plot_final
```



Рисунок 12 График предсказанных значений зависимой переменной с указанием исходных значений

После визуализации модели необходимо интерпретировать полученные результаты. График на рисунке 12 позволяет сделать вывод, что у некурящих

рожениц с возрастом есть тенденция рожать детей с большим весом, у курящих рожениц тенденция обратная. Расчеты показали, что существует значимое взаимодействие между предикторами «возраст» и «пристрастие к курению», поскольку уровень значимости был меньше 0,05. Данную тенденцию можно увидеть и на данном графике предсказанных значений.

Таким образом, R позволяет делать статистический анализ как в дескриптивной статистике, так и в более сложной, требующей построения регрессионных моделей как с линейной связью предикторов и отклика, так и с нелинейной, позволяет делать анализ временных рядов и работать с непараметрической статистикой, делать сложные визуализации для нескольких предикторов. Однако для того, чтобы выполнять все эти расчеты, нужно познакомиться с разными функциями и пакетами, созданными для программирования на языке R.

Практическое задание

Построить модель линейной регрессии на основании датасета whiteside в пакете MASS.

1.3. Сетевой анализ с использованием Pajek

Построение сети с использованием программного обеспечения Pajek

В последнее десятилетие большую популярность в социальных исследованиях набирает методология сетевого анализа. В политической науке в рамках неинституционализма зародился и бурно развивается сетевой подход [26, С. 14-32]. Конечно, этот подход имеет как ряд преимуществ, так и ряд недостатков. К сильным сторонам сетевого подхода относится возможность визуализации связей между разнообразными акторами, вычисление их роли в данном сетевом взаимодействии. В отличие от структурно-функционального анализа сетевой подход указывает на возможность построения горизонтальных связей между политическими акторами, например, [27, С. 68-86]. К слабым сторонам данного подхода относится сложность в сборе данных об акторах, по крайней мере, в России, отсутствие однозначной взаимосвязи между активностью в политической сети и изменением политической повестки дня. Кроме того, очень сложно доказать мобилизационный эффект социальных сетей, связь взаимодействия акторов онлайн и оффлайн [28], [29].

В социологии сетевая методология позволяет визуализировать отношения между акторами, которыми являются индивиды, а в политологии мы можем выстроить эти отношения между государственными учреждениями и организациями гражданского общества. Такая визуализация позволяет определить акторов (узлы), которые находятся ближе к центру или периферии сети, в связи с этим определить их роли, рассчитать центральности произвести ранжирование. Для автоматизации процесса построения сетей были разработаны специальные компьютерные программы: Pajek, UCINET, Gephi.

В нашей работе мы остановимся на программном обеспечении Pajek (от слова «паук»), разработанный словенскими социологами [30]. Pajek является бесплатным программным обеспечением, и его можно скачать по ссылке <http://mrvar.fdv.uni-lj.si/pajek/>. Для того, чтобы корректно скачать данное программное обеспечение, нужно выбрать зеленую иконку 32 или 64 bit в зависимости от возможностей вашего компьютера. Данный вариант Pajek рассчитан на 1000 узлов (акторов) и позволяет визуализировать информацию при помощи специального инструмента (карандаш в правом углу командной строки Сеть/Network). Другие варианты данной программы – Pajek XXL, Pajek 3XL – работают с количеством узлов значительно больше тысячи узлов и не позволяют визуализировать данную информацию, мы можем только производить расчеты по этим данным.

В социологии принято классифицировать социологические (социальные) исследования по трем типам: разведывательное, описательное и объяснительное [31, С. 43]. Эти типы исследования можно представить как движение по лестнице

вверх: от самого простого вида исследования к самому сложному. Всегда желательно проводить исследование на уровне объяснительного, т.е. с использованием серьезной аналитики, но, к сожалению, это не всегда возможно. И проблема не в том, что у исследователей не хватает аналитических способностей и навыков проведения таких исследований, камнем преткновения служат возможности соответствующего дизайна исследования. Так, сетевой анализ по определению является разведывательным (эксплораторным). Он не предполагает выдвижение конкретной гипотезы в момент подготовки программы исследования. Сначала мы строим сеть (визуализируем информацию), а потом с ее использованием пытаемся выявить закономерности и сделать значимые выводы.

Сетевой анализ в Pajek позволяет не только визуализировать информацию, но и делать несложные математические расчеты. Например, мы можем рассчитать плотность сети, коэффициенты корреляции Пирсона и Спирмена, показатели центральности. На основании расчетов мы можем также выделить наиболее связанные участки сети (ядра, кластеры). Таким образом, Pajek дает возможность комплексного анализа построенной сети.

Для построения сети нам необходимо четко сформулировать проблему исследования и, исходя из нее, идентифицировать акторов и наличие взаимодействий между ними. Все это можно сделать, используя ивент-анализ, контент-анализ и интервью по типу «снежного кома». К сожалению, использование интервью не всегда возможно, его возможности ограничены статусом опрашиваемого респондента. Если мы имеем дело с проблемой местного сообщества, то скорее всего встретиться и переговорить с депутатом муниципального совета для нас не составит труда. Гораздо сложнее, когда в качестве респондентов выступают представители государственных органов федерального уровня [32, Р. 47-50]. Здесь мы ощущаем эффект «стеклянного потолка», и поэтому лучше использовать контент-анализ (анализ интернет-медиа и традиционных СМИ) или ивент-анализ (анализ участников знаковых политических мероприятий). Когда нужно остановиться в сборе данных? Тогда, когда информация начнет повторяться.

В основе визуализации в сетевом анализе лежит теория графов, где граф – это набор вершин и линий между парами вершин. Сеть – это такой же вариант графов, но уже с подписанными вершинами. Сеть в сетевом анализе бывает двух видов: направленная и ненаправленная. Направленная сеть предполагает, что вершины связаны между собой направленными дугами, в то время как в ненаправленной сети вершины связаны двухсторонними дугами. В зависимости от типа сети мы можем использовать к ней те или иные аналитические процедуры, заложенные в программное обеспечение.

Обсуждение использования Pajek начнем с построения сети. Сеть в Pajek можно построить двумя способами: 1) при помощи написания матрицы

сопряженности в программе Блокнот/Notepad; 2) используя функции диалогового окна Сеть/Network в Pajek.

Возьмем некоторую гипотетическую страну Хэппилэнд, в которой есть три уровня власти. Федеральный уровень власти представлен президентом, премьер-министром и тремя министрами. Региональный уровень представлен губернатором, вице-губернатором и тремя руководителями комитетов. Муниципальный уровень представлен главой муниципального образования, заместителем главы муниципального образования и тремя специалистами. Всего у нас в данной сети получается 15 акторов.

При написании мини-программы для построения сети с использованием программы Блокнот мы должны отдельно прописать вершины и матрицу сопряженности.

```
*Vertices 15
 1 "President"
 2 "Prime-minister"
 3 "Minister 1"
 4 "Minister 2"
 5 "Minister 3"
 6 "Governor"
 7 "Vice-governor"
 8 "Committee head 1"
 9 "Committee head 2"
10 "Committee head 3"
11 "Municipal head"
12 "Minisipal deputy"
13 "Specialist 1"
14 "Specialist 2"
15 "SPecialist 3"
```

Таблица 3 Матрица сопряженности в программе Блокнот/Notepad

```
*Matrix
0 4 2 3 3 2 1 1 1 0 1 0 0 0 0
2 0 3 3 3 2 0 0 0 0 1 0 0 0 0
1 2 0 2 2 1 0 0 0 0 0 1 0 0 0
1 2 1 0 1 1 0 0 1 0 1 0 1 0 0
2 2 1 1 0 1 0 0 1 0 0 1 0 0 0
2 2 1 1 2 0 2 2 2 2 2 0 1 1 1
1 0 1 0 0 2 0 1 1 1 1 0 0 0 0
0 0 1 0 1 2 1 0 1 1 1 0 0 1 0
0 0 0 1 1 2 2 0 0 0 1 0 1 0 0
0 1 1 0 1 2 2 1 1 0 1 0 0 0 1
0 0 0 0 0 3 3 1 1 1 0 1 1 1 1
0 0 0 0 0 0 0 1 0 1 3 0 1 1 1
0 0 0 0 0 0 0 0 0 1 2 1 0 1 1
```

```
0 1 0 0 0 0 0 0 0 3 1 1 0 1
0 0 1 0 0 0 0 0 0 1 2 1 1 0
```

При написании программы мы начинаем со звездочки (*), далее с большой буквы мы указываем наименование Vertices и количество вершин «15». Далее на каждой строке через пробел мы прописываем номер и название вершины в кавычках. По умолчанию все вершины в Pajek круглые, но мы можем их сделать, например, квадратными, указав после названия вершины square. После перечисления вершин без лишней строки разделителя пишем звездочку (*) и Matrix (матрицу сопряженности). В матрице сопряженности по горизонтали мы указываем тех, кто выбирает; по вертикали тех, кого выбирают. Например, вершина «President» (первая строка) выбирает «Prime-minister» довольно часто (4 вертикаль). Цифры показывают значение линий или силу связи. Так «0» означает, что связи между этими акторами (вершинами) нет, «1» показывает, что связь есть, но она не очень сильная. Чем больше используемое число, тем больше связь. Между цифрами в матрице сопряженности должны быть пробелы, на диагонали должны быть нули, поскольку мы не можем выбирать сами себя. Данные в файле пишутся на английском, и сам файл мы сохраняем латинскими буквами. С русскими обозначениями Pajek не работает. После того, как программа написана, мы открываем ее через Pajek при помощи команды File>Network>Read. Если все правильно, то во вспомогательном окне будет указано, что «прочитано такое-то количество линий». Далее мы можем визуализировать построенную сеть при помощи «карандаша» в правом углу диалогового окна Сеть/Network.

При визуализации открывается окно для рисования. Здесь для нас важными функциями являются подразделы меню окна рисования Layout, Options, Export. При помощи команды Kamada-Kawai в подразделе Layout мы можем видоизменять (энергенизировать) нашу сеть до тех пор, пока мы не сочтем визуализацию наиболее удачной. При помощи подраздела Options мы можем отображать вершины, обозначенные либо именами, либо цифрами; показывать значение связей на линиях; менять толщину линий и цвет фона. При помощи подраздела Export мы можем выкачать полученный рисунок сети в формате eps, jpg, что является важным условием при написании презентаций.

Давайте сохраним полученную сеть в файл HappyLand.net. Теперь мы можем создать проектный файл, добавив к нему файл с классификацией. Напомним, что у нас есть три уровня власти в данном государстве. Давайте сгруппируем акторов в три группы соответственно. Для этого используем файл Partition.clu.

Теперь можно сделать совместную визуализацию построенной сети и классификации (partition). В разделе сеть открываем нашу сеть HappyLand.net, в разделе классификации ставим наш файл Partition.clu, помечаем галочкой. Визуализация сети со сгруппированными вершинами представлена на Рис. 13.

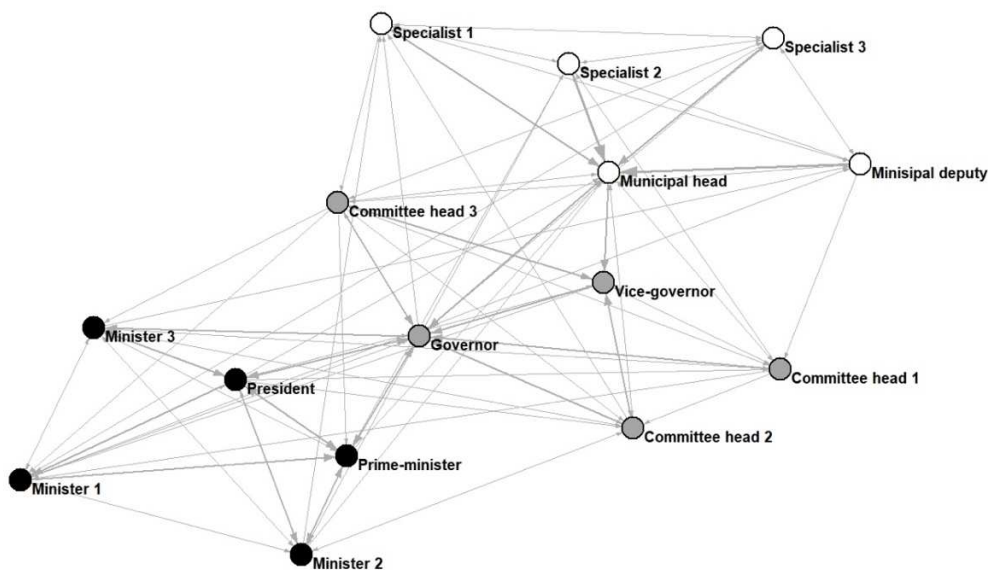


Рисунок 13 Визуализация уровней власти в стране Хэппилэнд.

Теперь можно создать проектный файл с расширением .raj. Для этого нужно указать в первом окне сети нашу сеть HappyLand.net, в первом окне классификации наш файл Partition.clu. Затем выполнить команду File>Pajek Project File>Save. Теперь у нас появился проектный файл HappyLand.raj. С ним можно работать далее. Теперь мы можем воспользоваться командой File>Pajek Project File>Read, и в окне сети, и в окне классификации одновременно откроются документы, сохраненные в проектном файле.

Второй способ построения сети позволяет нам использовать непосредственно функционал Pajek. Для начала необходимо создать пустую сеть при помощи команды Network>Create New Network>Empty Network. В диалоговом окне нужно напечатать то количество вершин, которые нам нужны. У нас появляется новая сеть в Network drop-down menu. Если визуализировать данную информацию, то пока мы увидим только набор вершин, расположенных в виде эллипса. Продолжаем рисовать, используя команду File>Network>View/Edit. Эта команда открывает диалоговое окно, которое позволяет выбрать вершину по ее порядковому номеру или названию. Появляется Editing Network screen. В нем мы двойным щелчком активируем Newline, что позволяет пользователю добавить линию в/из выбранной вершины. Например, вершина «1». Чтобы добавить двухстороннюю дугу, нужно просто напечатать номер другой вершины. Если ввести знак «+» перед новой вершиной, то добавляется дуга в выбранную вершину (дуга в вершину «1»). Если ввести знак «-», то будет добавлена дуга из выбранной вершины, т.е. из вершины «1» в новую вершину. Каждая новая линия или дуга отражается как линия в Editing Network Screen. Обозначение «4.1» — это дуга из вершины «4» в вершину «1». Обозначение «1.3» — это дуга из вершины «1» в вершину «3». И, наконец, «1-2» — это линия (двухсторонняя дуга) из вершины 1 в вершину 2. Линию можно удалить, кликнув по ней дважды.

Полученную сеть нужно сохранить с расширением «.net». Далее через программу Блокнот/Notepad мы можем редактировать названия вершин, так как Pajek обозначает их цифрами. В ручном режиме, открыв документ в Блокноте, мы сможем переименовать данные вершины. Вообще важно помнить, что Pajek не сохраняет сети и их модификации по умолчанию. Если нужно что-то сохранить, то желательно переименовать сеть в момент сохранения, чтобы не стереть предыдущие данные.

В Pajek мы можем работать с шестью разными диалоговыми окнами, которые соответственно открывают документы с разными расширениями. Это Сеть/Network, Классификации/Partition, Векторы/Vectors, Перестановки/Permutation, Кластеры/Clusters, Иерархии/Hierarchy. В случае классификаций документы имеют расширение «.clu», в случае векторов расширение «.vec». Также мы можем работать с проектными файлами с расширением «.raj», которые объединяют несколько разных типов документов в Pajek.

Для более подробного знакомства с программным обеспечением Pajek можно приобрести учебник W. de Nooy, A.Mrvar, V.Batagelj «Exploratory social network analysis with Pajek» с базами данных [de Nooy, Mrvar, Batageli, 2018. с. 36-222, 149-169]. Задачей же данного учебного пособия является общий обзор возможностей использования Pajek с объяснением сути наиболее часто используемых команд.

Использование классификаций для упорядочивания данных

Следующим диалоговым окном Pajek являются Классификации/Partitions. Они позволяют распределить вершины по принадлежности к определенным группам. Группы задаются исследователем произвольно в соответствии с дизайном исследования. Посмотреть принадлежность вершин к группам можно при помощи View/Edit на вкладке Partitions. Посмотреть распределение вершин по группам (частотный анализ) можно при помощи команды Info на вкладке Partitions. Классификации/Partitions позволяют не только распределять вершины (узлы) по группам, но и производить некоторые манипуляции с сетью. Допустим, мы хотим посмотреть, как выглядят отношения между акторами на федеральном уровне. Для этого мы используем команду Operations>Network+Partition>Extract>SubNetwork Induced by Union of Selected Clusters. В появившемся окне мы набираем цифру «1», которой закодирован данный уровень власти. Результаты визуализации представлены на Рис. 14.

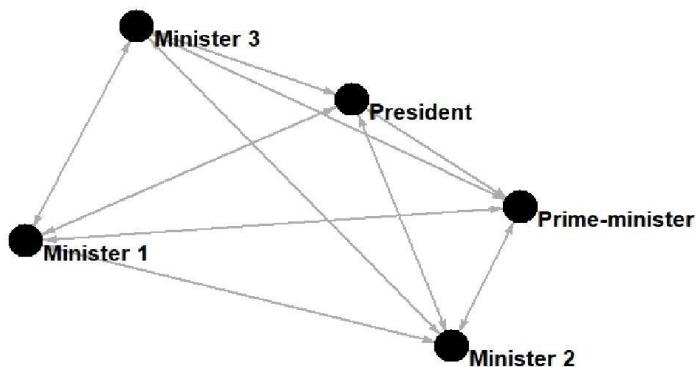


Рисунок 14 Отношения между акторами на федеральном уровне власти в стране Хэппилэнд.

Если мы хотим посмотреть отношения между уровнями власти, мы можем сжать совокупность акторов на каждом уровне до одного узла. Для этого мы используем команду Operations>Network+Partition>Shrink Network. В диалоговом окне, в котором нас спрашивают о минимальном количестве связей, мы ставим «1». Названия уровней власти добавляем вручную через функцию File>Partition>View/Edit. По умолчанию каждый уровень власти будет обозначен названием должности, которая стоит первой в списке по алфавиту. Данная команда позволяет нам посмотреть характер связей между уровнями власти, Рис. 15.

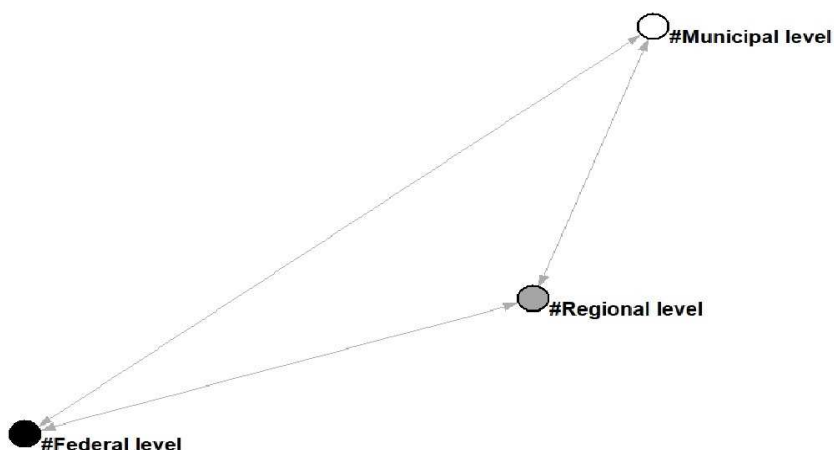


Рисунок 15 Отношения между уровнями власти в стране Хэппилэнд

Если мы захотим проверить, как выглядит отношения между уровнями власти и внутри конкретной властной группы, мы можем использовать ту же команду, но указав во второй строке диалогового окна номер группы, который не

нужно «вырезать», например, единицу. Тогда мы получим следующий рисунок сети (см. Рис. 16).

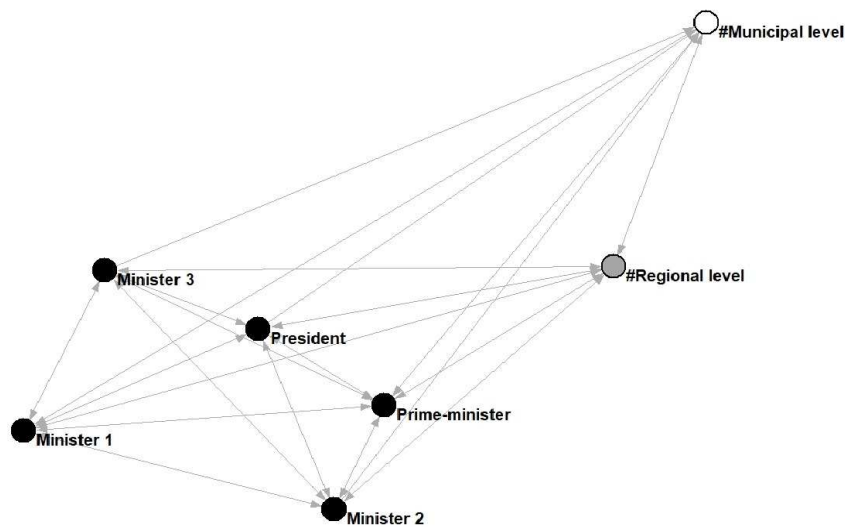


Рисунок 16 Отношения между акторами внутри федерального уровня власти с другими уровнями власти.

Информация, касающаяся частотного распределения акторов, расчеты показателей сети может являться важной, но отображается на вспомогательной вкладке *Rajek*. Для того, чтобы вызвать эту вкладку, необходимо выполнить следующую команду: *Info>Child Windows>Report Window>Show*.

Для визуализации сетей можно использовать не только Классификации/*Partitions* и Векторы/*Vectors*, но и другие инструменты, такие, как Компоненты, Ядра/*K-cores*, Кластеры. Эти инструменты позволяют определить наиболее плотные участки сети. О них мы поговорим подробнее.

Определение плотных участков сети при помощи *Rajek*

Плотные участки сети связаны с тем, что акторы, образующие сети, в своих подгруппах могут быть связаны не только взаимодействием, но и солидарностью, групповыми нормами, общими интересами. Более тесные контакты внутри подгрупп сети связаны с явлением гомофилии или ассертивности, когда подобные акторы чаще контактируют с себе подобными.

Прежде чем перейти к объяснению визуализации плотных участков сети, определим простые числовые характеристики сетей. Начнем с плотности сети. Плотность сети представляет собой отношение числа всех существующих взаимосвязей в сети ко всем возможным. Полная сеть – это такая сеть, в которой все вершины взаимосвязаны. Ее плотность равна единице. Она берется за эталон. Все остальные сети сопоставляются с этой эталонной сетью, и их плотность колеблется в пределах 0,05-0,6. Множественные линии между вершинами служат индикатором более плотных участков сети. С увеличением размера сети увеличивается ее плотность.

Следующей базовой числовой характеристикой является степень вершины, которая определяется количеством линий, связанных с вершиной. Это более полезная характеристика сети, так как не зависит от ее размера. Необходимо отметить, что вершины с большей степенью находятся в более плотных участках сети. Более высокая степень вершин ведет к увеличению плотности сети, так как такие вершины содержат больше связей. Еще одним базовым показателем является средняя степень. Он используется для измерения структурной связанности сети. Подобно степени вершины, данный показатель не зависит от размера сетей и может быть использован для сравнения сетей.

Для расчета плотности сети и средней степени вершин используем команду [Main] Network>Info>General. Если нам интересна только плотность и средняя степень, то вводим 0. Pajek показывает плотность с петлями и без. Нас интересует показатель плотности сети без петель.

Для того, чтобы определить среднюю степень сети для ненаправленной сети, преобразуем HappyLand.net в HappyLand_symmertized.net, используя команду Network>Create New Network>Transform>Arc->Edges>All. В диалоговом окне нам предлагают указать одну из 6 опций, выберем «1». Сохраняем новую сеть HappyLand_symmertized.net при помощи команды File>Network>Save. Далее мы можем рассчитать степень группы, используя команду: Network>Create Partition>Degree>submenu: Input, Output, All. Поскольку сеть ненаправленная, мы можем выбрать любой из предложенных вариантов (Input, Output, All). Для того, чтобы посмотреть степень группы, используем команду Partition>Info. Номер кластера отражает степень группы, но мы не можем рассчитать среднюю степень группы, поэтому считаем среднюю степень сети. Считаем среднюю степень для ненаправленной сети Attiro_symmertized.net, используя команду Network>Info>General. Во вспомогательной вкладке получаем плотность сети 0,65, средняя степень сети 9,07. Это означает, что плотность сети довольно высока, и в среднем представители власти довольно часто контактируют друг с другом.

Плотные участки сети можно идентифицировать с использованием ряда команд Pajek и получить не только их числовые характеристики, но и визуализировать. Первой характеристикой плотных участков сети служат компоненты. Они бывают слабые, т.е. представляют собой максимально слабо связанную подсеть, и сильные, которые представляют собой максимально сильно связанную подсеть. Для характеристики слабых и сильных компонентов сети используется понятие тропы (полутропы). Тропой называется такая ситуация, когда в сети мы можем двигаться только по направлениям дуг, не повторяя вершины. Такая сеть является сильно связанной. Если же мы пренебрегаем направлением дуг и движемся по сети в любом направлении, но не повторяя вершины, то такая сеть называется слабо связанной. Для нахождения сильных компонентов сети используем направленную сеть Attiro.net. Команда Network>Create Partition>Component>Strong позволяет выявить одну сильную

компоненту сети, которую также можно представить визуально при совместном использовании сети и классификации. Поскольку количество слабых компонентов в направленной сети равно общему количеству компонентов в ненаправленной сети, используем сеть HappyLand_symmertized.net и команду Network>Create Partition>Component>Week. В ненаправленной сети количество слабых компонент равно единице.

Характеристика Ядра/К-cores также указывает на плотные участки сети, где К-cores – это кластеры, а К — это минимальная степень каждой вершины внутри ядра. Так, 3-cores содержит все вершины, которые связаны со степенью 3 и более с другими вершинами. Таким образом, К-core – это максимальная подсеть, в которой каждая вершина имеет, по крайней мере, степень К в подсети. Данные ядра «гнездятся друг в друге». Так, вершина в 3-Cores является частью 2-Cores. Однако не все члены 2-Cores принадлежат 3-Cores. Таким образом, удаление из сети ядер с более низкой степенью позволяет идентифицировать более плотные участки компоненты сети.

Еще одна характеристика плотности сети – это клика. Клика представляет собой набор вершин, в котором каждая вершина напрямую связана со всеми другими вершинами, т.е. подсеть максимальной плотности состоит из трех вершин и более. Клики могут перехлестываться, так как одна и та же вершина может принадлежать разным кликам. Минимальная клика представляет собой триаду, т.е. максимально плотный участок сети, состоящий из трех вершин. Для определения всех полных триад внутри сети используем фрагмент сети triad_undir.net в первом окне и ненаправленную сеть HappyLand_symmertized.net. Далее найдем все полные триады в сети, используя команду Networks>Fragment (First in Second). В разделе Иерархия/Hierarchy появится новый документ. Корень иерархии – это узел, который связывает все группы. Он равен количеству вершин во всех группах: 126 групп умножить 3 вершины равно 378. Чтобы посмотреть на набор триад, заходим в иконку View/Edit Hierarchy и кликаем на плюс слева около корня. После этого раскрывается набор из «126» триад, в каждой из которых можно посмотреть, какие вершины входят, кликнув на левую кнопку мыши.

Расчеты центральностей вершин и централизации сети

Актеры в сети могут располагаться или ближе к центру, или ближе к периферии. Raјek позволяет произвести расчеты показателей центральности акторов и централизации сети. Показатели центральности используются для определения позиций индивидуальных вершин. Показатель централизации характеризует всю сеть. Высокоцентрализованная сеть имеет четкую границу между центром и периферией. В этой сети распространение информации идет быстро, но центр незаменим для передачи информации.

Для измерения центральности используются следующие показатели: степень вершины, близость к центру, центральность по посредничеству и центральность

по собственному вектору. Степень центральности вершины – это ее степень, т.е. количество линий, которые она содержит. Для анализа показателей центральности используем наш файл HappyLand.paj.

Степень вершин в Pajek вычисляется с использованием команды Network>Create Partition>Degree>All. Степень вершин можно посмотреть при помощи команды Partition>Info. Номер кластера – это степень вершины или количество линий, связанных с данной вершиной. В Pajek группирование (partition) – это классификации, приписывающие вершины кластерам. Однако для вычисления центральностей вершин и централизации сети используют векторы. Для того, чтобы рассчитать степень централизации сети, необходимо использовать команду Network>Create Vector>Centrality>Degree. В нашей сети степень централизации составляет 0,32. Этот показатель имеет смысл только при сравнении с другими показателями.

Следующий показатель – это близость вершины к центру, который определяется как количество остальных вершин, деленное на сумму всех расстояний между этой вершиной и другими вершинами. Централизация близости – это вариация близости вершин к центру, деленное на максимальную вариацию близости к центру, возможную для сети данного размера. Расчет показателей близости вершин к центру и централизации близости осуществляется посредством команды Network>Create Vector>Centrality>Closeness>All. Значение показателей близости вершин к центру можно посмотреть при помощи команды View/Edit.

Центральность по посредничеству показывает, насколько часто актер является посредником между любыми другими двумя участниками, находясь на кратчайшем пути между ними. Централизация по посредничеству представляет собой отношение вариации центральности по посредничеству к максимальной центральности по посредничеству, которая возможна в сети данного размера. Для расчета этих показателей воспользуемся командой Network>Create Vector>Centrality>Betweenness. Показатели централизации отражаются во вспомогательной вкладке Pajek (Report screen).

Показатель по собственному вектору демонстрирует зависимость между положением актора (например, его приближенности к центру) и центральности связанных с ним других акторов. Наибольший показатель центральности по собственному вектору будет у того актора, у которого много связей, и который связан с другим актором, у которого также много связей. Для расчета центральности по собственному вектору и показателя централизации сети по собственному вектору используем команду Network>Create Vector>Centrality>Hubs-Authorities.

Еще одной характеристикой положения актора в сети является расстояние, которое определяется как количество шагов или посредников для кого-то, чтобы достичь индивида в сети. Чем меньше расстояние между индивидами, тем легче обмен информацией. Geodesic – это наиболее короткое расстояние между

вершинами. В нашем примере наибольшей степенью обладает губернатор (Governor). Для расчета расстояния между данным актором и другими вершинами используем команду: Network>Create Partition>k-neighbors. Данная команда создает группы классов, указывающие на расстояние между актором Governor и другими вершинами.

Кратчайшее расстояние между двумя вершинами можно найти, используя команду: Network>Create New Network>Subnetwork with Path>All Shortest Path between Two Vertices. Найдем кратчайшее расстояние между двумя крайними вершинами «President» и «Committee head 1». На вопрос «Forget values of lines?» отвечаем утвердительно, так как мы не хотим взвешивать линии в соответствии с их значениями. Кратчайшее расстояние между этими вершинами равно двум.

Таким образом, программное обеспечение Pajek дает нам богатые возможности по построению и визуализации сетей, являясь важным инструментом при проведении сетевого анализа не только в социологии, но и в политической науке. Его несомненным преимуществом является также возможность математической оценки положения акторов и определения их ролей в сети. Методология сетевого анализа является, несомненно, уникальным подходом к исследованию социальных и политических феноменов, и ее недостаток в отношении объяснительного потенциала будет со временем преодолён.

Практическое задание

Построить сеть дружбы из 10 акторов с использованием команды Kamada-Kawai, посчитать центральности вершин и дать им интерпретацию.

1.4. Сбор текстовых данных

Виды данных, генерируемых пользователями, подходы к их получению и систематизации

Пользователи генерируют большое количество информации, оставляя свои следы в интернете. По отдельности эта информация не представляет интереса, но, когда речь идет о большом количестве пользователей, такие данные уже можно использовать для анализа.

Для понимания процесса сбора информации из сети важно выделить основные типы источников, это позволит более точно подобрать инструмент и правильно спланировать исследование. Для проведения исследований можно выделить следующие типы ресурсов:

- соцсети;
- блоги;
- новостные ресурсы;
- интернет-приемная;
- интернет-ресурсы для размещения общественных инициатив;
- личные дневники;
- другие ресурсы, на которых контент генерируют пользователи;

Рассмотренные далее инструменты позволяют проводить сбор информации со всех типов ресурсов.

Для качественного анализа нужен большой массив данных, его можно собирать вручную, копируя текст, либо, применить автоматизированный подход.

Существует 2 основных подхода для автоматизированного сбора информации с веб ресурсов:

- парсинг данных через API;
- парсинг данных путем скачивания контента.

Сбор данных через API

Для разработчиков и опытных пользователей, с целью быстрого получения массивов данных в машиночитаемых форматах на сайтах часто реализована возможность подачи запросов посредством API.

API — это способ программного взаимодействия с отдельным программным компонентом или ресурсом. API (application programming interface) - программный интерфейс приложения, интерфейс прикладного программирования. API определяет взаимодействие между несколькими программными посредниками. Он определяет виды вызовов или запросов, которые могут быть сделаны, форматы данных, которые следует использовать, соглашения, которым следует следовать, и т. д.

API также может обеспечить механизмы расширения, чтобы пользователи могли расширять существующие функциональные возможности различными способами и в разной степени. API может быть полностью настраиваемым, специфичным для компонента или разработанным на основе отраслевого стандарта для обеспечения совместимости.

API представляет собой совокупность различных инструментов, функций, реализованных в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой.

В общем случае данный механизм применяется с целью объединения работы различных приложений в единую систему. Это достаточно удобно для исполнителей. Ведь в таком случае к другому приложению можно обращаться как к «черному ящику». При этом не имеет значения его внутренний механизм.

Важно, чтобы исследователь понимал общий принцип, что, если сайт или сервис имеет публичный API, это означает, что вы можете подключиться к базе данных этого сайта и с помощью определенных запросов получить информацию в структурированном виде.

Для работы с API сайтов можно использовать разные языки программирования, это может быть PHP, C, Python и другие. Если исследуемый сайт имеет API, можно ознакомиться с документацией и заранее выяснить, какую информацию вы сможете выкачивать и какими методами.

Чтобы понять, есть ли на ресурсе API, нужно найти соответствующий раздел на сайте. Если визуально раздел найти не удалось, это еще не означает, что его нет. К примеру, изучение главной страницы "Портал открытых данных Российской Федерации" <https://data.gov.ru> не говорит нам о том, что на сайте есть API, но, если воспользоваться поиском и ввести комбинацию "data.gov.ru api", мы найдем соответствующий раздел: <https://data.gov.ru/api-portala-otkrytyh-dannyh-rf-polnoe-rukovodstvo>, документ содержит полную информацию по использованию API Портала открытых данных РФ.

Это инструкция к действию, в которой описан формат хранения данных, методы и способы подключения, перечислены доступные сервисы, а также приведены примеры запросов.

Также, при работе с API важно понимать такие форматы данных, как XML и JSON. Это самые распространенные форматы вывода результатов запроса.

Используя доступное API, можно выгружать разнообразную информацию, к которой предоставлен доступ.

Пример работы с API для выгрузки данных из электронной библиотеки ELIBRARY.RU

Для хранения данных научных публикациях используются цифровые библиографические и реферативные базы данных. Такие базы представляют собой инструмент для отслеживания цитируемости статей, опубликованных в научных

изданиях, являются одним из главных источников получения наукометрических данных для проведения оценочных исследований. К международным реферативным базам относятся, например, Web of Science и Scopus.

В России основная база - eLIBRARY. eLIBRARY.RU - крупнейшая в России электронная библиотека научных публикаций, обладающая богатыми возможностями поиска и получения информации. Библиотека интегрирована с Российским индексом научного цитирования (РИНЦ), созданным по заказу Минобрнауки РФ бесплатным общедоступным инструментом измерения и анализа публикационной активности ученых и организаций.

Основная задача API eLIBRARY.RU - предоставить научным организациям и авторским коллективам возможность выгрузки научной и библиометрической информации из базы данных РИНЦ для проведения научных и статистических исследований.

Основные сервисы API eLIBRARY.RU и их возможности

- Получение библиометрических показателей авторов. Сервис предусматривает выгрузку по уникальному коду ученого в РИНЦ аналитических показателей автора со страницы "Анализ публикационной активности автора".
- Получение библиометрических показателей журналов. Сервис предусматривает выгрузку по уникальному коду журнала в РИНЦ аналитических показателей журнала со страницы "Анализ публикационной активности журнала".
- Получение библиографической записи публикации в РИНЦ.

API сайта Elibrary представляет собой набор сервисов для получения данных о публикациях, размещённых в научной электронной библиотеке. Всего доступно 10 сервисов, каждый из которых имеет собственный идентификатор, название и список обязательных и необязательных параметров.

Доступ к API Elibrary предоставляется только для статических IP адресов. Статический адрес отличается от динамического тем, что последний может меняться каждый раз, когда пользователь посещает Интернет, статический же будет закреплен за абонентом и не будет меняться при выходе в Интернет. Статический IP адрес приобретается у Интернет-провайдера. Каждому IP адресу, имеющему доступ к API, присваивается уникальный код пользователя, который необходимо передавать как один из параметров для каждого сервиса.

В проекте исследователей ИТМО [33] разработка программы для получения данных из библиотеки Elibrary с помощью API сервисов осуществлялась на языке C# в Visual Studio 2019. C# является современным объектно-ориентированным языком программирования, а среда разработки Visual Studio 2019 содержит удобный менеджер пакетов NuGet, в котором можно скачать уже готовые наборы функций и типов данных для своих разработок.

Входными данными разработанной программы являются файл в формате .csv, содержащий список ссылок на публикации в библиотеке, а также INI файл настроек, содержащий код пользователя, выдаваемый IP адресу. Выходными данными является тот же файл, что подавался на вход, но дополненный информацией, полученной запросами к API.

Входной и выходной файлы имеют определённый набор колонок. Для работы с файлами с помощью менеджера NuGet была скачана библиотека CsvHelper. Она позволяет читать и записывать данные по колонкам, используя объекты со свойствами. Объектом является публикация, то есть строка из файла, а свойствами – колонки, по которым будут получены данные о публикации.

Главное окно программы представлено на рисунке 17. Оно состоит из кнопки открытия входного csv файла и строки, в которую будет помещён путь до этого файла.

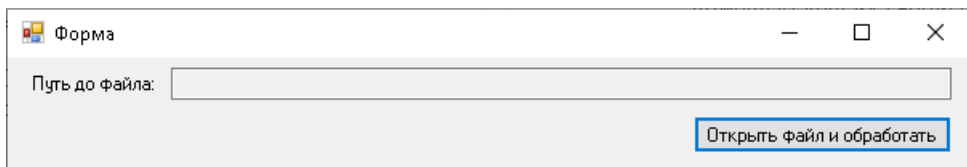


Рисунок 17 Пример интерфейса программы для сбора данных

После чтения списка ссылок на публикации из входного файла запускается процесс формирования асинхронных запросов к сервисам API. Формирование, отсылка и чтение ответов на запросы было реализовано с помощью библиотеки RestSharp, предназначенной для работы с HTTP REST API. Асинхронность запросов означает, что одновременно запрашивается информация по нескольким публикациям, что позволяет увеличить скорость формирования результирующих данных.

В программе реализовано использование двух сервисов Elibrary API: GetItem2 (026) и GetRefs (040). Оба сервиса имеют три обязательных параметра:

- ucode – уникальный код пользователя, полученный для IP адреса;
- sid – ИД сервис: 026 или 040;
- itemid – ИД публикации из библиотеки (извлекается из ссылки).

Полученные параметры и ссылка на API передаются в объект RestRequest, который формирует запрос и возвращает ответ от сервера.

Сервисы API научной библиотеки возвращают ответ в формате XML, который представляет собой набор тэгов и информации, заключённой в них.

Пример ответа по запросу выше:

```
<root>
<request>
<remoteAddr>5.189.197.4</remoteAddr>
<itemid>23455064</itemid>
</request>
```

```
<references numOfRefs="26">
...
<references refNum="3" subRefNum="" targetId="13121604">
<text> Быков И.А. Интернет-технологии в избирательной кампании Барака
Обамы//Вестник Пермского университета. Серия: Политология. 2010. № 1 (9). С. 48-
58</text>
</references>
<references refNum="4" subRefNum="" targetId="">
<text> Вершинин М.С. Политическая коммуникация в информационном обществе:
перспективные направления исследований//Актуальные проблемы теории коммуникации:
сб. научн. трудов. СПб.: Изд-во СПб "ПУ, 2004. С. 98-107.</text>
</references>
...
</root>
```

В ответ на запрос списка цитируемой литературы получен список публикаций с указанием авторов, наименования, источника, даты издания и страниц. Так же, если публикация размещена в библиотеке Elibrary, то в теге references параметр targetId будет содержать идентификатор, по которому возможно получить больше информации с помощью других сервисов API.

Полученная от сервисов информация сохраняется в соответствующие поля объекта публикации, а затем переносится в выходной файл.

Таким образом, используя API, можно получить информацию о неограниченном количестве публикаций за короткий промежуток времени.

Пример работы с API для выгрузки данных Twitter

Для более широкого доступа к информации Твиттер предоставляет компаниям, разработчикам и пользователям программный доступ к данным Твиттера с помощью API-интерфейсов.

Твиттер предоставляет доступ к некоторым своим службам с помощью API-интерфейсов, чтобы программисты могли разрабатывать программное обеспечение, тесно взаимодействующее с Твиттером. API-платформа предоставляет широкий доступ к открытым данным Твиттера, которыми пользователи готовы поделиться со всем миром.

Основные типы данных, которые вы можете получить:

- Твитты и ответы. Для доступа к твиттам разработчики могут использовать поиск по определенным ключевым словам или запрашивать образцы твиттов отдельных учетных записей.
- Учетные записи и пользователи. Разработчики программным образом управляют профилем и настройками учетной записи, добавляют пользователей в список игнорируемых или черный список, запрашивают информацию о разрешенных действиях учетной записи и др.

К примеру, международная команда исследователей [34] разработала уникальную вычислительную модель для прогнозирования распространения

сезонного гриппа в режиме реального времени. Они используют сообщения в Твиттере, собранные через API, в сочетании с ключевыми параметрами эпидемии каждого сезона, включая инкубационный период заболевания, уровень иммунизации, количество людей, которых может заразить человек с вирусом, и присутствующие вирусные штаммы.

В работе по изучению городских сообществ [35], мы использовали анализ сообщений из социальных сетей, в том числе из Твиттера, основная задача – по имеющемуся набору данных, представляющих собой сообщения пользователей и собранных на разных площадках, автоматически определить, какой объект городского хозяйства упоминается в тексте отзыва, а также определить отношение пользователя к данному объекту (позитивное, нейтральное или негативное). Парсинг в Твиттере позволяет производить поиск по координатам с заданным радиусом поиска.

Основная особенность парсинга в Твиттере - ключ разработчика. Потребуется написать обоснование, в каких целях вы планируете использовать приложение, пройти модерацию. Дополнительно, Твиттер предлагает доступ API Twitter для академических исследователей.

Вся основная документация собрана в разделе <https://developer.twitter.com/en>.

Сбор данных без использования API

Современные ресурсы, такие как vk.com, twitter.com, roi.ru, change.org и другие популярные платформы, имеют интерфейсы API, позволяя мгновенно обмениваться информацией с площадками, выгружать доступные данные.

Однако большая часть ресурсов, блоги, форумы, интернет-приемные органов власти, как правило, не имеют подобных интерфейсов. Для сбора информации с таких ресурсов существует иной метод парсинга (веб-скрейпинга) – это процесс автоматизированного сбора структурированных веб-данных с помощью специального программного обеспечения.

Основные варианты использования включают мониторинг цен, ценовую разведку, мониторинг новостей, лидогенерацию и маркетинговые исследования.

Как правило, извлечение веб-данных используется людьми и компаниями, которые хотят использовать огромное количество общедоступных веб-данных, или для копирования информации, к примеру, это отличный способ сохранить все товары магазина-конкурента.

Если вы когда-либо копировали и вставляли информацию с веб-сайта, вы выполняли ту же функцию, что и любой веб-парсер, только в микроскопическом ручном масштабе. В отличие от утомительного процесса ручного извлечения данных, веб-парсинг использует автоматизацию для извлечения сотен, миллионов позиций.

Парсинг — это принятое в информатике определение последовательного синтаксического анализа информации, размещённой на интернет - страницах. Парсер — это программа или скрипт, позволяющая выполнить такой анализ и представить результат в нужном для пользователя виде.

Парсинг сайтов, описанным способом, является эффективным решением автоматизации сбора и обработки информации.

Основной принцип работы состоит в том, что весь контент (текст) содержится в разметке HTML, которая повторяется.

Пример - мы хотим выгрузить все сообщения (только текст, без персональных данных) интернет-приемной Администрации Советского района города Челябинска по адресу <http://sovadm74.ru/online-priemnaya>, и дальше анализировать текст. Приемная содержит более 250 страниц, на каждой странице по 10 обращений, 2500 обращений за 8 лет. Это может стать интересной информацией – как менялись запросы граждан за 8 лет. Данный сайт не имеет API, как мы выяснили ранее, это значит, что у нас не получится обратиться к базе данных и быстро получить информацию. Именно для таких целей и нужны программы для парсинга.

Если говорить коротко, то такое программное обеспечение просто откроет все 250 страниц и по заданному алгоритму заберет текст обращений, при этом оператору не обязательно обладать навыками программирования, достаточно понимать основы HTML, чтобы сделать разметку на примере одного поста, дальше программа все сделает сама.

Принцип работы парсера контента можно разделить на 4 основных этапа:

- Шаг первый - разметка одной страницы. Мы выделяем границы парсинга на одном примере. Это единственная операция, которая требуется от оператора;
- Шаг второй - получение исходного кода html - страницы. На этом шаге выполняется копирование исходного кода страницы с дальнейшим извлечением из неё информации;
- Следующий шаг - извлечение из полученного кода нужной информации. Получив исходный код html - страницы, необходимо выполнить над ним обработку, т.е. отделить искомый текст от гипертекстовой разметки, выстроить иерархическое дерево элементов документа (DOM) и извлечь из страницы искомую информацию. По заданным критериям выделить только основную информацию, которая представляет интерес. Для сбора важных параметров задаются границы парсинга каждого поля, на примере одной страницы. К примеру, название страницы всегда содержится в метатеге: `<title>Название страницы</title>`;
- Четвертый шаг - конвертация собранных данных в нужный формат;

- Последний шаг - сохранение результата. После успешного извлечения данных страницы их необходимо сохранить в требуемом виде для дальнейшей обработки.

В итоге с помощью парсинга с заданных URL происходит сбор основных данных страницы. Общая схема работы представлена на рисунке 18.

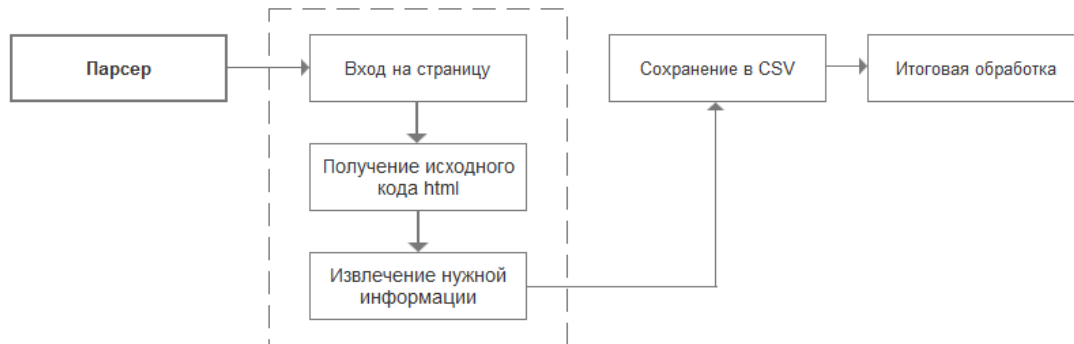


Рисунок 18 Общая схема работы парсера

Рассмотрим на примере. Если мы изучим разметку одной страницы, то выясним, что все вопросы и ответы обернуты в одинаковую разметку.

```

<div class="view-content">
  <div class="views-row contextual-links-region">
    <div class="question-text">
      <div class="question-text-field">
        Добрый день! Прошу отремонтировать дорогу к дому свободы
        886, заезд с улицы Ломоносова. Ямы, дыры, ненужные лежачие полицейские, улица и
        так узкая, там не разогнаться, плюс такие ямы, что на скорости 20 км в час можно
        убиться! Заранее благодарю
      </div>
    </div>
    <div class="question-answer">
      <div class="question-answer-field">
        <p>Ваше обращение по вопросу ремонта дороги вблизи
        жилых домов № 84,84а, 86а, 886 по ул. Калинина, рассмотрено администрацией Северного
        района (далее администрация района).</p>
      </div>
    </div>
  </div>
  <div class="views-row contextual-links-region">
    <div class="question-text">
      <div class="question-text-field">
        Здравствуйте, по улице Ленина 9 нет никакого освещения во дворе. Страшно и
        небезопасно ходить. Пожалуйста примите меры
      </div>
    </div>
  </div>

```



```
<div class="question-answer">
  <div class="question-answer-field">
    <p>Ваше обращение, поступившее в администрацию
города Челябинска (далее администрация района) по вопросу освещения дворовой
территории многоквартирного жилого дома № 9, рассмотрено администрацией района. В
соответствие со статьями 36, 39 Жилищного Кодекса Российской Федерации,
постановлением Правительства Российской Федерации от 13.08.2006 № 491 «Об
утверждении правил содержания общего имущества в многоквартирном доме и правил
изменения размера платы за содержание и ремонт жилого помещения в случае оказания
услуг и выполнения работ по управлению, содержанию и ремонту общего имущества в
многоквартирном доме ненадлежащего качества и (или) с перерывами, превышающими
установленную продолжительность» собственники несут бремя расходов на содержание
общего имущества в многоквартирном доме.</p>
  </div>
</div>
</div>
```

О чем нам говорит эта разметка:

- Каждый вопрос на странице обернут в тег `<div class="question-text-field">Текст вопроса</div>`
- Каждый ответ обернут в тег `<div class="question-answer-field"> Текст ответа</div>`

Наша задача состоит именно в том, чтобы разметить нужные области. Далее программа сама, по заданным ссылкам совершит обход и сохранит все вопросы и ответы с учетом нашей разметки. Исследователю остается только работать с текстом.

Программное обеспечение для парсинга сайтов

Существует большое количество специализированного программного обеспечения, которое применяется для парсинга, это могут быть отдельные скрипты, дополнения к браузерам или специальные программы.

Одно из таких решений - Scrapy. Это бесплатный фреймворк для веб-краулинга, находящийся в открытом доступе, который написан на языке программирования Python.

Scrapy — это прикладная платформа для обхода веб-сайтов и извлечения структурированных данных, которые могут быть использованы для широкого спектра полезных приложений, таких как интеллектуальный анализ данных, обработка информации. Scrapy имеет много преимуществ, лучше всего подходит для разработки сложных веб-краулеров. Потребляет меньше оперативной памяти и использует минимальные ресурсы процессора.

Как правило, дополняет Scrapy библиотека Selenium — это бесплатная и открытая библиотека для автоматизированного тестирования веб-приложений, иными словами, это инструмент для автоматизации действий веб-браузера.

Используя данную связку, можно разрабатывать гибкие парсеры для извлечения данных. Однако, несмотря на свои преимущества, Scrapy и другие фреймворки не очень дружелюбны к новичкам и требуют знания языков программирования, к примеру Python.

Так как это пособие не включает курс по Python, мы сосредоточимся на готовых программных продуктах, которые не требуют навыков программирования. Выбор зависит от вашей задачи.

В таблице, см. Таблица 4, приведены популярные программы для парсинга, которые не требуют навыков программирования, достаточно базового понимания HTML. Все программное обеспечение является платным, но, как правило, стоимость невелика, кроме того, почти все имеют демо-доступ для демонстрации возможностей.

Таблица 4. Список универсальных парсеров

Название программы, ссылка	Возможности выгрузки	Основная применимость	Дополнительные особенности
Datacol http://web-data-extractor.net/	CSV/TXT/База данных/Excel; WordPress; DLE; Webasyst; Joomla; И др.	Универсальный (есть наборы настроек для конкретных целей – интернет-магазины, соцсети, порталы объявлений).	<ul style="list-style-type: none"> • Простота использования. • Удобная расширяемость с помощью плагинов. • Готовые настройки.
Content Downloader X1 http://sbfactory.ru/	В один файл/в несколько файлов Расширения: CSV (с любыми заданными столбцами), htm, txt, php, MySQL	Универсальный - подходит для парсинга блогов, соц сетей, страниц с новостями.	<ul style="list-style-type: none"> • Возможность использовать список прокси серверов для предотвращения бана сайтов. • Возможность создания любых POST или GET Запросов (так же и JSON) для подгрузки

Название программы, ссылка	Возможности выгрузки	Основная применимость	Дополнительные особенности
			<p>дополнительных данных при парсинге контента.</p> <ul style="list-style-type: none"> • Возможность парсинга данных из больших XML файлов. • Возможность подключения CSV или XML файлов при парсинге во вкладке “Контент”. • Возможность парсить с использованием браузера (автоматизации действий пользователя в браузере.
<p>Octoparse https://www.octoparse.com/</p>	<p>Excel, JSON, HTML, базы данных</p>	<p>Универсальный (в т.ч. сканирует все типы социальных сетей, включая таких крупных игроков, как Facebook, Twitter, Instagram, YouTube, Weibo и многие другие. Быстрые и точные результаты Массовое извлечение полей</p>	<ul style="list-style-type: none"> • Ротация IP • Автоматическая ротация IP-адресов для предотвращения • блокировки IP-адресов

Название программы, ссылка	Возможности выгрузки	Основная применимость	Дополнительные особенности
		данных, включая сообщения, твиты, комментарии, репосты, лайки, хэштеги, даты, подписчиков, влиятельных лиц, ключевых лидеров мнений, URL-адреса изображений и многое другое.)	
Content Grabber https://contentgrabber.com/Manual/understanding_the_concept.htm	CSV, Excel, XML, SQL Server, MySQL, Oracle и OleDb.	Универсальный	Возможность настроить панель команд агента (автоматизация), что позволяет автоматически запускать агент в заранее определенные интервалы времени, когда вам нужно, чтобы он запускался. Это можно делать каждый час, каждый день, месяц, год и так далее.
Parsehub https://www.parsehub.com/	API CSV/ Excel Google-таблицы Tableau	Универсальный	Плюсы: Parsehub поддерживается большинством систем, в отличие от Octoparse. И еще он очень

Название программы, ссылка	Возможности выгрузки	Основная применимость	Дополнительные особенности
			гибкий, когда дело касается парсинга данных онлайн для разных нужд. Минусы: Parsehub более прост в использовании для программистов с API доступом. Бесплатная версия довольно ограничена – только 5 потоков и 200 страниц за раз. Как и Octoparse, Parsehub не поддерживает извлечение информации из PDF-документов. И некоторые продвинутые функции могут быть довольно запутанными.

Все эти программы имеют одинаковый принцип работы, на первом шаге размечается область, которую нужно сохранять, и далее, по заданным ссылкам, программа сама проходит и сохраняет данные.

Пример. Парсинг публикаций с информационного портала

Пример <http://news.egov.itmo.ru> — это новостной сайт Центра технологий электронного правительства Университета ИТМО, на нем нет API, но мы хотим выгрузить все новости за 10 лет. Все новости представлены на странице <https://news.egov.itmo.ru/news/>.

Поэтапно разберем, как нам выгрузить все новости, на примере программы Content Downloader X1 из таблицы 2.

Парсинг статей и текстов с сайтов состоит из двух этапов: сбор ссылок и парсинг контента по этим ссылкам.

Шаг 1. Нам необходимо получить ссылки на страницы с новостями. Смотрим общую выдачу, на ней 437 страниц с результатами, по 10 новостей на каждой, всего 4370 новостей. На рисунке 19 отмечена общая пагинация.

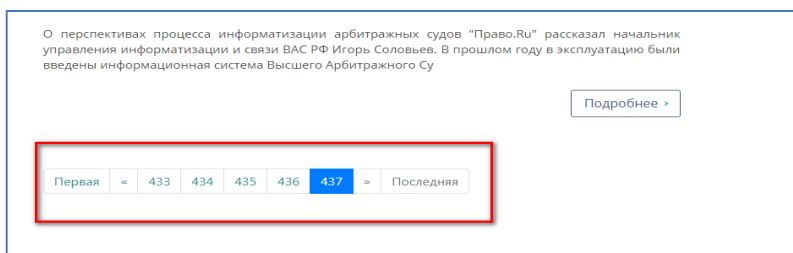


Рисунок 19 Постраничная пагинация

Каждая ссылка на свой контейнер имеет вид <https://news.egov.itmo.ru/news/?page=1>, где `?page=1` отвечает за конкретную страницу. Т.е. на первом шаге нам нужно перебрать все варианты от 1 до 437. В программе есть специальный инструмент для генерации списков ссылок с использованием заданных числовых и/или строковых значений (ключевых слов), Рис. 20.



Рисунок 20 Пример генерации ссылок

В нашем случае шаблон для генерации списка ссылок будет иметь вид:
<https://news.egov.itmo.ru/news/?page={num}>

Результатом работы этого скрипта станет список ссылок, Рис. 21:

- <https://news.egov.itmo.ru/news/?page=1>
- <https://news.egov.itmo.ru/news/?page=2>
- <https://news.egov.itmo.ru/news/?page=3>
-
- <https://news.egov.itmo.ru/news/?page=437>

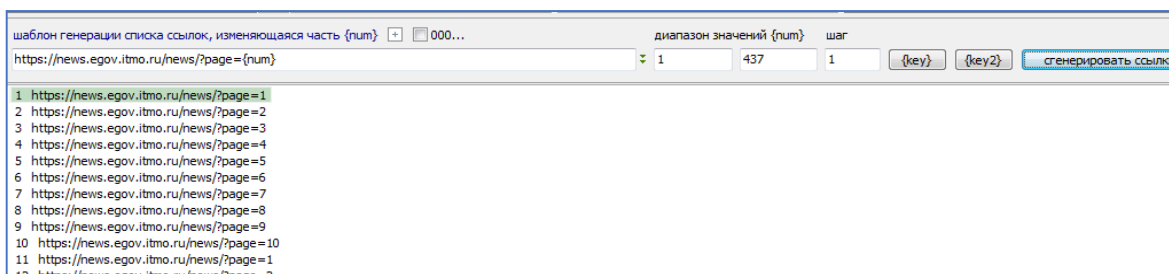


Рисунок 21 Результат генерации ссылок

Шаг 2. Теперь соберем ссылки на сами новости. Ссылки на отдельную новость имеют вид:

- https://news.egov.itmo.ru/news_01_09_07.html

-
- https://news.egov.itmo.ru/20_11_26.html

То есть в них нет сквозной нумерации, значит, их нельзя сгенерировать и нужно собирать все значения. Мы уже знаем, что страницы вида `news/?page=1` содержат в своей разметке ссылки на конечные новости.

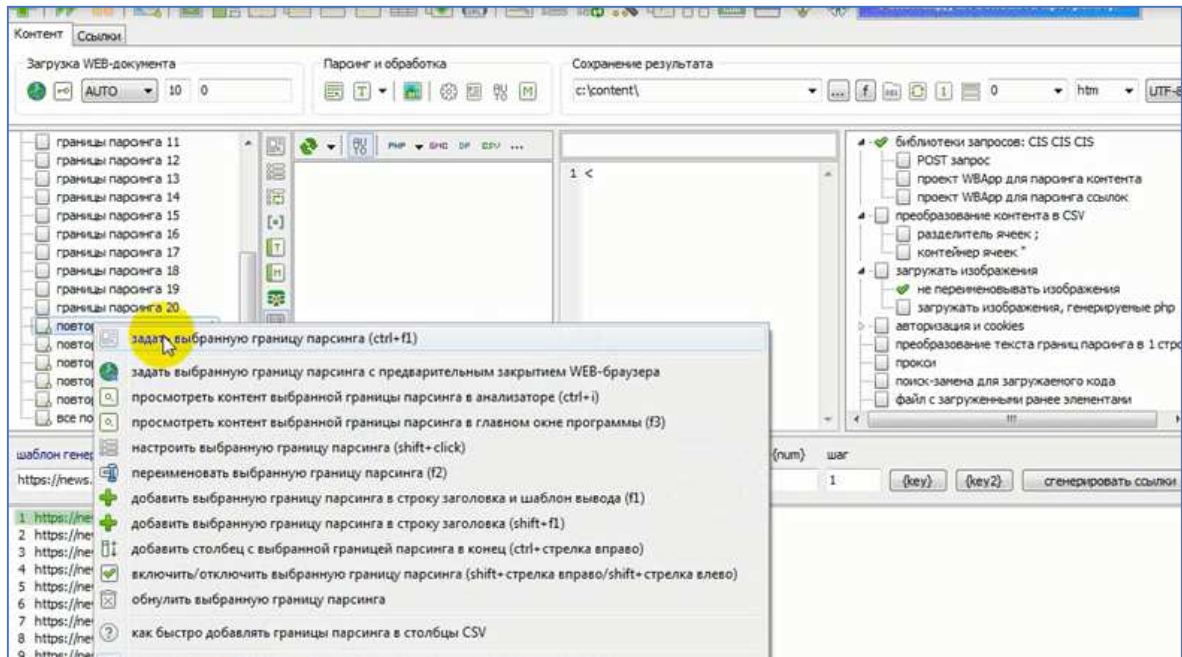


Рисунок 22 Границы парсинга

Сейчас мы разметим одну ссылку, чтобы показать системе, как на сайте хранятся ссылки на новости, и далее их и соберем. Для этого выбираем "Задать выбранную границу парсинга", Рис. 22.

В открывшемся окне мы видим две области, с разметкой и непосредственно сайтом. На сайте находим любую ссылку на новость, подводим на нее курсор и нажимаем F4, в разметке подсветится код ссылки. Пример:

```
<div class="news_box_more">
<a href="20_11_20-3.html" class="news_more">Подробнее </a>
</div>
```

Выделяем в разметке

```
<div class="news_box_more"> <a href="
```

и нажимаем - "Задать начало парсинга". Далее, выделяем "`class="news_more">`

после ссылки и нажимаем "Задать конец парсинга", Рис. 23.

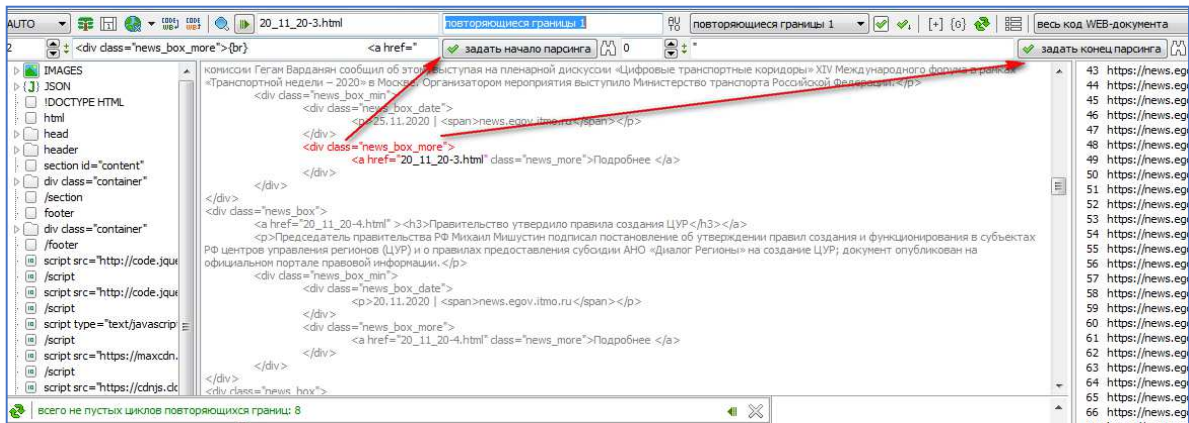


Рисунок 23 Как задать начало и конец парсинга

Готово, на примере одной ссылки мы показали системе, где хранятся адреса на новости. Теперь откроем настройки области (Рис. 24), и укажем такое значение: [https://news.egov.itmo.ru/\[VALUE\]/\[CSVLB\]](https://news.egov.itmo.ru/[VALUE]/[CSVLB])

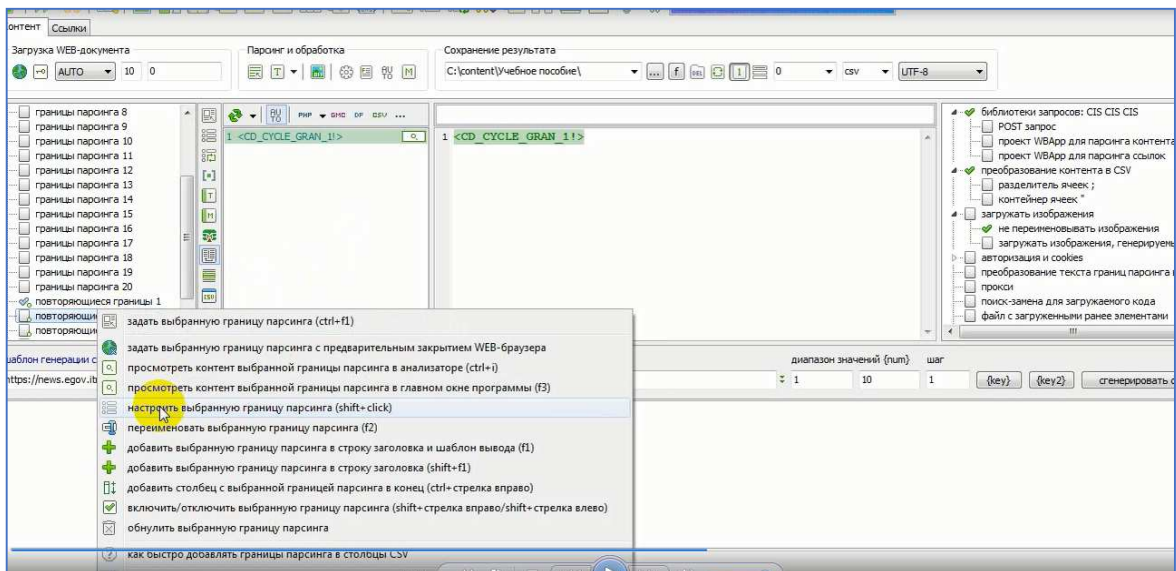


Рисунок 24 Настройка области

Это нужно сделать, так как ссылка в разметке хранится без основного домена, мы добавили к значению - адрес сайта <https://news.egov.itmo.ru/>, а в конце поставили оператор [CSVLB], чтобы каждое новое значение сохранять в новую строку (Рис. 25).

Осталось перетащить нашу границу в рабочее поле. Также перед началом парсинга нужно задать формат хранения, рекомендуемые настройки: CSV UTF-8, Рис.26.

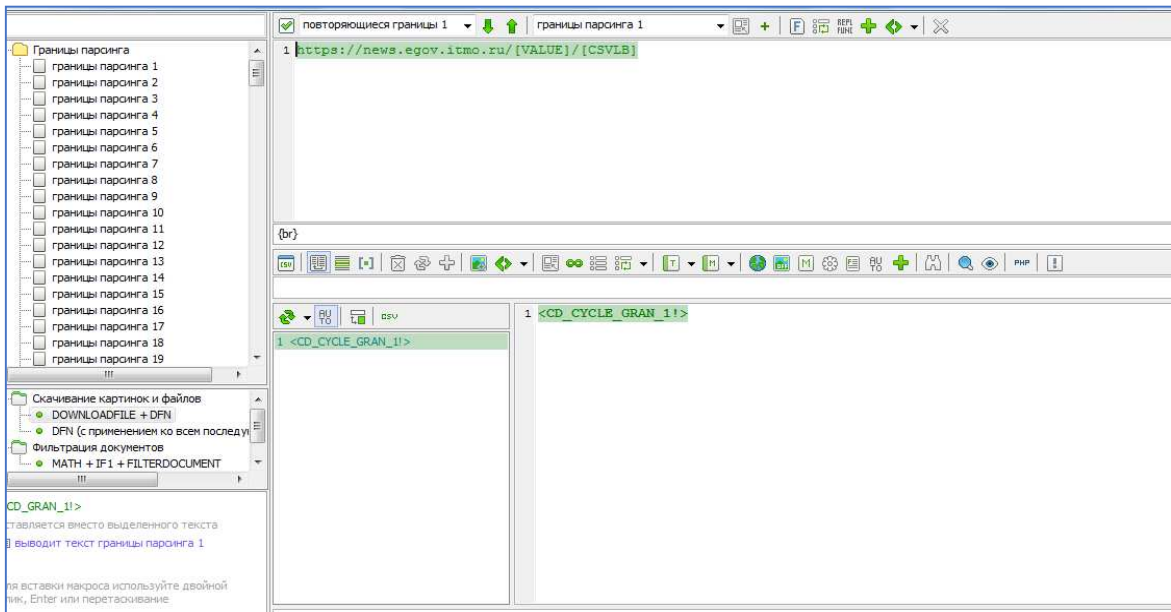


Рисунок 25 Настройка адреса

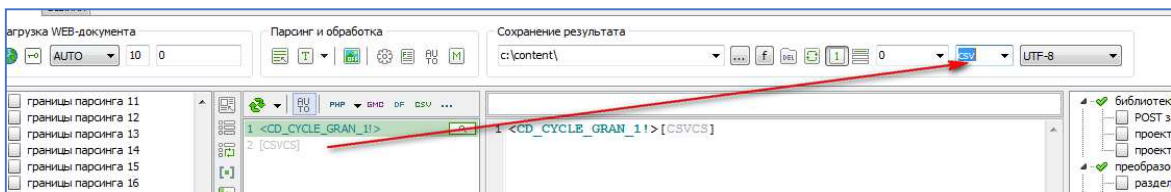


Рисунок 26 Настройка формата хранения данных

Все, программа готова к сбору ссылок новостей, все размечено, нажимаем "Старт". В результирующем файле должны собраться все ссылки на новости, вида:

- https://news.egov.itmo.ru/20_11_20-3.html/
- https://news.egov.itmo.ru/20_11_20-4.html/
- https://news.egov.itmo.ru/20_11_20-2.html/
- https://news.egov.itmo.ru/20_11_20-1.html/
-
- https://news.egov.itmo.ru/news_20_08_11-0.html/

Мы получили массив адресов с новостями, остался последний шаг – собрать сами новости.

Шаг 3. Сохраняем предыдущий проект и создаем новый. В него размещаем все наши ранее собранные ссылки и снова задаем границы парсинга, в этот раз, на примере одной новости, нам нужно показать программе, где хранится текст.

Контент одной новости имеет вид:

<h1>Правительство утвердило правила создания ЦУР</h1>

<div class="text-1">

<p>Председатель правительства РФ Михаил Мишустин подписал постановление об утверждении правил создания и функционирования в субъектах РФ центров управления регионов (ЦУР) и о правилах предоставления субсидии АНО «Диалог Регионы» на создание ЦУР; документ опубликован на официальном портале правовой информации в четверг.</p>

<p>«Создание и функционирование центров управления регионов – неотъемлемая часть цифровой трансформации регионов», – говорится в документе.</p>

20.11.2020

Как видно, заголовок обернут в тег:

<h1>

текст в тег, с классом:

<div class="text-1">

дата в тег, с классом:

Задаем начало и конец парсинга для каждого значения. И после этого, перетаскиваем все границы в рабочее поле. Задаем параметры сохранения и нажимаем "Старт". Итогом работы станут все собранные новости, Рис.27.

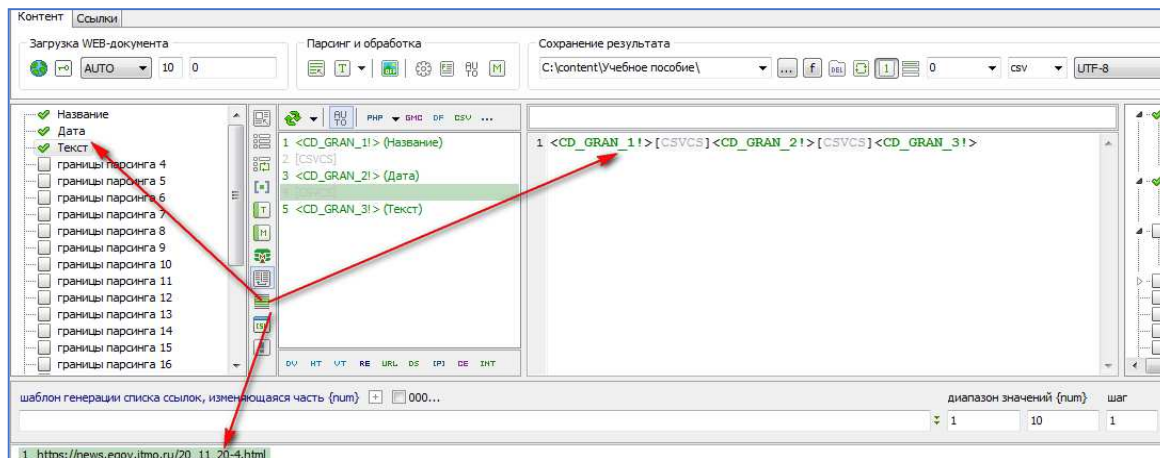


Рисунок 27 Границы парсинга

Это пример работы с программой, основная цель – показать общий принцип работы подобного программного обеспечения. Данный софт имеет большое количество продвинутых настроек – от фильтрации и замены до большого количества операторов и симуляции действий (нажатие, скрол и др.).

По ссылке <http://sbfactory.ru/> вы можете найти подробное описание и инструкции на русском языке от разработчиков программы, скачать демо версию или приобрести лицензию для работы.

Парсинг социальных сетей

Социальные сети представляют отдельный интерес для парсинга, в них собрано большое количество информации, в том числе, геопривязанной. Как правило, социальные сети имеют API для сбора основной информации, но, кроме того, есть и готовые инструменты, которые можно использовать для сбора информации без навыков программирования.

Есть большое количество готовых сервисов, которые дают возможность выгружать информацию. В таблице, см. Таблица 5, приведены примеры таких сервисов.

Таблица 5 Список универсальных парсеров для социальных сетей

№ п/п	Название программы, ссылка	Возможности парсинга
1	PEPPER.NINJA https://pepper.ninja/	Pepper парсит аудиторию Вконтакте с помощью продвинутых алгоритмов. Он может собирать пользователей по нескольким десяткам признаков: возраст, пол, геоположение, семейное положение, место учебы и работы, какую активность проявлял, интересы и многое другое.
2	VK.BARKOV.NET https://vk.barkov.net/	Набор инструментов, который позволяет доставать самые разнообразные данные из ВКонтакте и Одноклассников в удобном виде
3	CLICK.RU click.ru	Инструмент создан для сбора активной целевой аудитории в социальной сети «ВКонтакте» для запуска таргетированной рекламы. Например, при помощи инструмента вы можете собрать узкую аудиторию с общими интересами. Для сбора пользователей укажите группы по интересам и получите готовый список, который можно загрузить в рекламный кабинет «ВКонтакте» или MyTarget. Инструмент доступен всем пользователям системы совершенно бесплатно.

Ограничение на скачивание информации

При парсинге данных важно помнить о правовом моменте. Кроме того, важно понимать, что парсинг может создать нагрузку на инфраструктуру ресурса, поэтому каждый раз при сборе информации важно оценить все правовые нормы и вопросы этики. По сути, роботы поисковых систем занимаются тем же, но для них

можно сделать ограничение, и они не создают такой нагрузки. Также при парсинге важно не собирать персональные данные, это противоречит закону.

Приведем общие рекомендации при сборе информации:

- Извлекаемый контент не должен быть защищен авторским правом.
- Процесс парсинга не должен мешать работе сайта, который подвергается парсингу.
- Парсинг не должен нарушать условия использования сайта.
- Парсер не должен извлекать личную (персональную) информацию пользователя.
- Контент, который подвергается парсингу, должен отвечать стандартам правомерного использования.

2. Особенности работы с «большими данными»

2.1 Основные подходы к постановке задачи при создании многопользовательских информационно-аналитических систем в задачах обработки данных в сфере здравоохранения

Информационная система (ИС) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (см. ГОСТ 33707-2016 (ISO/IEC 2382:2015) Информационные технологии (ИТ). Словарь) [36].

ИС соединяет людей, “железо”, бюрократические процедуры, данные, прочие ресурсы и позволяет предоставить нужную информацию нужным людям в необходимый момент времени удобным способом. В качестве примера информационных систем можно привести как традиционную библиотеку, общедоступную многоязычную универсальную интернет-энциклопедию со свободным контентом Википедию, так и - счёты.

Приведем примеры классов ИС, связанных с обработкой больших данных в сфере здравоохранения и государственного управления, в качестве инструментов анализа или поставщиков данных:

- медицинские информационные системы (МИС);
- интегрированные медицинские информационные системы, к примеру, региональная МИС (РМИС) - система, интегрирующая МИС медицинских организаций (МО) всего региона;
- хранилища данных (data warehouse);
- реляционные базы данных (БД). К примеру PostgreSQL;
- БД NoSQL (not only SQL) – термин, обозначающий ряд подходов, направленных на реализацию систем управления базами данных, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL. К примеру, документо-ориентированная MongoDB.
- колоночные БД - специализированные аналитические хранилища, служащие для быстрых вычислений агрегатных выражений. Аналитические БД не подходят для отражения бизнес-логики, но отлично справляются с вычислением агрегатных выражений. Пример аналитической БД – Clickhouse [37].
- OLAP-системы;
- системы визуализации и распространения данных;
- интеллектуальная обработка данных (data mining);
- BI (Business intelligence)-системы;
- информационно-аналитические системы (далее - ИАС).

Информационные системы, как и любые социальные инструменты существуют и воспроизводятся, когда приносят пользу. В здравоохранении можно выделить две глобальные цели:

- сокращение смертности, увеличение ожидаемой продолжительности жизни;
- сокращение расходов на сбор, анализ, обработку и распространение данных.

Примеры пользы от ИАС:

1. Пример, связанный с необходимостью направления биоматериала пациентов на дополнительную диагностику в специализированную лабораторию в зависимости от диагноза и результатов лабораторных исследований из лаборатории общего профиля, Рис. 28.

Проблема: биоматериал пациентов не направляют на узкоспециализированные тесты в специализированной лаборатории

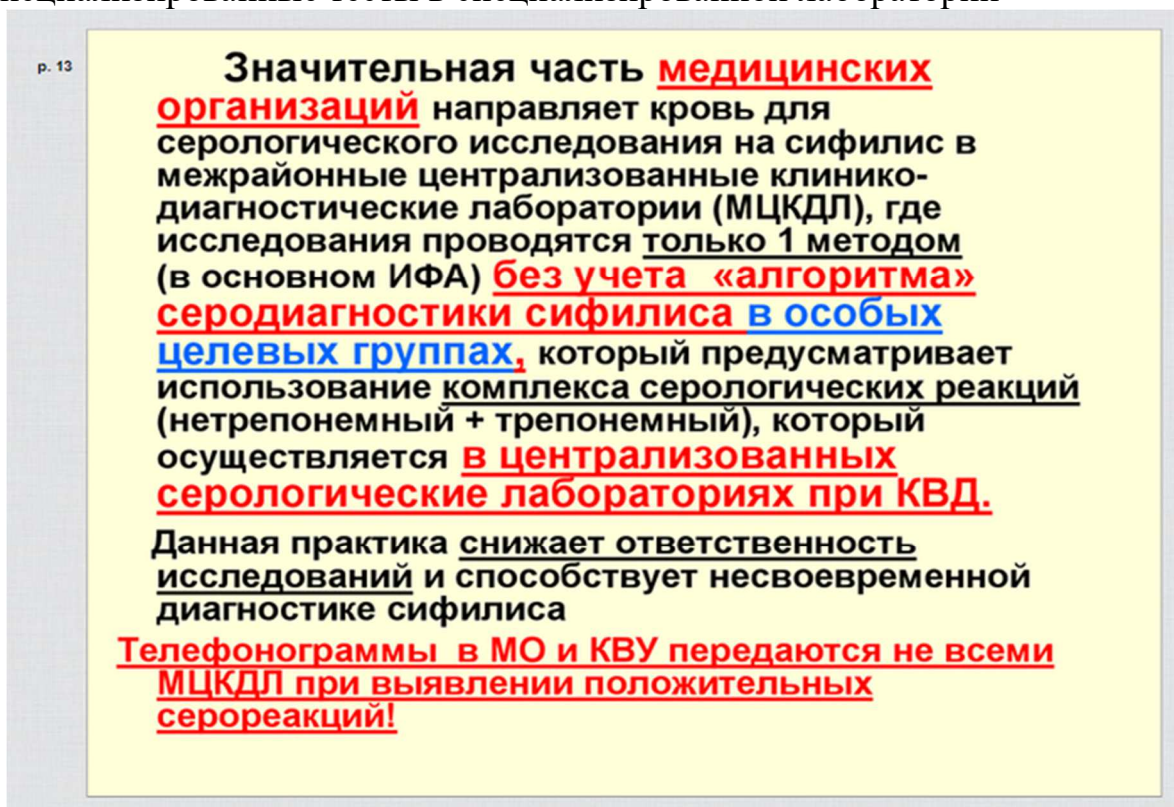


Рисунок 28 Доклад о проблемах в организации лабораторных исследований сотрудника, ведущего учёт инфекционных болезней.

Решением может быть аналитический сервис на базе ИАС анализа результатов лабораторных исследований. Необходимо составить список соответствующих лабораторных тестов, разработать организационно-распорядительный документ о порядке направления в различные типы лабораторий и реализовать в ИАС соответствующую информационную поддержку выписки направлений врачом с учетом требуемой лаборатории для данного типа

тестов, а также реализовать возможность управленческого контроля - построить отчёт с разбивкой по МО, дате исследований, уникальным пациентам, со списком некорректно проведенных лабораторных тестов.

2. Пример, связанный с оптимизацией назначений на лабораторные исследования в медицинском учреждении. Согласно Генеральному тарифному соглашению Территориального фонда ОМС Санкт-Петербурга (ТФОМС СПб), плановые лимиты внешних услуг (лабораторные исследования) по медицинским организациям на 2017 год составляли 1 745 783 284,5 рублей. В итоге работы, проведенной группой специалистов из СЗГМУ им. Мечникова по оптимизации назначений на лабораторные исследования, удалось получить сокращение издержек в учреждении порядка 20%. Этот случай оптимизации расходов учреждения описан в презентации “Интеграция медицинской информационной системы в управление назначениями (лекарственными и лабораторными)” А.С. Федоренко [38].

Если предположить, что аналогично можно оптимизировать лабораторную службу и других учреждений в Санкт-Петербурге, то выгода от оптимизации назначений на исследования может достичь 349 млн рублей в год (1,745 млрд рублей * 0,2).

Базовые инструменты составления требований заказчика к ИАС

«Практика - основа познания и критерий истины»

Руткевич М.Н.

На первом этом этапе работы необходимо выявить экспертов в предметной области, реальных пользователей Системы, правильно их опросить и согласовать результат с лицами, принимающими решения. В этом помогают базовые инструменты составления требований при опросе Заказчика, выявление реальных проблем, “боли” заказчика.

Пользовательская история

Одним из базовых инструментов составления требований заказчика является «Пользовательская история» или «рабочая история» (user story/job story) - краткая формулировка взаимодействия ИС с пользователем, решающего проблему или несущая пользу пользователю.

Пользовательскими историями удобно описывать общий функционал. Удобно общаться с Заказчиком, расставлять приоритеты, обсуждать порядок реализации с разработчиками. Однако user story - ещё не требования, их необходимо существенно детализировать для передачи в разработку.

Пример user story: КАК пользователь ХОЧУ изменить шрифт текста ДЛЯ выставления акцентов в тексте.

Пример job story: КОГДА удаляю текст ХОЧУ кликом отменить удаление ЧТОБЫ не набирать его заново.

Пример реальной практики: в Таблице 6 описан один и тот же функционал, посмотрите и ответьте на вопрос: с каким текстом легче работать на верхнем уровне, обсуждать с Заказчиком, согласовывать?

Таблица 6 Варианты описания функционала ИАС

Вариант 1	Вариант 2
1. Реализация процедур обработки дубликатов в области persistence stage. Реализация шаблонов скриптов для извлечения и агрегации инкрементальной...	2. КАК пользователь НЕ ХОЧУ дублирующих записей в таблицах ЧТОБЫ не хранить лишние данные, понятнее считать метрики. 3. КАК пользователь ХОЧУ чтобы данные обновлялись каждый день ДЛЯ предоставления ежедневных отчётов в соответствии с НПА.

К пользовательским историям можно сформулировать следующие требования:

- 1) структура пользовательской истории:
 - а) КТО, пользователь, администратор, контролёр или другая роль, или КОГДА, контекст действия (предшествующие действия, открытые меню и т.п.) - иногда удобнее использовать контекст для общего функционала,
 - б) ХОЧУ - требуемое действие, реакция, возможность действия пользователя информационной системы. Действие необходимо описывать однозначно,
 - в) ЧТОБЫ, ДЛЯ, С ЦЕЛЬЮ – польза;
- 2) простота. Пользовательские истории должны в одном-двух предложениях описывать определённый функционал. Они должны быть простыми и понятными участникам процесса: менеджеру, разработчику, функциональному заказчику, эксперту предметной области и прочим сторонам;
- 3) однозначность. Желательно не допускать двояких и размытых толкований истории. Для этого лучше проговорить историю с разными участниками;
- 4) независимость и ценность. Отдельная история должна иметь собственную, независимую ценность для пользователя. Желательно, чтобы истории были независимы друг от друга. Это позволяет просто отменить и доработать функционал story;

- 5) оцениваемость. Хорошо, когда история позволяет получить оценку трудозатрат на реализацию лучше, чем “с потолка”;
- 6) проверяемость. Воплощение истории можно проверить, протестировать и однозначно заключить, реализована она или нет.

Примеры историй

1. КАК пользователь ХОЧУ получить все информационные потоки в ситуационный центр Губернатора Ленобласти ЧТОБЫ управлять регионом.

Плохой пример, полный неоднозначности. Трудозатраты не поддаются оценке, непонятен конечный итог и реальная польза функционала.

2. КАК сотрудник, ведущий учёт исполнения поручений Губернатора, ХОЧУ автоматически получать отчёты из Системы электронного документооборота Ленобласти (СЭД) о сроках, ответственных и статусе исполнения поручений ЧТОБЫ не вести этот учёт вручную.

Пример лучше. Можно оценить требуемые компоненты решения - модули визуализации, обработки данных, интеграции с СЭД. Понятна цель, которая может быть финансово обчислена в сокращении рабочих часов отдела. Дополнительного анализа требуют отчёты и данные, которые необходимо получать.

3. КОГДА работаю со списками отчётов в ИАС ХОЧУ собирать их в папки как в проводнике ЧТОБЫ получать быстрый доступ к нужным отчётам.

4. КАК владелец данных ХОЧУ ограничивать доступ к информации сотрудников поликлиник только данными, переданными их поликлиникой, ЧТОБЫ соответствовать НПА.

5. КАК аналитик ХОЧУ видеть запрос, по которому были получены данные для отчёта, ДЛЯ отладки и проверки.

6. КАК пользователь ХОЧУ перемещать показатели (метрики) в конструкторе отчётов, кликая и зажимая ЛКМ (drag&drop), ЧТОБЫ быстрее строить отчёты.

Практическое задание

Описать в 3-10 пользовательских историях любимое приложение на телефоне или ноутбуке. Можно придумать новый функционал.

Сценарий использования

Вторым базовым инструментом составления требований заказчика является «Сценарий использования» или «Вариант использования», (англ. use case) — детальное описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды. Система может отвечать на внешние запросы Пользователя, может сама выступать инициатором взаимодействия. Сценарий использования описывает, «кто» и «что» может сделать с рассматриваемой системой, или что система может сделать с «кем» или «чем». Методика сценариев

использования применяется для выявления требований к поведению системы, известных также как пользовательские и функциональные требования. Сценарии использования рассматривают систему как «черный ящик», и взаимодействия с системой, включая системные ответы, описываются с точки зрения внешнего наблюдателя. Это преднамеренная политика, потому что это вынуждает автора сосредоточиться на том, что система должна сделать, а не как это должно быть сделано, и позволяет избегать создания предположений о том, как функциональные возможности будут реализованы [36].

Шаблон сценария использования [38]

- 1) название сценария;
- 2) действующие лица: пользователи, внешние ИС являющиеся активаторами (триггером) работы всего сценария или его частей;
- 3) заинтересованные стороны: к примеру, отраслевые эксперты, руководители, принимающие ИС, ответственные лица за приёмку или эксплуатацию системы;
- 4) цели;
- 5) предусловия;
- 6) термины и сокращения;
- 7) начало сценария;
- 8) успешный сценарий;
- 9) результат;
- 10) расширения;
- 11) примечания.

Пример сценария использования «Drag&Drop показателей (метрик)»

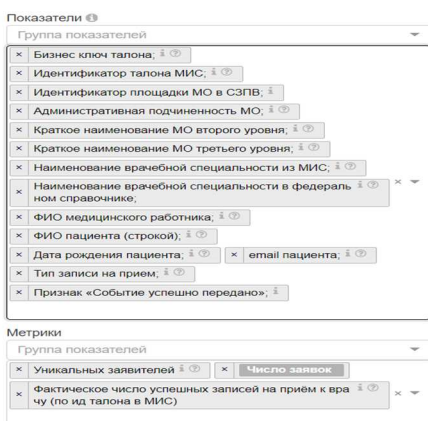
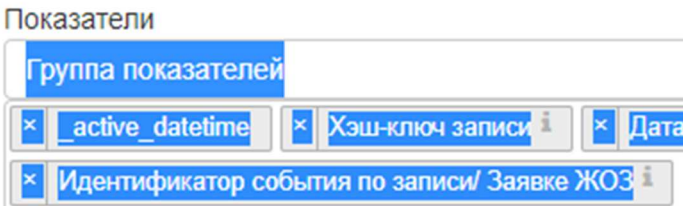


Рисунок 29 Показатели и метрики в пользовательском интерфейсе superset

На рисунке 29 приведен пользовательский интерфейс выбора показателей и метрик, в Таблице 7 – соответствующий сценарий.

User story: КАК пользователь ХОЧУ перемещать показатели (метрики) кликая и зажимая левую клавишу мыши (drag&drop) ЧТОБЫ быстрее строить отчёты.

Таблица 7 Сценарий использования Drag&Drop показателей (метрик)

Раздел сценария	Содержание
Действующие лица	Пользователь, Superset (Система)
Цели	Переместить показатель (метрику)
Предусловие	1. Пользователь находится в конструкторе отчётов 2. В отчёте есть хоть один показатель (метрика)
Термины и сокращения	ЛКМ - левая кнопка мыши. Область показателей (метрик) - скруглённый прямоугольник внутри конструктора отчётов, где могут находиться 1 или несколько показателей (метрик). Может быть несколько областей показателей (метрик). К примеру, для Гистограммы. Пиктограмма показателя (метрики) - скруглённый прямоугольник вокруг названия показателя (метрики). В пределах области - подразумевается, что курсор внутри визуальной границы области.
Успешный сценарий:	<p>1. Пользователь открывает или создаёт отчёт. 2. Пользователь кликает по пиктограмме показателя (метрики), зажимает ЛКМ и перемещает курсор. 3. Система меняет прозрачность прямоугольника и перемещает его вслед за курсором. Система не выделяет текст при перемещении курсора.</p>  <p>4. Пользователь отпустил ЛКМ с пиктограммой показателя в пределах области показателей (той же или другой). Пользователь отпустил ЛКМ с пиктограммой метрики в пределах области метрики.</p>
Результат	Показатель (метрика) X добавлен в область показателей (метрик) в порядке, определяемым местом нахождения

Раздел сценария	Содержание
	<p>курсора по отношению к другим пиктограммам показателей (метрик).</p> <p>Левым показателем (метрикой) L для курсора назовём показатель (метрику), который</p> <ul style="list-style-type: none"> • в одной строке с курсором, • его центр левее курсора, • если левый показатель не определяется по правилам выше, левым показателем назначается последний показатель в строке выше. • Назовём L_i - индекс показателя (метрики) по порядку следования в списке области показателей (метрик). • Если левого показателя (метрики) нет (т.е. нет показателя левее курсора, нет показателей в строках выше курсора) - перетаскиваемый показатель становится первым по порядку следования. Если есть - индекс показателя (метрики) по порядку следования в области становится L_{i+1} (становится за левым показателем).
Расширения:	
2а	Пользователь попал по крестику в прямоугольнике с названием показателя (метрики). Работает стандартный сценарий удаления показателя (метрики).
4а	Пользователь отпустил ЛКМ за пределами любой области показателей (метрик). Результат: отчёт не меняется, пиктограмма показателя (метрики) возвращается на предыдущее положение.
4б	Пользователь отпустил метрику в области показателей. Результат: отчёт не меняется, пиктограмма метрики возвращается на предыдущее положение.
4в	Пользователь отпустил показатель в области метрик. Результат: отчёт не меняется, пиктограмма метрики возвращается на предыдущее положение.
4г	Показатель отпустил показатель X в области показателей, в этой области показателей такой показатель X уже есть. Результат: показатель добавлен в область показателей (метрик) в порядке, определяемым местом нахождения курсора по отношению к другим пиктограммам показателей

Раздел сценария	Содержание
	(метрик). Дублирующий показатель удалён из области показателей.
Примечание	Перемещение показателей (метрик) сохраняется в рамках работы с конструктором отчёта. Перемещение показателя (метрики) НЕ приводит к сохранению отчёта.

Пример некачественного описания функционала

Необходимо иметь возможность “провалиться” (drill-down) в данные в отчётах ИАС.

Drill-down Отчёт 1 - Отчёт 2 - возможность перехода (и возврата обратно) с Отчёта 1 на Отчёт 2 по заданным правилам.

Ограничения точки перехода - набор ограничений, который соответствует тому уникальному сектору диаграммы, который кликает пользователь.

1) Пусть на Отчёт 2 выставлены [Ограничения Отчёта 2].

2) Пусть пользователь кликает Отчёт 1.

3) Тогда Drill-down Отчёт 1 - Отчёт 2 должен строить Отчёт 2 со следующими ограничениями:

a) [Ограничения Отчёта 2]

b) [Ограничения точки перехода]

Заключение разработчика.

Ничего не понял. Что входит в ограничения точки перехода? Где это отображается в системе? Можно как-то подробнее описать и структурировать? Что там вот так-то вызывается меню drill-down, так-то она выполняется, так-то контекст передается и формируется.

Что такое "Ограничения Отчёта"? Как они соединяются с ограничениями точки перехода? У объекта "Отчёт" нет ограничений. Есть временная шкала, фильтры, sql ограничения where.

Надо все формализовывать. Вообще сейчас читаю, и мне кажется, что надо документ переформатировать, а то какая-то каша.

Практическое задание

Открыть телефон или ноутбук, выбрать любимое приложение, описать 1-2 сценария использования. Каждый сценарий должен занимать не менее 1 страницы 12 шрифтом с одинарным отступом.

Придумать свой функционал. Описать его в виде user story и сценария использования.

Скриншоты

Третий базовый инструмент составления требований заказчика, который мы рассмотрим, – снимки экранов пользовательского интерфейса или «скриншоты». Главное в пользовательских историях, сценариях использования и других требованиях – достичь взаимопонимания участников процесса. Если это понятно всем, если это просто - используй это. Множество требований, тесткейсов, особенно к ui/ux, удобно описать простыми скриншотами, см. Рис. 30.



Рисунок 30 Скриншот, описывающий требования к интерфейсу

Специфичные инструменты для ИАС и ВІ

При работе с ИАС или ВІ можно выделить часто возникающие требования и, как следствие, соответствующие им инструменты:

1) Паспорт информационно-аналитического сервиса. Составляется как базовая памятка по основополагающим вопросам:

- a) Откуда брать данные?
- b) Как получать данные?
 - i. Нормативно-правовые вопросы. Кому принадлежат данные? На каком основании можно получать данные?
 - ii. Технический способ получения данных. К примеру, получение данных по API или напрямую из СУБД.
 - iii. Каналы связи для получения данных. Доступ к каналам связи. К примеру, доступ в защищённые корпоративные или государственные сети требует отдельных работ.
- c) Какие отчёты нужны?
 - i. Постановка задачи на отчёт. Какие данные с какими складывать, что учитывать, что - нет и т.д. Постановку задачи следует обсуждать с экспертами, практиками предметной области.
 - ii. Кто потребитель аналитических отчётов?

2) Макеты визуальных отчётов, наборов отчётов (инфопанелей, дашбордов). Красивая картинка стоит половины презентации – это лучшая user story по ВІ для руководителя.

3) Тестовый набор данных. Стоит получить тестовую выгрузку данных или доступ к тестовой БД, построить в любых электронных таблицах примеры отчётов.

Примеры

Пример на внешних носителях 1. Паспорт цифрового сервиса
Ситуационного Центра Губернатора

Пример на внешних носителях 2. Жизненный цикл цифрового сервиса
Ситуационного Центра Губернатора

Пример на внешних носителях 3. Копилка идей для Ситуационного
Центра Губернатора

Макет Аналитической панели по мониторингу карт маршрутизации пациентов с подозрением на злокачественные новообразования приведен на Рис. 31.

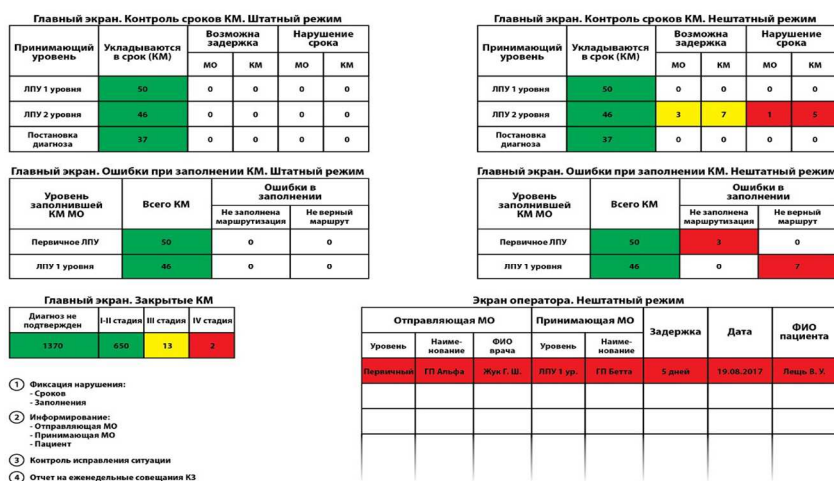


Рисунок 31 Макет Аналитической панели по мониторингу карт маршрутизации пациентов с подозрением на злокачественные новообразования

Анализ источника данных

Пусть описаны требования Заказчика, составлены сценарии использования, паспорт аналитического сервиса, нарисованы визуальные информационные панели, достигнута договорённость о терминологической и нормативно-правовой базе, определён источник данных. Следующий шаг - детализированный анализ источника данных.

Во-первых, необходимо определить состав данных, которые необходимо получать, описать обработку данных:

- требуется получить формализованное описание структуры БД;
- требуется получить доступ к БД, её бэкапу или тестовой БД;
- желательно получить описание бизнес-процессов использования данных. Документ понадобится при подготовке отчётов для Заказчика, поиска проблем в составленных отчётах, анализа и определения нештатных ситуаций;
- базовые вопросы, которые необходимо выяснить у специалистов заказчика:
 - Что за данные находятся в ИС источнике данных?
 - Какой предполагаемый объём данных?
 - Могут ли они обновляться или только дополняются?
 - Какой предполагаемый объём при обновлении данных?
 - Есть ли в изменяющихся данных даты обновления?
 - Какие идентификаторы есть в системе? Где они хранятся?
 - Есть ли справочные значения?
 - Где хранятся справочники?
 - Есть ли данные, связанные с другими ИС?
- специфичные вопросы, которые необходимо выяснить у специалистов заказчика:
 - К каким данным есть доступ? Какие требования к информационной безопасности применяются?
 - Есть ли персональные данные или другая информация особой важности?
 - Нужно ли загружать накопленные данные, существовавшие до запуска аналитической системы? На источнике данные могут быть накоплены до запуска ВІ. Требуется отдельно анализировать историческую загрузку данных, их структуру, справочники и версии.

Во-вторых, необходимо определить технический способ получения данных путём переговоров с разработчиками и операторами ИС источника данных в зависимости от требований.

- Какие каналы связи для получения данных?
- Какие требования к информационной безопасности?
- Как получить данные для нулевой (первичной) загрузки данных в ИАС?
- Как получать обновления в данных за заданный период?
- Как получать справочные значения для кодифицированных данных с учётом версионности справочников, если такая ведётся?
- Наиболее распространённые способы получения данных:
 - API, Web API. ИС источник данных формирует и поддерживает на своей стороне программный интерфейс с методами получения данных (API), удовлетворяющих требованиям, описывает и документирует запросы. К примеру, <http://api.n3zdrav.ru/api/> [39],

<https://yandex.ru/dev/metrika/doc/api2/concept/about.html>, <https://api-fns.ru/>;

- доступ к БД (таблицам или представлениям в БД) на получение информации.
- Специфичные, полуручные способы. К примеру, выгрузка раз в период данных из ИС источника данных файлов в формате CSV или JSON в определённую директорию с согласованными названиями, включающими даты выгрузки и т.п.

В-третьих, необходимо определить способ и регулярность обновления данных – к примеру, ежедневно, ежемесячно, в режиме реального времени.

Полученные данные необходимо зафиксировать и использовать для проектирования аналитического решения на следующем этапе.

Создание терминологической базы аналитического решения и изучение нормативно-правовой базы

Обязательным условием при составлении требований является определение терминологической и нормативно-правовой базы, достаточной для однозначного понимания составляемых требований в практическом применении всеми участниками.

Как посчитать число проведённых лабораторных исследований? Для начала нужно узнать, что такое лабораторное исследование. Есть несколько разных определений:

- Лабораторное исследование - исполненная заявка на лабораторное исследование по пациенту, переданная от врача в лабораторию.
- Лабораторное исследование - определённая услуга класса «лабораторные исследования» по номенклатуре медицинских услуг, утверждённой Минздравом России.
- Лабораторное исследование - специфичный тест, произведённый на лабораторном оборудовании определённым методом некоторого биоматериала по справочнику ЛАТЕУС.
- Лабораторное исследование - единица медицинской услуги, оплачиваемая ТФОМС в соответствии с Генеральным тарифным соглашением и Территориальной программой государственных гарантий бесплатного оказания гражданам медицинской помощи.

Разные определения, разные справочники формируют различные результаты, сравнение которых похоже на сравнение числа яблок с числом ящиков на складе в магазине.

По общению с врачами, клиническим стандартам и рекомендациям, есть ряд основных разделов в лечении пациента, которые необходимо четко определить и применять:

1. Общая информация о СМО: дата поступления пациента, дата выписки, состояние при поступлении, результат лечения и т.д.
2. Данные о медицинской организации, к примеру, профиль, тип, число коек. Важный справочник – Федеральный реестр медицинских организаций.
3. Данные о медицинских работниках, к примеру, лечащем враче, врачах, участвовавших в лечении. Важный справочник – Федеральный реестр медицинских работников.
4. Диагноз пациента. К примеру, информация о дате установки, клиническая формулировка диагноза, основное или сопутствующее заболевание, предварительный или заключительный диагноз. Важный справочник – Справочник диагнозов. Международная классификация болезней (МКБ).
5. Диагностика. Лабораторные исследования, инструментальные исследования, в т.ч. с применением обработки изображений. Важные справочники:
 - a. Справочник лабораторных исследований.
 - i. Федеральный справочник лабораторных исследований (ФСЛИ).
 - ii. Международный справочник LOINC.
 - iii. ЛАТЕУС (Справочник лабораторных исследований, используемый в Санкт-Петербурге).
 - b. Номенклатура медицинских услуг.
6. Лечение. Лекарства и услуги, которые оказывались пациенту.
 - a. Справочник лекарственных средств. При работе с лекарствами важно разделить торговое наименование и Международное непатентованное наименование (МНН). К примеру,
 - i. Аспирин-С. Торговое наименование.
 - ii. Препарат состоит из ацетилсалициловой и аскорбиновой кислоты (МНН).
 - b. Номенклатура медицинских услуг.
7. Наблюдение и профилактика. Различные регистры пациентов – хронических сердечников, сахарных диабетиков и других пациентов.
8. Оплата лечения. Сколько заплатили, за какую услугу, как произведена оплата. Базовыми видами оплаты являются: подушевой норматив (метод оплаты услуг амбулаторно-поликлинического звена за каждого прикрепленного человека) по ОМС, оплата за законченный случай лечения по ОМС, оплата лечения по ДМС, платные услуги. Важные документы и справочники:
 - a. Номенклатура медицинских услуг.
 - b. Территориальная программа государственных гарантий.
 - c. Генеральное тарифное соглашение.

С одним из примеров описания стандартов медицинской помощи можно познакомиться на свободно доступной медицинской базе знаний «Энцикломедия», которую ведет портал российского врача «Медвестник»: Стандарт лечения при отравлении (<https://bz.medvestnik.ru/standarts/1375n.html/about>).

В качестве полезных ресурсов для ознакомления с терминологией здравоохранения можно рекомендовать следующие:

- Центральный научно-исследовательский институт организации и информатизации здравоохранения» Министерства здравоохранения Российской Федерации (<https://mednet.ru/>).
- Справочники Минздрава России (<https://nsi.rosminzdrav.ru/#!/refbook>).
- Справочник лекарственных средств (<https://esklp.egisz.rosminzdrav.ru/>).
- Справочник SNOMED CT (http://www.ihtsdo.ru/snomed_ct/). Систематизированная медицинская номенклатура, подходящая для обработки машинными методами.
- Медицинская база знаний (<https://bz.medvestnik.ru/>).
- Стандарт обмена медицинской информацией FHIR (<http://fhir.ru/>).

Примеры информационно-логических моделей ИАС

Пример на внешних носителях 4. Информационно-логическая модель витрины данных ОДЛИ;

Пример на внешних носителях 5. Информационно-логическая модель витрины данных ИЭМК.

Также важным этапом при создании аналитического решения является изучение нормативно-правовой базы.

Рассмотрим ситуацию. Создан сервис, пользующийся спросом у населения за свою простоту и удобство. Сервис внедрён и интегрирован с различными смежными информационными системами, приносит ощутимый, практичный результат, но сервис закрывается по причине несоблюдения законодательства, к примеру, в части ФЗ "О персональных данных" от 27.07.2006 N 152-ФЗ или использования медицинской информации без явного согласия пациента.

Это – следствие ошибки при проектировании сервиса. Обязательным требованием при разработке ИС является работа с нормативно-правовыми актами, регулирующим сферу деятельности ИАС.

Приведем ряд базовых нормативно-правовых актов в сфере здравоохранения и информатизации здравоохранения РФ, с которыми следует ознакомиться при работе с медицинской информацией [40, 41]:

1. Федеральный закон "Об основах охраны здоровья граждан в Российской Федерации" от 21.11.2011 N 323-ФЗ.
2. Федеральный закон "Об обязательном медицинском страховании в Российской Федерации" от 29.11.2010 N 326-ФЗ.
3. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ.

4. Федеральный закон от 29 июля 2017 г. N 242-ФЗ "О внесении изменений в отдельные законодательные акты Российской Федерации по вопросам применения информационных технологий в сфере охраны здоровья". Основной закон по информатизации здравоохранения.
5. Приказ Министерства здравоохранения РФ от 24 декабря 2018 г. N 911н "Об утверждении Требований к государственным информационным системам в сфере здравоохранения субъектов Российской Федерации, медицинским информационным системам медицинских организаций и информационным системам фармацевтических организаций".
6. Постановление Правительства РФ от 12 апреля 2018 г. № 447 "Об утверждении Правил взаимодействия иных информационных систем, предназначенных для сбора, хранения, обработки и предоставления информации, касающейся деятельности медицинских организаций и предоставляемых ими услуг, с информационными системами в сфере здравоохранения и медицинскими организациями".
7. Постановление Правительства РФ от 5 мая 2018 г. № 555 "О единой государственной информационной системе в сфере здравоохранения" [42].
8. Региональные НПА (например, Распоряжение Комитета по Здравоохранению Санкт-Петербурга 88-р от 21.02.2018 [40]. Установление формальной, нормативной структуры ЭМК петербуржца. По распоряжению в 2018 были запущены рейтинги МО по подключению к РЕГИЗ и передаче данных в ЭМК, <https://spbmiac.ru/ehlektronnoe-zdravookhranenie/rejtingi-e-zdravookhraneniya/rejtingi-mo-emk-peterburzhca/>).

2.2 Проектирование витрин данных аналитического решения в колоночных СУБД

В настоящем разделе будут рассмотрены подходы по проектированию витрин данных аналитического решения в широко применяемых колоночных СУБД.

При проектировании аналитического решения требуется регулярное общение с командой разработки: программистами, архитекторами, системными администраторами.

При выборе аналитического решения следует отталкиваться от требований заказчика, доступных ресурсов (имеющихся в наличии у заказчика аналитических систем, ресурса на разработку, серверных мощностей и т.п.), способа получения данных.

Базовые виды аналитических решений

Можно привести различные примеры вендоров информационно-аналитических систем:

- Power BI — комплексное программное обеспечение бизнес-анализа компании Microsoft;
- Tableau — американская компания, разработчик одноимённого программного обеспечения для интерактивной визуализации данных и бизнес-аналитики;
- QlikView — компания-разработчик программного обеспечения для систем BI и программная BI-платформа с ассоциативным поиском в оперативной памяти со встроенными средствами ETL.

Примеры реализаций ИАС, обрабатывающих большие данные в России:

- Яндекс метрика (<https://metrika.yandex.ru/dashboard?id=29761725>) стоит на миллионах веб-сайтов. Сотни тысяч аналитиков каждый день сидят и смотрят в интерфейс «Метрики», запрашивают какие-то отчёты, выбирают фильтры и пытаются понять, что происходит у них на сайте, купил ли этот человек этот холодильник, или что происходит. Десятки миллиардов событий принимаются каждый день [37].
- Криста BI (<https://www.krista.ru/catalog/platformkbi/>) - ИАС в сфере государственного управления:
 - <http://budget.gov.ru/>.
 - <http://datamarts.roskazna.ru/>.
- Нетрика BI (<https://netrika.ru/solution/bi>) - победитель конкурса «Лучшее ИТ решение для здравоохранения 2020». Включает проблемно-ориентированные витрины данных по различным подсистемам. К примеру, более 500 млн результатов лабораторных исследований доступны для анализа, исходя из [43].

Можно выделить следующие виды аналитических решений:

- Составление фиксированных отчётов, с обращением к первичной базе, её копии, выгрузки из БД. Плюсами данного вида решения является быстрая реализация, минусами – то, что пользователь получит только определённый, фиксированный отчёт без возможности детализации, фильтрации, часто с ограничениями по визуализации, отсутствие преаналитической обработки данных или его ручная реализация: расшифровка справочных значений, удаление спецсимволов, подсчёт возраста по дате рождения и т.п.
- Составление многомерных кубов данных, обогащение возможностей средствами визуализации, ролевого доступа и т.п. По своей сути кубы напоминают сводные таблицы. Традиционные схемы хранения данных для кубов состоят из таблиц фактов и измерений: звезда, снежинка. В качестве примера можно привести многомерный куб «Число записей на приём к врачу», в котором факты – это записи на приём к врачу, а измерения - дата, пациент, врач, тип записи, медицинская организация. В качестве плюсов решения можно отметить то, что все данные по аналитическому направлению сосредоточены в одном месте. Решение позволяет проводить более глубокую и разнообразную аналитику в сравнении с фиксированными отчётами, оперировать данными без привлечения программистов, при должной визуализации обеспечивает простой доступ к данным. Минусами решения является сложность построения произвольных отчётов. Многомерный куб - программное решение, требующее технических и кадровых ресурсов [44].
- Составление проблемно-ориентированных витрин данных в виде плоских таблиц в аналитических хранилищах. Схемы хранения данных: звезда (расширение - снежинка), Data vault (хабы, линки, сателлиты) [45]. В качестве примеров использования можно привести Нетрика VI, Яндекс метрика – где сохраняется небольшое количество очень широких таблиц исходных событий. Например, в случае Яндекс метрики — это просмотры страниц. Одна строчка означает один просмотр страницы, у одного просмотра страницы очень много разных атрибутов: пол человека, возраст, куплен холодильник или нет, и еще какие-то колонки. В нашем случае в просмотрах мы записываем более 500 разных атрибутов для каждого просмотра.” [37]. Плюсы решения – позволяет работать с первичными данными, дополненными необходимой информацией, позволяет строить отчёты, в зависимости от модуля визуализации, в различных срезах без привлечения программистов. Минусы решения - сложность, затратность проектирования и реализации решения: высокие затраты времени аналитиков и разработчиков,

серверных мощностей. Сложность тестирования и поддержания качества данных.

Особенности проектирования плоских витрин больших данных

Первой особенностью проектирования плоских витрин больших данных является размножение данных. К примеру, лечат пациента с диагнозами Ds1, Ds2, Ds3. Пациенту выписали лекарства mnn1, mnn2, mnn3, mnn4, сделали ряд услуг serv1, serv2. Если расположить эти данные в одну плоскую таблицу, то результирующее число строк равно их декартовому произведению. В примере получим 24 строки. При работе с сотнями миллионов строк и десятками несвязанных строго параметров это становится большой проблемой. Методы работы с размножением данных:

1. Выявление. Для начала необходимо выявить размножающие данные. Один из удобных методов - превратить описание источника данных и проектируемой витрины в ER диаграммы и проанализировать.
2. Оценка. После нахождения логического размножения требуется оценить реальный масштаб проблемы исходя из фактических данных в источнике. Часто есть ряд ресурсов, которые необязательны к передаче (имеют кратность 0..*). С помощью запросов к источнику данных необходимо оценить частоту наполнения данными и превышения кратности в единицу, а затем провести встречу с разработчиком и системным администратором для практической оценки проблемы.
3. Примеры вариантов устранения множащих показателей:
 - a. Разнести показатели по разным витринам данных. Для получения отчётов использовать сложные SQL запросы.
 - b. Разнести по разным столбцам. Диагнозы в ОДИИ (пример ниже).
 - c. Выбрать первое (последнее, любое) из значений. Название устройства в ОДИИ (пример ниже).
 - d. Совокупность вышеперечисленных методов. Можно придумать множество различных способов обойти или устранить такую проблему. Важно точно определить три вещи: требования к отчётам и витрине данных со стороны Заказчика, доступные вычислительные мощности, фактический масштаб проблемы. Исходя из постановки задачи искать решение.

В качестве второй особенности проектирования плоских витрин больших данных следует выделить необходимость автоматического тестирования качества данных. 100 обработанных строк можно проверить вручную. 100 миллионов строк по 7463 показателям (столбцам) - нет. Для проверки необходимо продумывать автоматические методы проверки данных. Подробнее вопрос изложен в главах ниже.

Третьей особенностью проектирования плоских витрин больших данных является необходимость управления изменениями. Рано или поздно потребуется дополнить или изменить данные в витрине. К примеру, система-источник данных начнёт отправлять данные в федеральный сервис или перейдёт на новый справочник. Возможно, Заказчику понадобится информация по лабораторному оборудованию в составе витрины данных лабораторных исследований. На что следует обратить внимание:

1. Показатели витрины в документации необходимо дополнять информацией о дате введения показателя.
2. При возможных изменениях в источнике данных следует указывать версию и дату опорной документации - описания структуры БД, API и другой.
3. При смене справочников необходимо отслеживать ретроспективные изменения данных на источнике. Пусть 1 декабря произойдет изменение данных или структуры источника - будут ли меняться данные за 25 ноября? Какие? С какого числа? Как? Как программно определить изменения, версию системы? Изменения необходимо заложить в работу разработчика и системного администратора. Объём работ может быть нетривиальным.

ETL - процесс извлечения, преобразования и загрузки данных

До получения данных в витрины или кубы их необходимо обработать: убрать лишние точки, пробелы, расшифровать коды наименованиями и т.п. Этот процесс называется ETL (от англ. Extract, Transform, Load — дословно «извлечение, преобразование, загрузка») — один из основных процессов в управлении хранилищами данных, который включает в себя [36]:

- извлечение данных из внешних источников;
- их трансформацию и очистку, чтобы они соответствовали потребностям бизнес-модели;
- загрузку их в хранилище данных.

Документирование информационно-аналитического решения

Для документирования выбранного решения и постановки задачи на работу удобно использовать Проект витрины данных, ER диаграммы.

Основные части Проекта витрины данных:

- описание поля в витрине - англ. название, русское, дата создания и т.п.;
- описание обработки данных для получения поля из источника, указание справочников, если используются и т.п.;
- описание источника данных - БД, схема, таблица, поле, тип поля, может ли быть пустым, является ли идентификатором и т.п.

С основами ER (entity-relationship) диаграмм можно ознакомиться в примерах проектов витрин данных, приведенных ниже, где используется стиль нотации «воронья лапка». Согласно данной нотации, сущность изображается в виде прямоугольника или другого объекта, содержащего её имя, выражаемое существительным. Имя сущности должно быть уникальным в рамках одной модели. При этом имя сущности — это имя типа, а не конкретного экземпляра данного типа. Экземпляром сущности называется конкретный представитель данной сущности.

Связь изображается линией, которая связывает две сущности, участвующие в отношении. Степень конца связи указывается графически, множественность связи изображается в виде «вилки» на конце связи, см. Рис.32 [46].

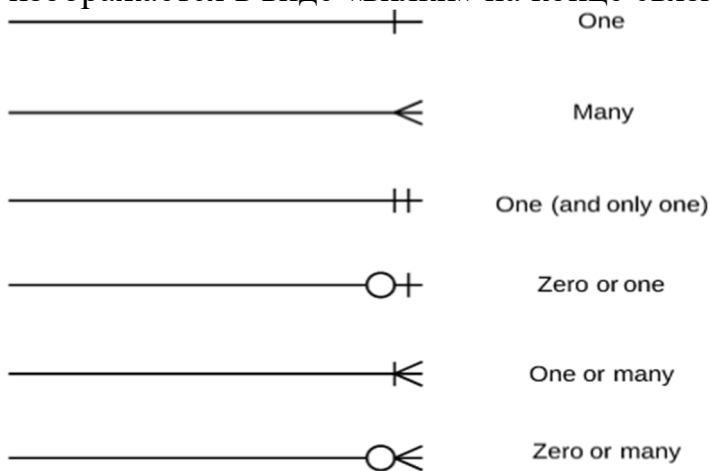


Рисунок 32 Обозначения нотации «воронья лапка» ER диаграмм

Примеры проектов витрин данных

Пример на внешних носителях 6. API Учёт движения коечного фонда.

Пример на внешних носителях 7. Витрина данных Учёт движения коечного фонда.

Пример на внешних носителях 8. API Обмена инструментальными исследованиями.

Пример на внешних носителях 9. Часть проекта витрины данных ОДИИ.

Пример на внешних носителях 10. ER диаграмма сборки и анализа на размножение данных витрине ОДИИ.

Практическое задание

Выберите любую подсистему <http://api.n3zdrav.ru/api/> [39], подготовьте проект проблемно-ориентированной витрины данных, включающий в себя не менее 3 различных ресурсов, методов, контейнеров или таблиц.

Проблемы, которые можно выбрать:

- Учёт числа больных сердечно-сосудистыми заболеваниями.
- Учёт числа больных злокачественными новообразованиями.

- Мониторинг передачи данных от МИС ЛПУ в подсистему-источник данных.
- Соблюдение сроков проведения лабораторных исследований. Согласно ТППГ в амбулаторном звене - 14 дней. 7 дней на выявление ЗНО.
- Соблюдение сроков проведения инструментальных исследований. Согласно ТППГ в амбулаторном звене - 14 дней. 7 дней на выявление ЗНО.
- Мониторинг записи на приём к врачу
- Сроки проведения госпитализации по направлению

Допустимо найти любую другую проблему, решение которой направлено на глобальные цели. Необходимо обосновать свой выбор.

2.3 Использование витрин данных для настройки и визуализации отчётов

После подготовки удобного источника данных, требуется непосредственно формировать отчёты. Есть 2 технические части при формировании отчёта - получить данные и визуализировать.

Большинство модулей визуализации, BI, ИАС в том или ином виде содержат конструктор отчётов. Для понимания принципов, стоящих за конструктором отчётов, получения нестандартных наборов данных и тестирования требуются базовые навыки обращения к БД. Основным языком, используемым для этого - SQL. Для аналитика базовым навыком является умение читать sql код, извлекать, объединять, фильтровать данные.

SQL (англ. structured query language — «язык структурированных запросов») — язык программирования, применяемый для создания, модификации и управления данными в базе данных, управляемой соответствующей системой управления базами данных [36].

СУБД Clickhouse. Клиент DBeaver для обращения к БД. Составление SQL запросов к витрине данных

Рассмотрим аналитическую колоночную open-source СУБД Clickhouse, разработанную компанией Яндекс. Для обращения к БД можно использовать различные клиенты, перечисленные в документации - [Визуальные интерфейсы от сторонних разработчиков](https://clickhouse.tech/docs/ru/interfaces/third-party/gui/#vizualnye-interfeisy-ot-storonnikh-razrabotchikov) (<https://clickhouse.tech/docs/ru/interfaces/third-party/gui/#vizualnye-interfeisy-ot-storonnikh-razrabotchikov>).

В разделах далее будет рассмотрен Dbeaver – свободно распространяемое ПО для обращения к БД.

Примечание. DBeaver имеет возможность подключения к различным базам данных: MySQL, Clickhouse, PostgreSQL, SQLite, Oracle, SQL Server, MS Access, Firebird, Apache Hive и другим.

Начало работы с DBeaver

Необходимо скачать и установить клиент с официального сайта (<https://dbeaver.io/>), см. Рис. 33.

После установки, можно увидеть интерфейс, приведенный на рисунке, см. Рис. 34.

Для подключения к новой БД необходимо кликнуть “Новое соединение”, выбрать СУБД, к которой производится подключение, см. Рис. 35, и ввести данные для доступа к БД. Данные для доступа следует получить у Системного администратора, см. Рис. 36.

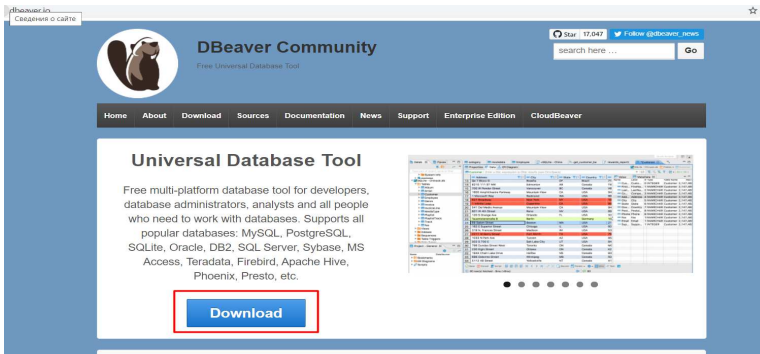


Рисунок 33 Скриншот установки клиента DBeaver с официального сайта

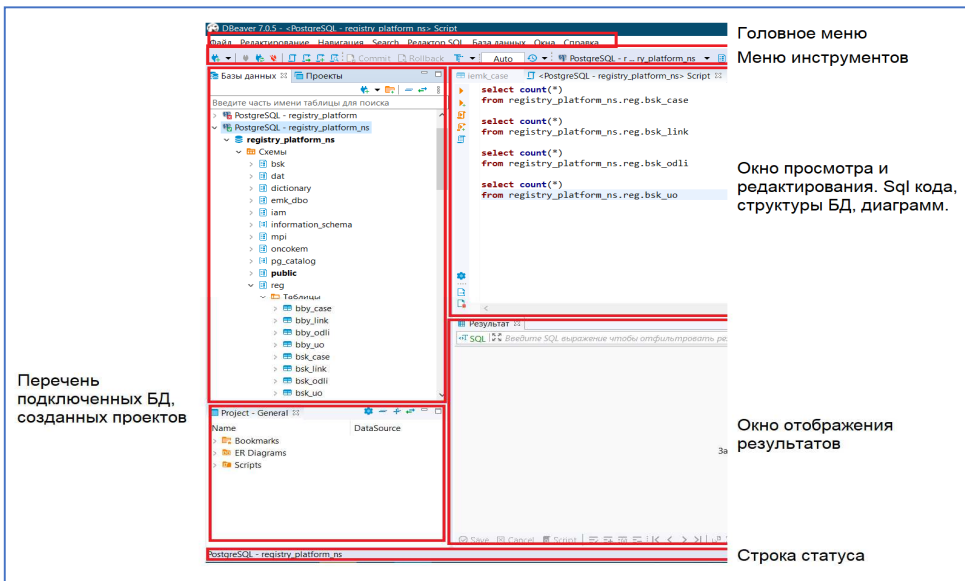


Рисунок 34 Скриншот DBeaver с перечнем подключенных БД

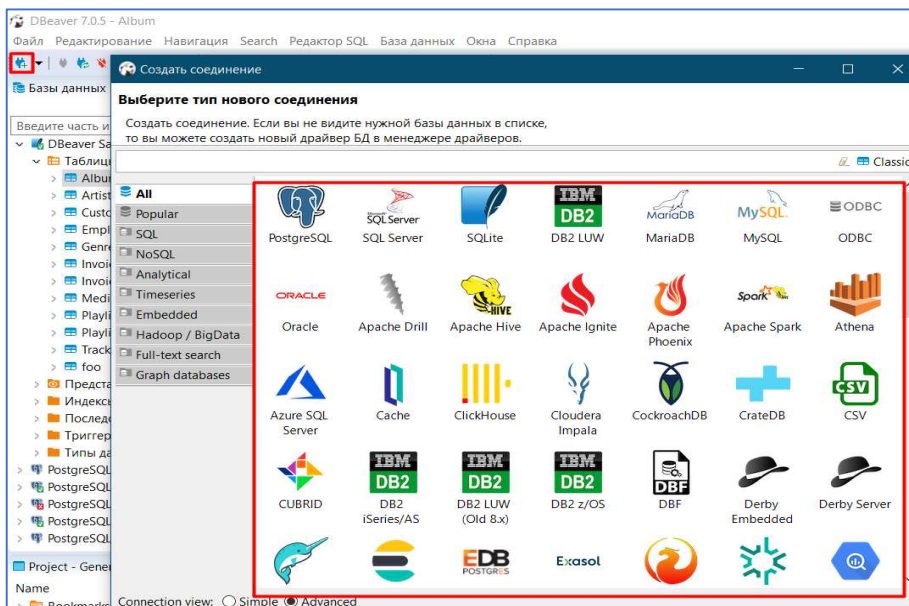


Рисунок 35 Выбор типа нового соединения с БД

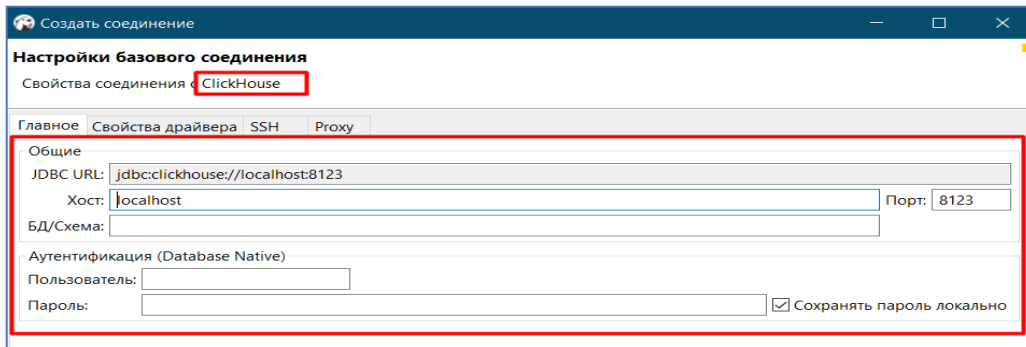


Рисунок 36 Настройки доступа к БД

Ошибки при подключении к БД могут возникать по причине двух основных проблем – ввода неверных данных для доступа: логин, пароль, название БД и т.д. или в случае отсутствия доступа к ресурсу.

Для проверки доступности БД нужно выполнить Ping <адрес БД> в командной строке. Команда проверяет доступность ресурса сети (сайта, адреса БД и т.п.) по протоколам технического уровня, см. Рис. 37.

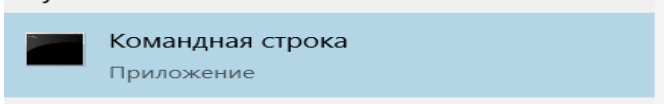


Рисунок 37 Командная строка

Пример доступного ресурса приведен на рисунке, см. Рис.38.

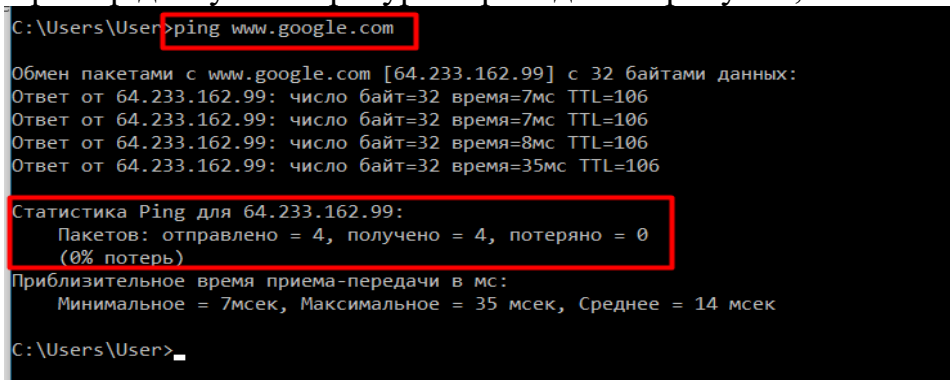


Рисунок 38 Пример доступного ресурса

Пример недоступного ресурса приведен на рисунке, см. Рис. 39:

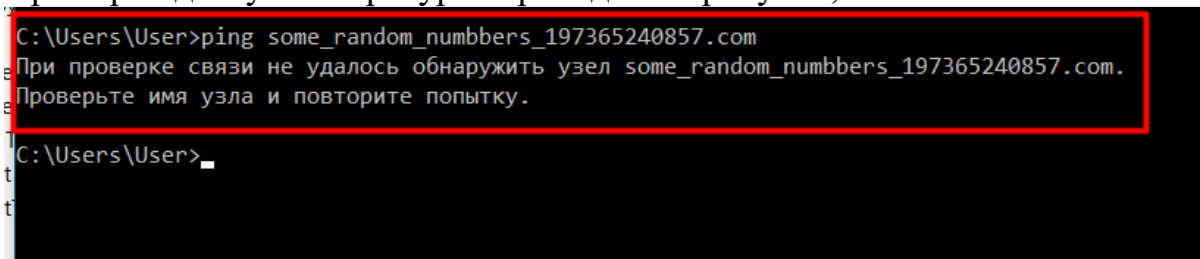


Рисунок 39 Пример недоступного ресурса

Базовый анализ БД

После подключения БД её структура отобразится в меню «Базы данных». В зависимости от СУБД структура может несколько отличаться, см. Рис. 40.

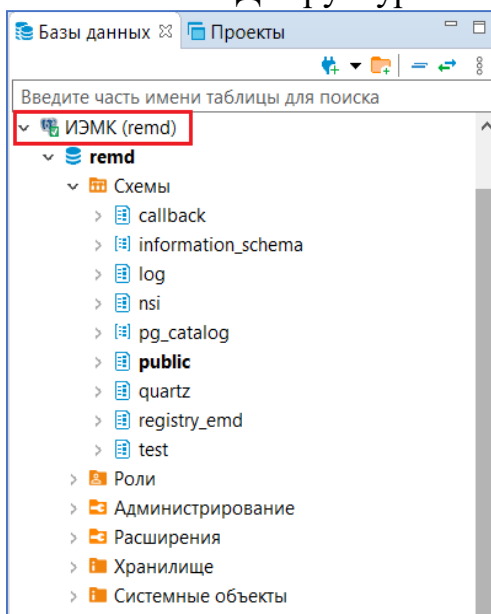


Рисунок 40 Базы данных DBeaver после подключения БД

В содержание и описание БД вложена различная информация, представляющая интерес для изучения. Базовой структурой являются таблицы, их описание и данные, см. Рис. 41.

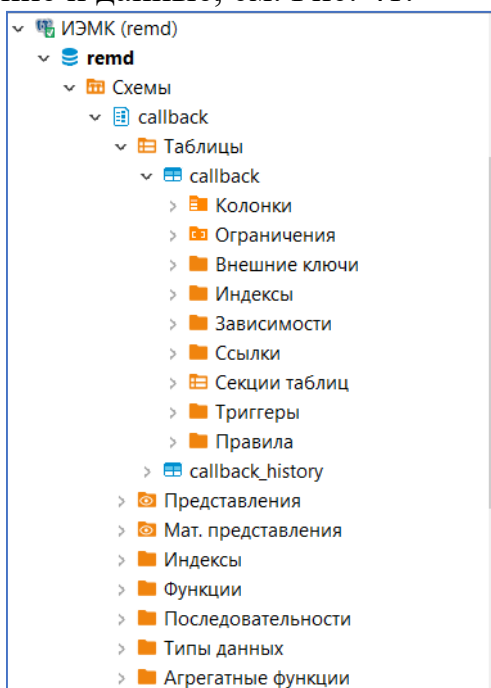


Рисунок 41 Таблицы, их описание и данные БД

При выборе таблицы двойным кликом можно посмотреть различную информацию по ней. В грамотно спроектированной системе большую часть необходимых для анализа сведений (таблиц, столбцов, типов данных, связей, информационных сущностей и т.п.) можно почерпнуть из описания БД.

Пример описания перечня столбцов, их свойств, названий, уникальных ключей и т.п. приведен на Рис. 42.

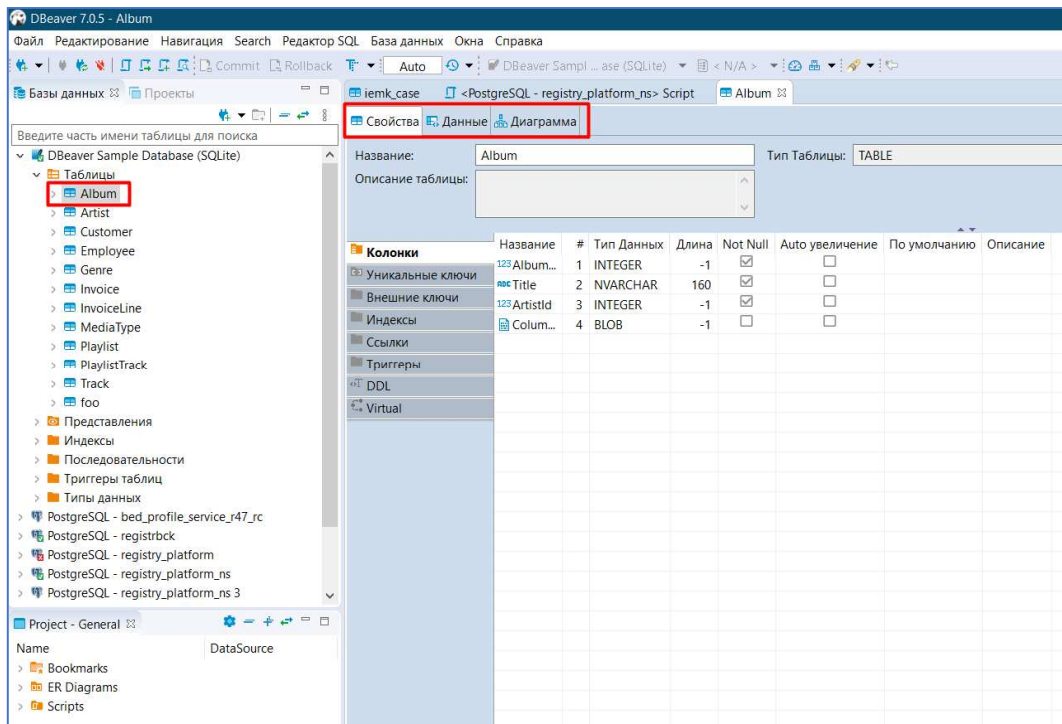


Рисунок 42 Описание перечня столбцов, их свойств, названий, уникальных ключей

Пример раздела данных, содержащихся в таблице приведен на Рис. 43, где видно, что могут появляться дополнительные инструменты в различных окнах.



Рисунок 43 Пример раздела данных, содержащихся в таблице

Пример ER диаграммы связей таблицы с другими таблицами приведен на Рис. 44.

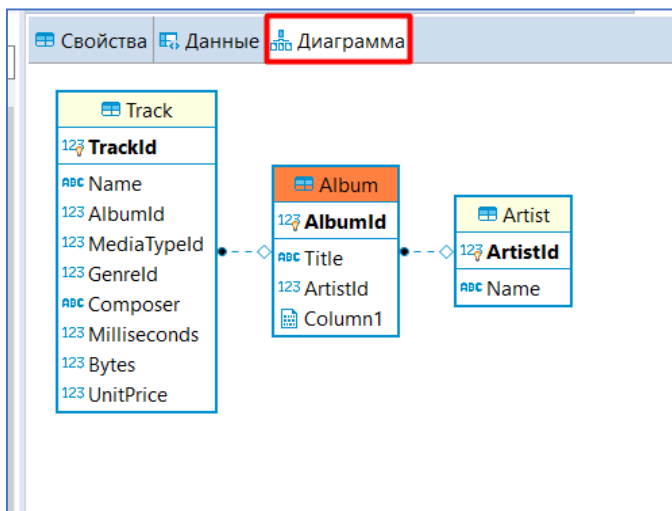


Рисунок 44 Пример раздела данных, содержащихся в таблице

Выполнение SQL запросов

Для выполнения SQL запросов или скриптов необходимо создать/открыть окно, см. Рис. 45.

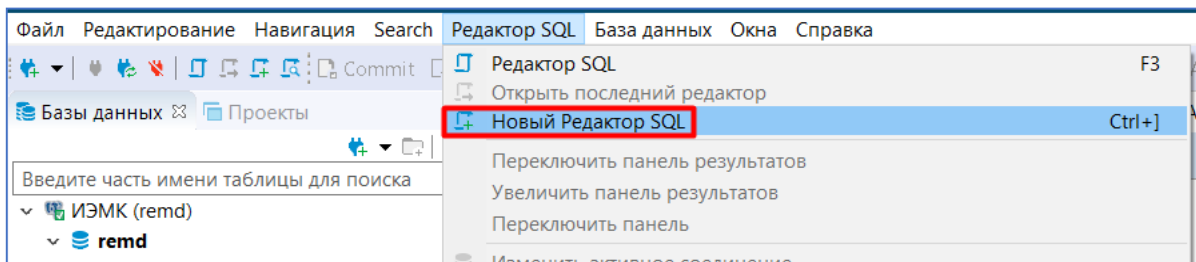


Рисунок 45 Пример раздела данных, содержащихся в таблице

Пример открытия окна с ранее выполненными запросами приведен на Рис. 46.

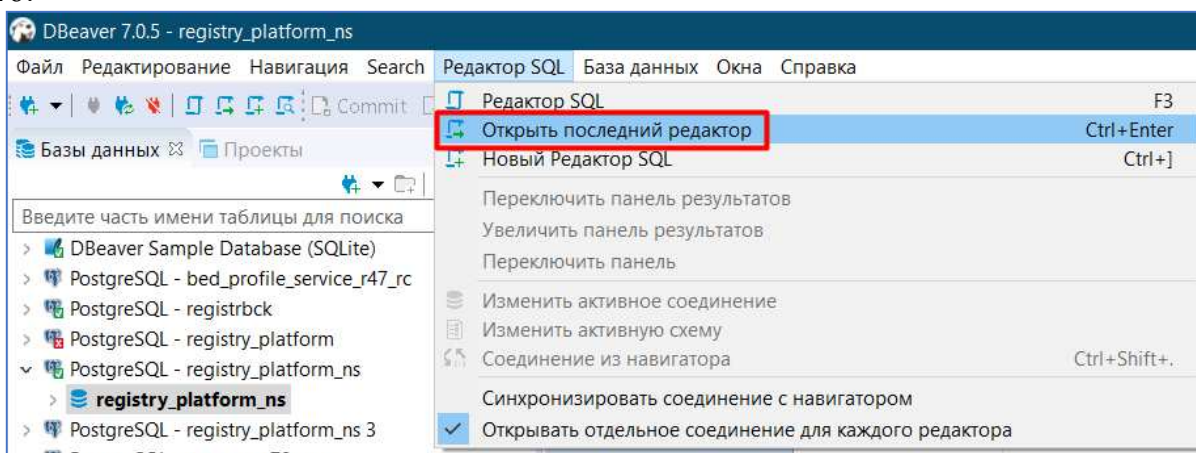


Рисунок 46 Пример раздела данных, содержащихся в таблице

При составлении запросов к таблице необходимо указать путь к ней.

Название БД указывать необязательно, но это является хорошей практикой формирования кода. Если запрос будет храниться в отдельных файлах, экспортироваться, пересылаться разработчикам – в запросе не потеряется название БД, Рис. 47, Рис. 48.

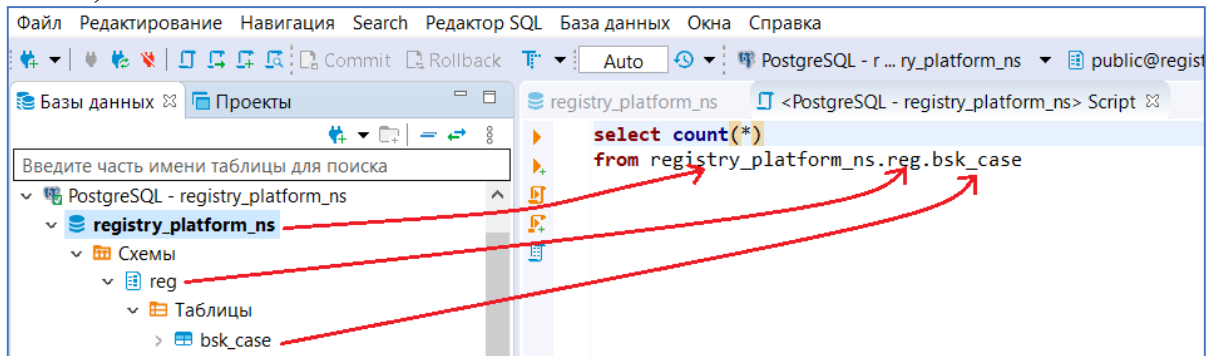


Рисунок 47 Пример обращения к БД

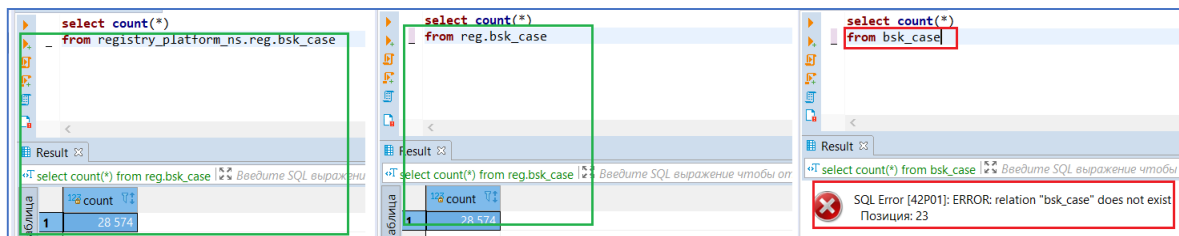


Рисунок 48 Примеры результатов выполнения запросов

Хорошей практикой являются комментарии в коде о решаемой бизнес-задаче или другие пояснения, см. Рис. 49, Рис. 50.

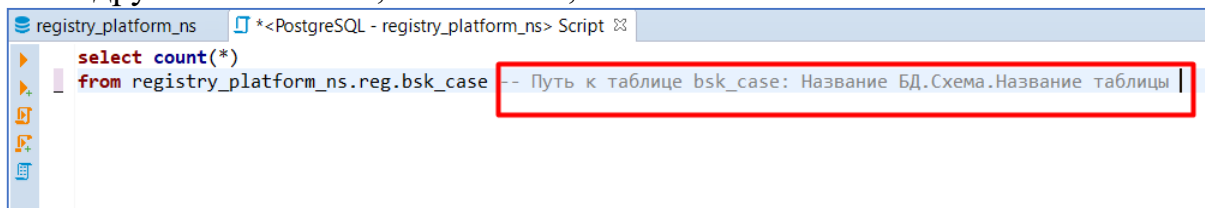


Рисунок 49 Комментарии в коде о решаемой бизнес-задаче

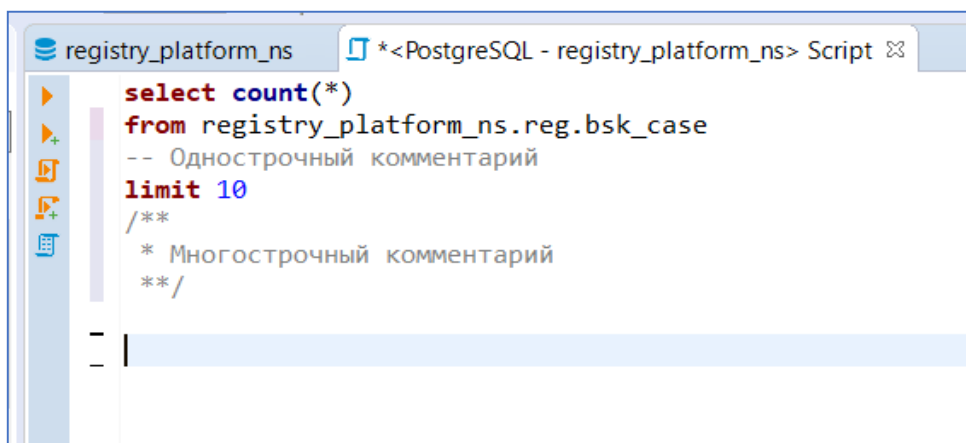


Рисунок 50 Пример различных комментариев в SQL запросе

Базовые SQL конструкции в терминах колоночной БД clickhouse

Приведённые ниже конструкции являются универсальными для СУБД, использующих SQL. Они могут незначительно отличаться синтаксисом в разных СУБД. К примеру, в ряде БД необходимо брать в кавычки названия столбцов.

Справку по функциям можно получить по ссылке - <https://clickhouse.tech/docs/ru/sql-reference/statements/> [47].

Синтаксис конструкции SELECT

Общая конструкция для извлечения данных. Синтаксис:

SELECT [**DISTINCT**] *expr_list* – перечисление отбираемых колонок через запятую.

[FROM [*db.*]**table** | (*subquery*)] – таблица или подзапрос, откуда извлекаются данные.

[WHERE *expr*] – фильтры, ограничения на исходные данные. К примеру, *where data > '2020-12-06' and IdCase is not null*. Базовые операции - *<, >, !=, =, <>*, *in (... , ...)* отрицание – через приставку *not*. Соединение условий – через логические операторы **OR**, **AND**.

[GROUP BY *expr_list*] – перечень колонок и порядок, по которому происходит группировка. При использовании агрегатных функций необходимо перечислять остальные колонки в *group by*.

[HAVING *expr*] – перечень условий, по которому происходит постфильтрация данных. К примеру, вывести только строки с агрегатной функцией *count(*)* больше 100 - *having count(*)>100*.

[ORDER BY *expr_list*] – перечисление колонок, по которым необходима сортировка, **ASC** - по возрастанию, **DESC** - по убыванию, **ORDER BY column DESC**.

[LIMIT [*offset_value*,]*n* **BY** *columns*] - установка лимита на число выводимых строк.

Примеры:

Получить 100 строк данных из витрины данных *iemk*:

```
SELECT *
FROM iemk
LIMIT 100
```

Получить названия и типы колонок в таблице *iemk*:

```
SELECT name, type
FROM system.columns
WHERE table='iemk'
```

COUNT. UNIQU и другие агрегатные функции

count(expr) - функция вычисляет количество раз, когда выражение возвращает не NULL. COUNT(DISTINCT expr) - оператор distinct отбирает уникальные значения, затем работает функция count.

uniq() – функция приближённо вычисляет количество различных значений аргумента, а функция uniqExact() – точно.

uniqIf(column, expression) - вычисляет функцию uniq от данных, в которых expression true.

Доступны и другие агрегатные функции – к примеру, avg, max, min, quantile и другие. Справка по агрегатным функциям - <https://clickhouse.tech/docs/ru/sql-reference/aggregate-functions/>.

JOIN. Конструкция для объединения таблиц данных

Синтаксис:

```
FROM <left_table>  
[INNER|LEFT|RIGHT|FULL|CROSS] JOIN <right_table>  
(ON <expr_list>)|(USING <column_list>) ...
```

- INNER JOIN, возвращаются только совпадающие строки.
- LEFT JOIN, несовпадающие строки из левой таблицы возвращаются в дополнение к совпадающим строкам.
- RIGHT JOIN, несовпадающие строки из правой таблицы возвращаются в дополнение к совпадающим строкам.
- FULL JOIN, несовпадающие строки из обеих таблиц возвращаются в дополнение к совпадающим строкам.
- CROSS JOIN, производит декартово произведение таблиц целиком, ключи соединения не указываются.

Без указания типа JOIN подразумевается INNER.

Требуется, чтобы столбцы, указанные в USING, назывались одинаково в обоих подзапросах, а остальные столбцы - по-разному. Изменить имена столбцов в подзапросах можно с помощью синонимов, к примеру, select column as column_to_join...

Пример. ab_user - перечень пользователей системы. logs - логи действий пользователей. Отчёт строит число действий по именам пользователей по дням.

```
SELECT  
  ab_user.username AS user,
```

```
        date_trunc('day', logs.dttm) AS day,  
        count(*)  
FROM  
    logs LEFT JOIN ab_user  
ON logs.user_id = ab_user.id  
GROUP BY ab_user.username  
ORDER BY day ASC
```

UNION. Конструкция для дополнения таблицы данными другого запроса

Предпочтительно использовать union для колоночных БД при объединении данных.

Можно использовать UNION ALL чтобы объединить любое количество SELECT запросы путем расширения их результатов.

Синтаксис:

```
SELECT columnID, ...  
FROM table1  
UNION ALL  
SELECT columnID, ...  
FROM table2  
UNION ALL  
UNION. Пример.  
SELECT CounterID, 1 AS table, toInt64(count()) AS c  
    FROM test.hits  
    GROUP BY CounterID  
UNION ALL  
SELECT CounterID, 2 AS table, sum(Sign) AS c  
    FROM test.visits  
    GROUP BY CounterID  
    HAVING c > 0
```

Практические задания

1. Исходная таблица случаев медицинского обслуживания case в схеме iemk. Названия соответствуют api [39]
 - a. Case:
 - i. IdCase
 - ii. IdOrganization
 - iii. IdDoctor
 - iv. IdPatient
 - v. OpenDate
 - vi. CloseDate
 - vii. CaseType

- b. MedDocument
 - i. IdMedDocument
 - ii. Attachments
 - iii. CreationDate
 - iv. Header
 - v. IdDocumentMis
- 2. Таблица CaseTypeTerminology в схеме nsi с колонками
 - a. IdCaseType
 - b. Name
- 3. Таблица OrganizationTerminology в схеме nsi с колонками
 - a. IdOrganization
 - b. Name
 - c. District
- 4. Требуется составить запросы к источнику данных:
 - a. Число СМО всего
 - b. Число медицинских документов всего
 - c. Число медицинских документов с вложением
 - d. Число СМО по дате закрытия случая медицинского обслуживания
 - e. Число СМО ежемесячно по дате открытия
 - f. Число СМО по наименованиям медицинских организаций
 - g. Среднее число пациентов, которых пролечили врачи
 - h. Число пациентов, пролеченных в медицинских организациях (с выведением наименования МО) ежемесячно
 - i. Вывести список СМО, врачей, наименование МО, в которых врач передал документы по пациенту без вложения

Пример

Коллеги из региона прислали скрипт – как делают отчёты на системе-источнике. Нужно разобраться, перевести в `ui` конструктор отчётов BI. Требуется уровень SQL, достаточный, чтобы читать и писать такие запросы.

Запрос ниже написан для PostgreSQL, структура соответствует `api` подсистемы ИЭМК (<http://api.n3zdrav.ru/api/iemk/iemk80/>). `[dbo].[nsi_Lpu]` - таблица с полями `Guid` (код ЛПУ) и `Name` (Название ЛПУ).

P.S. В исходном запросе могут быть ошибки в бизнес-логике.

P.P.S. Иногда задача нерешаемая. Тогда нужно общаться с Заказчиком и находить решение, иногда делать “кастомный” отчёт со сложными SQL запросами к витрине данных, иногда дорабатывать витрину данных, иногда дорабатывать систему-источник данных.

“[Переслано от Александр]

Обещал вам прислать скрипты которые использовал для получения количества кейсов

[Переслано от Александр]

```
select t1.IdLpu, t1.Количество, t2.[Количество документов], (select [Name]
from [dbo].[nsi_Lpu] where [nsi_Lpu].Guid=t1.IdLpu)
from
(select [case].IdLpu, COUNT( [case].IdLpu) as 'Количество'
from [case]
where [case].CreationDate>'01.01.2020'
group by [case].IdLpu
--order by [Количество] desc
) t1
left join
(select [case].IdLpu, COUNT( [case].IdLpu) as 'Количество документов'
from [case]
join Step on Step.IdCase=[case].IdCase
join MedDocument on MedDocument.IdStep=Step.IdStep
where [case].CreationDate>'01.01.2020'
group by [case].IdLpu) t2
on t1.IdLpu=t2.IdLpu
left join
(select [case].IdLpu, COUNT( distinct [case].IdCase) as 'Количество
документов'
from [case]
join Step on Step.IdCase=[case].IdCase
join MedDocument on MedDocument.IdStep=Step.IdStep
where [case].CreationDate>'01.01.2020'
group by [case].IdLpu) t3
on t1.IdLpu=t3.IdLpu
order by t1.Количество desc
[Переслано от Александр]
такой отчет формирую ежемесячно”
```

Вопросы:

1. Как будет выглядеть отчёт? Опишите перечень столбцов и колонок.
2. Какая бизнес-логика формирования отчёта в терминах апи? Что считает пользователь?
3. Есть ли ошибки при формировании отчёта? Какие?

Составление SQL запросов к витрине данных. Продвинутое выборки данных

Запросы над непересекающимися параметрами

Пусть для устранения размножения данных из раздела “Особенности проектирования плоских витрин больших данных” был выбран способ, когда размножающие параметры не пересекаются. Пример показан на Рис. 51.

datasource	calendar	case_biz_key	diagnosis_mkb_code3	diagnosis_mkb_code4	servise_type_code	servise_type_name
iemk_diagnosis	2020-11-17	200192922	null	null	null	null
iemk_services	2020-11-17	200192922	null	null	A16.12.056	Шунтирование аорты
iemk_diagnosis	2020-11-17	200192922	I21	I21.0	null	null
iemk_services	2020-11-17	200192587	null	null	A16.12.056	Шунтирование аорты
iemk_diagnosis	2020-11-17	200192587	null	null	null	null

Рисунок 51 Пример запроса с непересекающимися параметрами

Как видно, число строк в итоговой витрине данных зависит от суммы строк услуг и диагнозов в исходных данных вместо декартового произведения. Однако подобные витрины требуют особых способов обращения с ними.

Проблема – требуется посчитать число СМО, в рамках которых было проведено лечение по группе “Аортокоронарное шунтирование”. Пример данных приведён выше.

Решение. Терминологический разбор условий Заказчика. Аортокоронарное шунтирование определяется следующим образом. Диагноз принадлежит множеству ('I20.0','I20.8','I20.9','I24.0', 'I21','I22','I24','I25'), вместе с этим пациенту выполнялось Шунтирование аорты (код услуги A16.12.056).

В терминах витрины данных iemk:

- (diagnosis_mkb_code4 in ('I20.0','I20.8','I20.9','I24.0') OR diagnosis_mkb_code3 in ('I21','I22','I24','I25'))
- servise_type_code IN ('A16.12.056')

Решение. SQL запрос к витрине данных с использованием стандартных конструкций group by, having и подзапросов.

1. Требуется определить наличие в СМО указанных диагнозов и услуг. Для этого группируем данные по идентификатору СМО и считаем число диагнозов, услуг, которые попали в условия выше, Рис. 52.

```
select case_biz_key,
       uniqIf(1,(diagnosis_mkb_code4 in ('I20.0','I20.8','I20.9','I24.0')
                OR diagnosis_mkb_code3 in ('I21','I22','I24','I25'))) as
ao_shunt_d,
       uniqIf(1,lowerUTF8(servise_type_code) IN ('a16.12.056')) as
ao_shunt_s
from reg_platform_bsk_mart
```

group by case_biz_key

```
1 select
2   case_biz_key,
3   uniqIf(1,(diagnosis_mkb_code4 in ('I20.0','I20.8','I20.9','I24.0')
4     OR diagnosis_mkb_code3 in ('I21','I22','I24','I25'))) as ao_shunt_d,
5   uniqIf(1,lowerUTF8(servise_type_code) IN ('a16.12.056')) as ao_shunt_s
6 from r42_rc.reg_platform_bsk_mart
7 group by
8   case_biz_key
9 LIMIT 100
10
```

Выполнить запрос Сохранить запрос Поделиться запросом

Результаты История запросов

Построить отчет Экспорт в CSV В csv будет выгружено не более 65000 записей

case_biz_key	ao_shunt_d	ao_shunt_s
144794	null	null
200189218	1	0
210035437	1	null
914241	null	null

Рисунок 52 Пример SQL-запроса 1 и результата

2. Требуется отобразить СМО, у которых есть указанные диагнозы и услуги, для чего применим функцию HAVING на метрике. То есть, отбираем только те СМО, что удовлетворяют нашим условиям, Рис. 53.

```
select case_biz_key,
       uniqIf(1,(diagnosis_mkb_code4 in ('I20.0','I20.8','I20.9','I24.0')
       OR diagnosis_mkb_code3 in ('I21','I22','I24','I25'))) as
ao_shunt_d,
       uniqIf(1,lowerUTF8(servise_type_code) IN ('a16.12.056')) as
ao_shunt_s
from reg_platform_bsk_mart
group by case_biz_key
having (ao_shunt_d
       and ao_shunt_s)
```



```

1  select
2     case_biz_key,
3     uniqIf(1,(diagnosis_mkb_code4 in ('I20.0','I20.8','I20.9','I24.0')
4         OR diagnosis_mkb_code3 in ('I21','I22','I24','I25'))) as ao_shunt_d,
5     uniqIf(1,lowerUTF8(servise_type_code) IN ('a16.12.056')) as ao_shunt_s
6  from r42_rc.reg_platform_bsk_mart
7  group by
8     case_biz_key
9  having
10     (ao_shunt_d
11     and ao_shunt_s)
12  LIMIT 1000

```

В csv будет выгружено не более 65000 записей

case_biz_key	ao_shunt_d	ao_shunt_s
200190738	1	1
200192533	1	1

Рисунок 53 Пример SQL-запроса 2 и результата

3. Используя механизм подзапросов, считаем число СМО, Рис. 54.

```

select uniq(case_biz_key)
from
(select case_biz_key,
    uniqIf(1,(diagnosis_mkb_code4 in ('I20.0','I20.8','I20.9','I24.0')
        OR diagnosis_mkb_code3 in ('I21','I22','I24','I25'))) as
ao_shunt_d,
    uniqIf(1,lowerUTF8(servise_type_code) IN ('a16.12.056')) as
ao_shunt_s
from reg_platform_bsk_mart
group by
    case_biz_key
having (ao_shunt_d
    and ao_shunt_s))
LIMIT 1000

```

```

1 select uniq(case_biz_key)
2 from
3 (select case_biz_key,
4         uniqIf(1,(diagnosis_mkb_code4 in ('I20.0','I20.8','I20.9','I24.0')
5           OR diagnosis_mkb_code3 in ('I21','I22','I24','I25'))) as ao_shunt_d,
6         uniqIf(1,lowerUTF8(servise_type_code) IN ('a16.12.056')) as ao_shunt_s
7       from reg_platform_bsk_mart
8       group by
9         case_biz_key
10      having (ao_shunt_d
11             and ao_shunt_s))
12 LIMIT 1000

```

Выполнить запрос Сохранить запрос Поделиться запросом

Результаты История запросов

Построить отчет Экспорт в CSV В csv будет выгружено не более 65000 записей

uniq(case_biz_key)
44

Рисунок 54 Пример SQL-запроса 3 и результата

Решение. SQL запрос к витрине данных с использованием функции clickhouse groupUniqArray – см. Рис. 55.

```

select case
  when (hasAny(diagnosis_4, ['I20.0','I20.8','I20.9','I24.0'])
    or hasAny(diagnosis_3, ['I21','I22','I24','I25']))
    and has(servises, 'a16.12.056')
  then 'Аортокоронарное шунтирование (вид вмешательства АКШ)'
  end
  as estimate_for_bsk
,uniqExact(case_biz_key)
  as "Число сердечно-сосудистых
событий"
from (
  select case_biz_key
    as case_biz_key
    ,groupUniqArray(lowerUTF8(servise_type_code))
    as servises
    ,groupUniqArray(diagnosis_mkb_code3)
    as diagnosis_3
    ,groupUniqArray(diagnosis_mkb_code4)
    as diagnosis_4
  from reg_platform_bsk_mart
  group by case_biz_key)
where estimate_for_bsk is not null
group by estimate_for_bsk
LIMIT 1000

```

```

1  select case
2      when (hasAny(diagnosis_4, ['I20.0','I20.8','I20.9','I24.0'])
3           or hasAny(diagnosis_3, ['I21','I22','I24','I25']))
4           and has(services, 'a16.12.056')
5           then 'Аортокоронарное шунтирование (вид вмешательства АКШ)'
6           end
7           as estimate_for_bsk
8           ,uniqExact(case_biz_key)
9           as "Число сердечно-сосудистых событий"
10          from (
11              select case_biz_key
12                     ,groupUniqArray(lowerUTF8(servise_type_code))
13                     ,groupUniqArray(diagnosis_mkb_code3)
14                     ,groupUniqArray(diagnosis_mkb_code4)
15              from reg_platform_bsk_mart
16              group by case_biz_key
17          where estimate_for_bsk is not null
18          group by estimate_for_bsk
19          LIMIT 1000

```

Выполнить запрос | Сохранить запрос | Поделиться запросом

Результаты | История запросов

Построить отчет | Экспорт в CSV | В csv будет выгружено не более 65000 записей

estimate_for_bsk	Число сердечно-сосудистых событий
Аортокоронарное шунтирование (вид вмешательства АКШ)	44

Рисунок 55 Пример SQL-запроса 4 и результата

Использование подзапросов в фильтрах

Задача – вывести названия медицинских организаций, которые не передают данные.

Дано – таблица данных со всеми МО - organizations_list, витрина данных с передающими МО - remd_mart, Рис. 56, Рис. 57.

```

12 SELECT *
13 FROM remd_mart
14 LIMIT 50

```

Выполнить запрос | Сохранить запрос | Поделиться запросом

Результаты | История запросов

Построить отчет | Экспорт в CSV | В csv будет выгружено не более 65000 записей

calendar	organization_id	mis_name	med_document_type_Code	med_document_type	med_document
2020-01-01	06d99168-aadc-45e0-a183-e809e7cafcea	П	3	Протокол консультации	22
2020-01-01	7e0c9e34-1c45-469b-b3a6-709bc011ef1c	А	3	Протокол консультации	4
2020-01-01	18416900-707e-4e23-a0c4-959657f78071	М	3	Протокол консультации	1
2020-01-02	7e0c9e34-1c45-469b-b3a6-709bc011ef1c	А	3	Протокол консультации	5

Рисунок 56 Пример SQL-запроса 5 и результата

```

12 SELECT *
13 FROM organizations_list
14 LIMIT 50

```

Выполнить запрос | Сохранить запрос | Поделиться запросом

Результаты | История запросов

Построить отчет | Экспорт в CSV | В csv будет выгружено не более 65000 записей

oid	organization_level1_key	organization_level1_name
1.2.643.5.1.13.13.12.2.47.4460	178a2789-6b00-48d6-b77f-281c8f1567d6	ГБУЗ Л.
1.2.643.5.1.13.13.12.2.47.4424	2b16264a-57ea-4741-b387-8fa1f6586204	ГБУЗ ЛО "Г"
1.2.643.5.1.13.13.12.2.47.4420	75af55dd-cd5f-4018-869a-3a3397cdc4e5	ГБУЗ ЛО "К"

Рисунок 57 Пример SQL-запроса 6 и результата

Решение – вывести все МО по справочнику, кроме тех, кто передаёт. Передающие МО взять с помощью подзапроса в фильтр, Рис. 58.

```

SELECT organization_level1_short_name AS organization_level1_short_name
FROM organizations_list
WHERE (organization_level1_key not in
      (SELECT organization_id
       FROM remd_mart
       WHERE med_document_cnt>=1
       GROUP BY organization_id))
GROUP BY organization_level1_short_name
ORDER BY COUNT(*) ASC
LIMIT 500

```

```

1 SELECT organization_level1_short_name AS organization_level1_short_name
2 FROM organizations_list
3 WHERE (organization_level1_key not in --Убираем передающие МО
4
5      (SELECT organization_id
6       FROM remd_mart
7       WHERE med_document_cnt>=1
8       GROUP BY organization_id))
9
10 GROUP BY organization_level1_short_name
11 ORDER BY COUNT(*) ASC
12 LIMIT 500

```

Выполнить запрос | Сохранить запрос | Поделиться запросом

Результаты | История запросов

Построить отчет | Экспорт в CSV | В csv будет выгружено не более 65000 записей

organization_level1_short_name
ЛОГБУ "Г"
ЛОГБУ "В"

Рисунок 58 Пример SQL-запроса 7 и результата

Визуализация данных

Визуализация данных — представление данных в наиболее удобном для восприятия формате [48]. Распространённые, основные виды визуализаций приведены на Рис. 59.

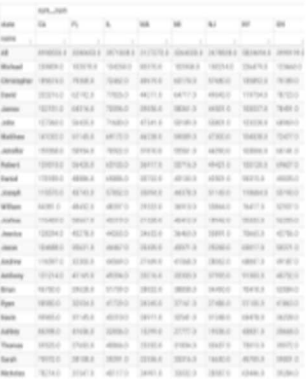
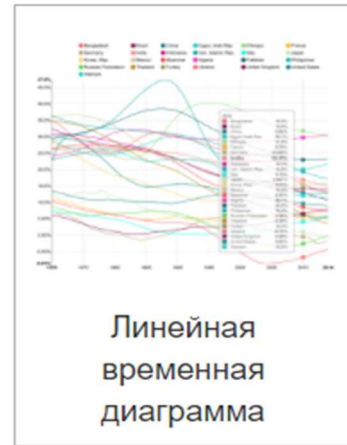
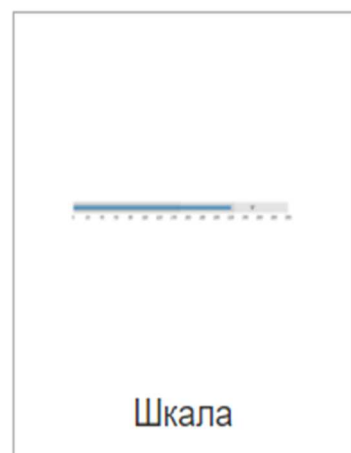
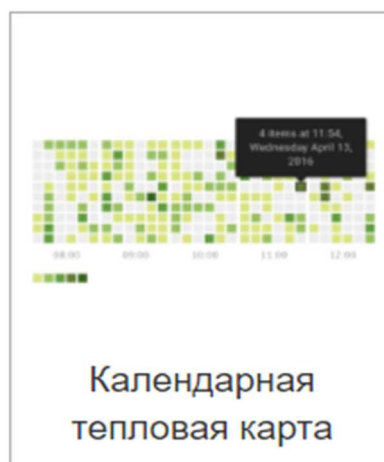


Table with multiple columns and rows of numerical data. The columns are labeled with letters (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z) and the rows are labeled with letters (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z). The data values are numerical, ranging from 0 to 1000.

Таблица среза



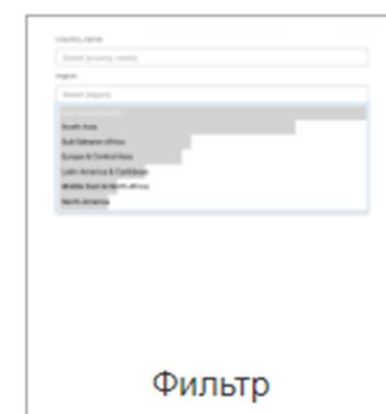
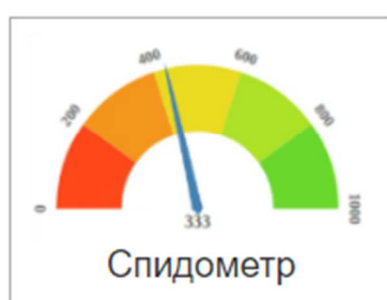
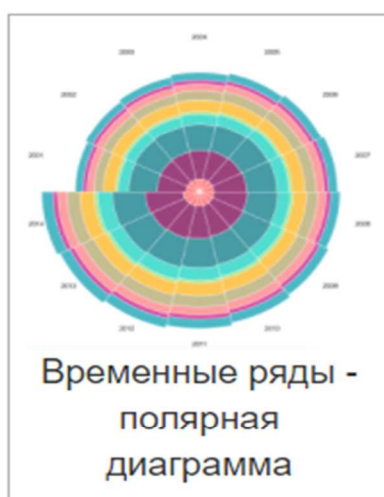
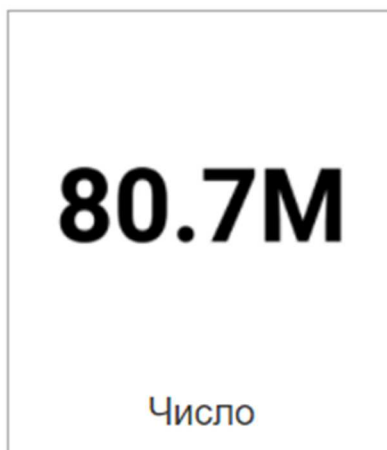


Рисунок 59 Основные виды визуализаций данных

Дашборд (информационная панель) - набор отчётов, размещённых в едином пространстве.

Дашборд представляет визуализацию данных по заданной проблематике определённой аудитории.

Drill-down (проваливание, углубление) - общее название для методов перехода, детализации между отчётами, инфопанелями, показателями и наборами данных.

При выстраивании информационных панелей полезно организовать отчёты в виде иерархии. Корневой дашборд покажет общие цифры, динамику показателей. При необходимости получить детализированные данные пользователь получит возможность в них перейти.

Есть различные инструменты для взаимодействия с отчётами, инфопанелями - интерактивный фильтр, возможность переходить по иерархии показателей и другие.

Примеры.

Дисклеймер. Данные искусственно изменены. Примеры служат только целям обучения.

Инфопанель по переданным лабораторным исследованиям на определение SARS coronavirus 2 приведена на рисунке, см. Рис. 60. С возможностью “провалиться” на инфопанель на Рис. 61 по клику на отчёт.

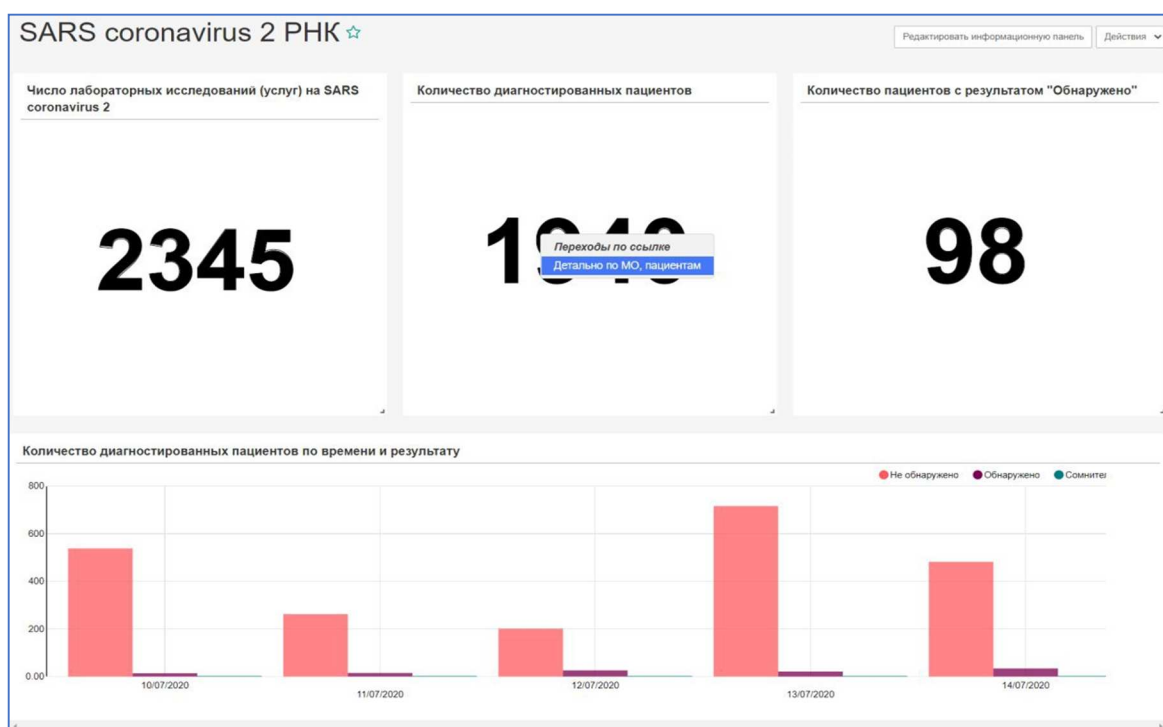


Рисунок 60 Инфопанель по переданным лабораторным исследованиям на определение SARS coronavirus 2

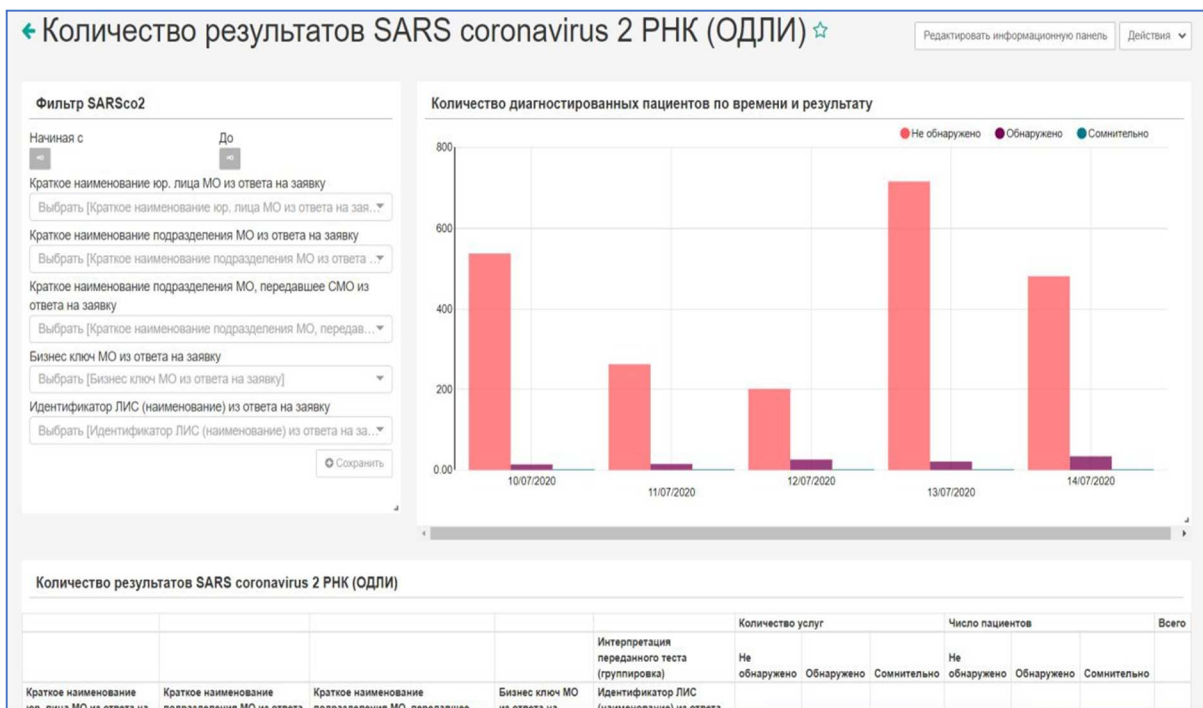


Рисунок 61 Инфопанель по количеству результатов SARS coronavirus 2 ПНК

Дашборд по показателям записи на приём к врачу приведен на рисунке, см. Рис. 62.

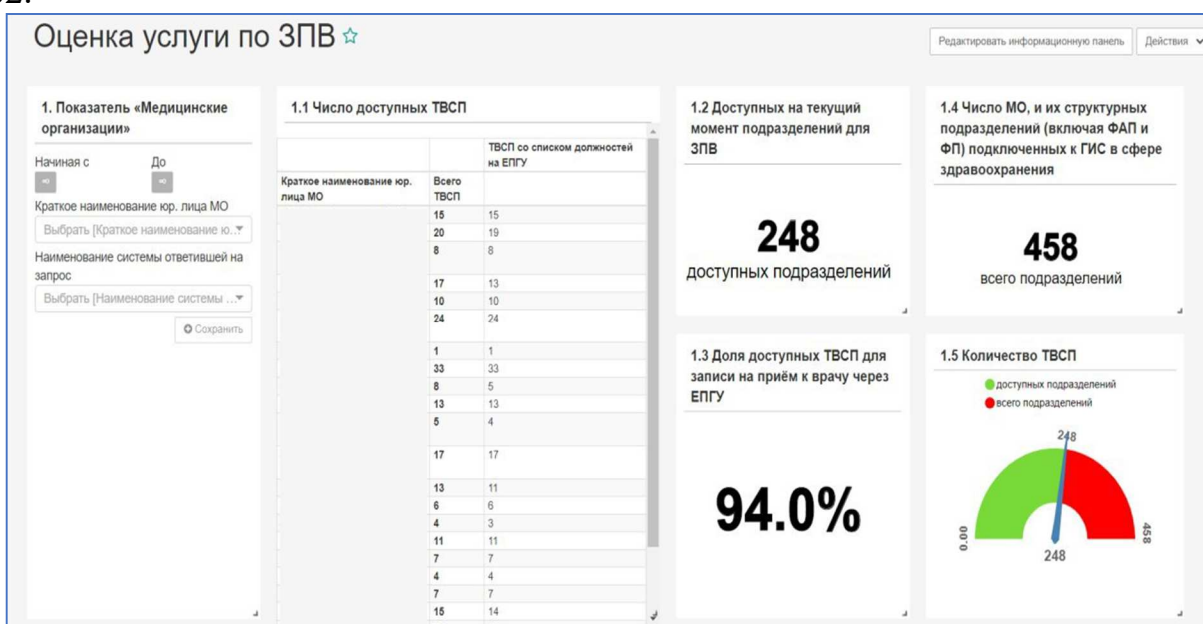


Рисунок 62 Дашборд по показателям записи на приём к врачу

Пример отчёта с визуальным выделением особо важных данных приведен на Рис. 63.

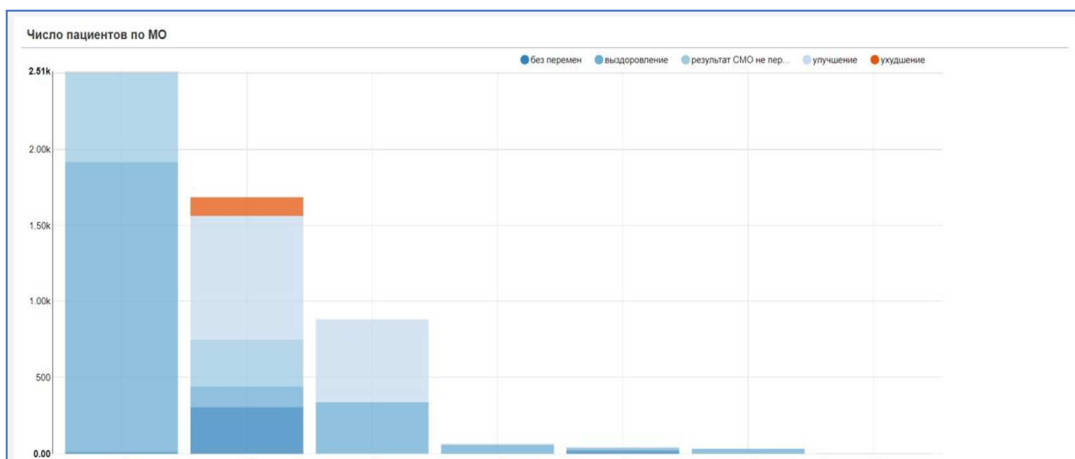


Рисунок 63 Пример отчёта с визуальным выделением особо важных данных

Пример отчёта с искусственной группировкой в соответствии с требованиями нормативных документов приведен на Рис. 64. По Территориальной программе государственных гарантий пациент должен иметь возможность попасть к врачам амбулаторного звена, к примеру – поликлиник, общих специальностей (терапевт, врач общей практики) в течение 24 часов, узких специальностей (ЛОР, хирург) - в течение 14 дней.

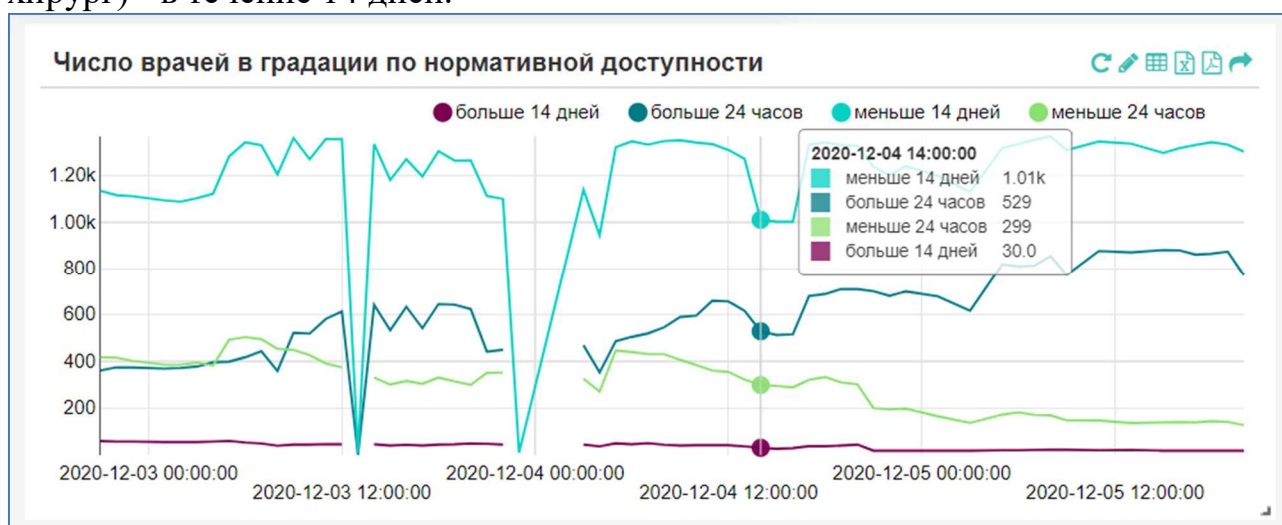


Рисунок 64 Пример отчёта с искусственной группировкой

Практическое задание

Выберите тему из представленных ниже или самостоятельно придумайте. Подготовьте дашборд. Считайте, что в ИАС есть все необходимые данные. Опишите почему выбраны именно такие показатели и визуализации, кто будет потребителем информации.

1. Проблема глобального потепления.
2. Экология города/области, края.
3. Уборка улиц СПб от снега.
4. Здоровье россиянина.

5. Занятость и труд.
6. Безопасность.
7. Финансово-экономическое состояние организации, региона.
8. Показатели социально-экономического развития.
9. Выборы.
10. Общественное мнение.
11. СМИ и социальные сети.
12. ЖКХ.

Пример. Показатели Национального проекта Создание единого цифрового контура в сфере здравоохранения

Важные показатели, которые задают тренд на информатизацию для всех медицинских организаций, оказывающих помощь по ОМС (официальный источник - <https://rosstat.gov.ru/metod/fed-proekt/FP0408.htm>):

1. Число граждан, воспользовавшихся услугами (сервисами) в Личном кабинете пациента «Мое здоровье» на Едином портале государственных услуг и функций в отчетном году.
2. Доля медицинских организаций государственной и муниципальной систем здравоохранения, обеспечивающих преемственность оказания медицинской помощи гражданам путем организации информационного взаимодействия с централизованными подсистемами государственных информационных систем в сфере здравоохранения субъектов Российской Федерации.
 - 2.1. Расчет показателя по годам реализации федерального проекта осуществляется по подключению МО региона к следующим централизованным подсистемам:
 - 2.1.1.1. Управление потоками пациентов;
 - 2.1.1.2. Управление скорой и неотложной медицинской помощью (в том числе санитарной авиации);
 - 2.1.1.3. Управление льготным лекарственным обеспечением (подключение ТВСП МО);
 - 2.1.1.4. Управление льготным лекарственным обеспечением (подключение Аптечных пунктов);
 - 2.1.1.5. Интегрированная электронная медицинская карта;
 - 2.1.1.6. Центральный архив медицинских изображений;
 - 2.1.1.7. Лабораторные исследования.
 - 2.1.1.8. Организации оказания медицинской помощи по профилям «Акушерство; и гинекология» и «Неонатология» (Мониторинг беременных);

- 2.1.1.10. Организации оказания медицинской помощи больным онкологическими заболеваниями;
 - 2.1.1.11. Организации оказания профилактической медицинской помощи (диспансеризация, диспансерное наблюдение, профилактические осмотры);
 - 2.1.1.12. Организации оказания медицинской помощи больным сердечно-сосудистыми заболеваниями;
 - 2.1.1.13. Телемедицинские консультации.
3. Доля медицинских организаций, обеспечивших создание и предоставление электронных медицинских документов гражданам в Личном кабинете пациента «Мое здоровье» на Едином портале государственных услуг, %.
 4. Доля медицинских организаций государственной и муниципальной систем здравоохранения, использующих медицинские информационные системы для организации и оказания медицинской помощи гражданам, обеспечивающих информационное взаимодействие с ЕГИСЗ.

Примеры информационных панелей по показателям Цифрового контура приведены на рисунках, см. Рис. 65, Рис. 66, Рис. 67, Рис. 68, Рис. 69.

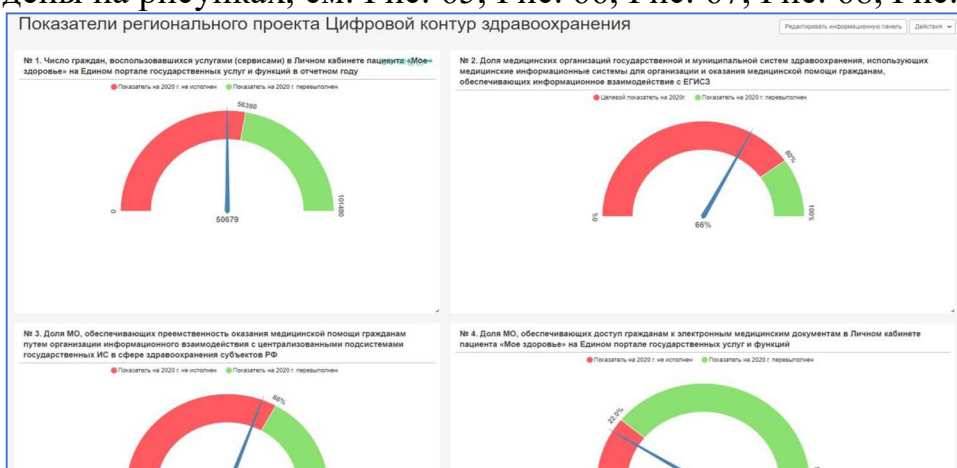


Рисунок 65 Пример 1 информационной панели по показателям Цифрового контура



Рисунок 66 Пример 2 информационной панели по показателям Цифрового контура

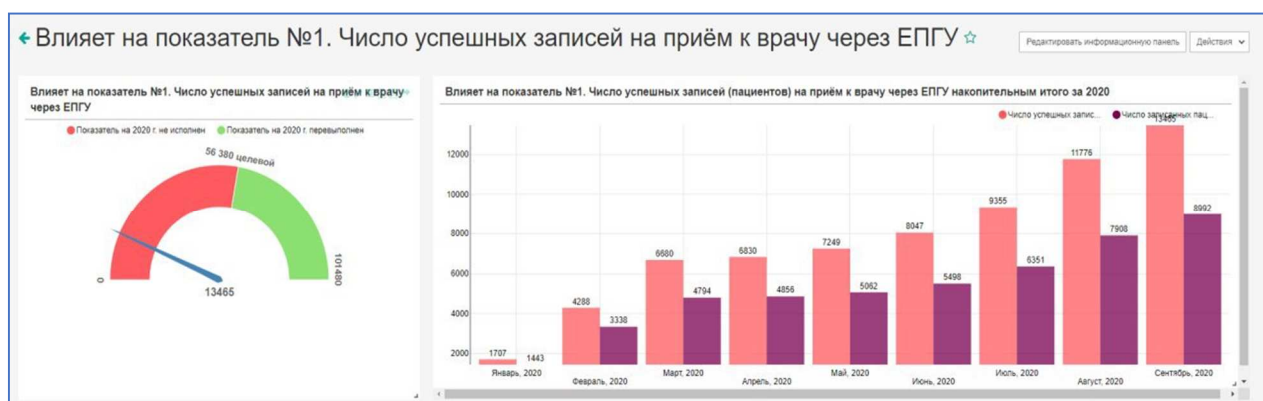


Рисунок 67 Пример 3 информационной панели по показателям Цифрового контура



Рисунок 68 Пример 4 информационной панели по показателям Цифрового контура

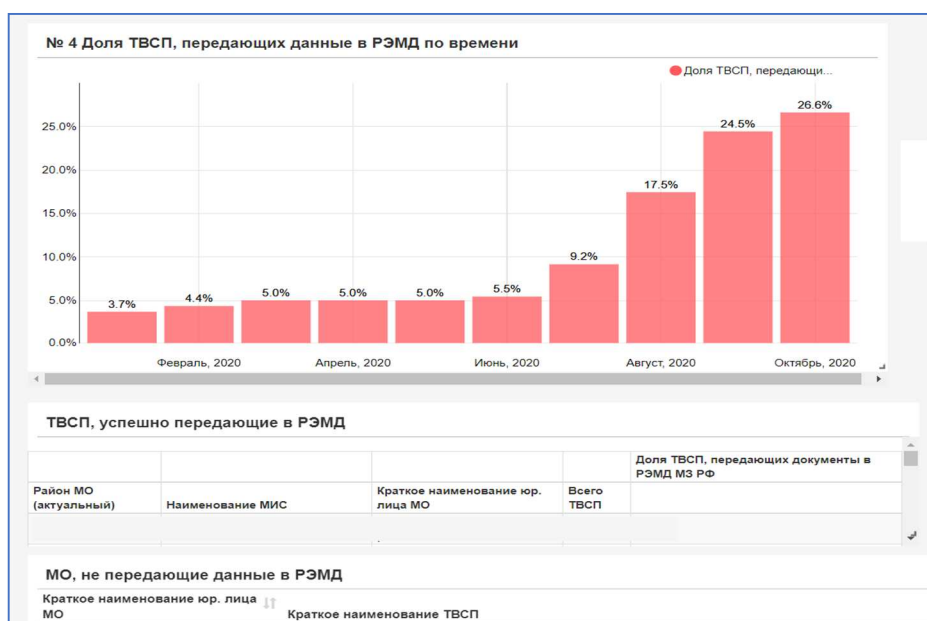


Рисунок 69 Пример 5 информационной панели по показателям Цифрового контура

2.4 Проверка ИАС на требуемую функциональность и качество данных

Тестовый сценарий, тесткейс, test case - формально описанный алгоритм проверки требований к ИС, разработанный для проверки определённых ситуаций работы Системы, т.е. в тесткейсе описываются конкретные действия конкретного пользователя, результат его действий. По результату действий делается вывод о соответствии ИС требованиям. Тесткейс может включать эталонный набор входных и выходных данных и может быть автоматизирован (автотест) [36].

Одна из хороших практик в разработке ПО - составить набор тесткейсов до разработки.

За основу составления тесткейсов следует брать:

1. формализованные требования к ИС, к примеру:
 - 1.1. функциональные требования ТЗ;
 - 1.2. требования ПМИ;
 - 1.3. сценарии использования системы (use case);
 - 1.4. пользовательские истории (user story);
2. “узкие места” информационной системы, которые часто ломаются;
3. распространённые сценарии использования системы пользователями.

Тесткейс обычно образуется на основе алгоритма проверки по определённым шагам и эталонным наборам входных и выходных данных:

1. название тесткейса;
2. проверяемые требования;
3. предусловия, настройки окружения;
4. последовательность тестирования;
5. ожидаемый результат.

Виды тестирования

По степени автоматизации можно выделить следующие виды тестирования [36]:

- ручное тестирование - с полным участием человека в процессе тестирования;
- автоматизированное тестирование - с некоторым участием человека;
- автоматическое тестирование - без участия человека.

По степени формализации можно выделить следующие виды тестирования:

- тестирование по подготовленным сценариям, документам;
- свободное плавание, интуитивное тестирование (англ. ad hoc testing) - использование функционала в максимально широком смысле, без привязки к заготовленным тестам, алгоритмам и формальной логике работы. Тестировщик произвольно нажимает кнопки мыши или клавиатуры, где есть отклик и наблюдает за результатом;

- статическое - проверка без исполнения программного кода. К примеру, тестирование по информационной безопасности через анализ исходного кода;

- динамическое - тестирование с исполнением программного кода.

Проверка качества обработанных данных производится:

- на соответствие исходным данным;
- на внутреннюю структуру и целостность.

Уровни производимого тестирования делятся на:

- Модульное тестирование или юнит-тестирование (англ. unit testing) - процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции, метода, микросервиса или модуля. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, а также облегчает обнаружение и устранение таких ошибок.

- Пример. Запрос к базе данных на проверку перечня таблиц, столбцов и типов данных.
- Плюсы. Просто. Требуется развернуть только БД.
- Минусы. Проверяет только локальный кусок функционала, выполнение всех юнит-тестов совсем не гарантирует, что система будет работать.
- Аналог модульных тестов по ГОСТ - Автономные испытания. Автономные испытания охватывают части АС [49].

- Интеграционное тестирование - проверка того, что компоненты системы корректно работают вместе. Оно охватывает систему в целом. Аналог интеграционного тестирования по ГОСТ - Комплексные испытания, их проводят для групп, взаимосвязанных частей АС или для АС в целом [49].

- Пример. Скрипт обращается к аналитическому хранилищу, строит метрики, обращается к тестовой БД системы-источника, строит проверочные метрики, сравнивает.
- Плюсы. Проверяет работоспособность системы. Приближено к реальному использованию системы пользователями.
- Минусы. Сложность автоматизации тестов. Сложность использования в разработке. В примере требуется развернуть на серверах хранилище данных, etl процессы, аналитическое хранилище, инициировать нулевую загрузку данных с тестовой БД системы-источника, заполнить БД тестовыми данными. Для рутинного тестирования при разработке участка кода использовать сложно.
- Как использовать. Интеграционные тесты удобно использовать для проверки работоспособности нескольких модулей, для подготовки

некоторых формальных документов, для регулярной проверки качества данных.

- Системное, приёмочное тестирование — это тестирование программного обеспечения (ПО), выполняемое на полной, интегрированной системе с целью проверки соответствия системы исходным требованиям. Системное тестирование относят к методам тестирования чёрного ящика, т.е. оно не требует знаний о внутреннем устройстве системы.
 - Аналог системного тестирования в ГОСТ - опытная эксплуатация, приемочные испытания [49]
 - Пример. Пользователь заходит в ИАС под ролью создателя отчётов, строит отчёт, смотрит загрузку данных в отчёт, сверяет данные с подсистемой-источником.
 - Плюсы. Проверяет работоспособность всей системы. Приближено к реальному, комплексному использованию системы пользователями.
 - Минусы. Сложность автоматизации тестов. Сложность использования в разработке. В примере требуется развернуть подсистему-источник, хранилище данных, etl процессы обработки данных, аналитическое хранилище, модуль визуализации, модуль ролевого доступа, отдельный ПК с браузером. Подсистема-источник также будет состоять из набора веб-серверов, серверов БД и т.п. компонент. Для рутинного тестирования при разработке кода использовать это сложно.
 - Как использовать. Интеграционные тесты удобно использовать для финальной проверки функционала, для проведения приёмо-сдаточных испытаний с Заказчиком.

Распространённые ошибки/замечания к юнит тесткейсам

1. Детализировать кейсы данными. Можно прилагать скриншоты. Примеры: как не надо – <https://prnt.sc/qgo249>, кейс по багу с разбивкой на юзерстори и тесткейс – <https://prnt.sc/qqnlq4>.
2. Излишнее повторение шагов кейсов – когда суть кейса начинается с шага 44, а все остальное относится к настройке отчета.
3. Избыточность кейсов. Например, "Пользователь нажал на форму HAVING Система успешно отобразила курсор в форме "HAVING".
4. По ожидаемому результату непонятно, какой результат валиден и вообще, что ожидается. Например, "Система успешно отобразила данные пользователя согласно заданному пользователю типу визуализации".
5. Отсутствуют предусловия, например, какие должны быть права у пользователя, который выполняет тест-кейс. В предусловии кратко указывать путь/действия к проверяемому функционалу/состоянию и/или добавлять ссылку на юзерстори.

Практическое задание

Подготовьте несколько юнит тесткейсов на функционал, подготовленный в сценариях тестирования.

Составьте методику тестирования витрины данных, подготовленной в задании раздела “Проектирование аналитического решения”.

Комплексный пример создания и тестирования витрины данных учёта движения коечного фонда

Исходные данные:

1. Информационно-аналитическая система. (<https://netrika.ru/solution/bi>)
Состав:
 - a. Хранилище данных (data warehouse, DWH).
 - b. ПО, обрабатывающее данные по etl.
 - c. Аналитическое хранилище. Данные представлены в виде плоских витрин данных в аналитических БД.
 - i. Пример по УО. Фактически – 5 таблиц по 200-500 столбцов с десятками, сотнями миллионов записей.
 - ii. Витрина данных лабораторных исследований СПб ежедневно получает более 500 тыс. записей, число накопленных записей приближается к миллиарду.
 - d. Модуль визуализации.
2. Есть подсистема-источник данных. Информационно-логическая структура описана в спецификации API [39].
3. Задача - разработать сценарии проверки данных.
 - a. Случаи проверки:
 - i. Деплой витрины данных с автообновлением в новом регионе.
 - ii. Регулярная проверка качества данных.
 - iii. Проверка при изменениях в коде ИАС - дополнение, изменение полей витрины данных.
 - iv. Проверка при изменениях в коде в ответ на изменение подсистемы-источника данных.
 - b. Требуется проверять все данные. Объёмы записей (десятки, сотни миллионов) не позволяют проверять данные вручную.
 - c. При изменениях в коде (дополнении витрины показателями, изменении) необходима возможность поддерживать изменения в тесткейсах.
 - d. Должна быть простая возможность развернуть тесткейсы в иной реализации витрины (другими типами полей, значениями, справочниками, etl процессами).
4. Решение

- a. Проверка внутренней информационно-логической структуры данных. Пример – витрина данных учёта движения коечного фонда.
- b. Точечная, ручная проверка набора данных (десяток записей) в подсистеме источнике, сервисе терминологии, аналитической витрине.
- c. Повторение основных отчётов, необходимых Заказчику на первичных данных (копии БД системы-источника данных), сравнение с отчётами по витрине данных.

2.5 Сводные панели руководителей как инструмент управления на основе данных системой здравоохранения

В 2016 г. в российском государственном управлении стартовал очередной виток инициатив, направленных на повышение эффективности: внедрение проектных методов на основе «приоритетных проектов». Важным элементом программно-целевого метода управления и приоритетных проектов стало введение целевых показателей проектов и их целевых значений по годам. Показатели и значения показателей должны объективно и однозначно определять эффективность и результативность реализации проекта, достижение его целей.

Для того, чтобы обеспечить успешную реализацию приоритетных проектов на уровне субъекта Российской Федерации и достижение целевых значений показателей, крайне важно провести декомпозицию целевых показателей на учреждения, сформировать на их основе перечень мониторируемых оперативных показателей и обеспечить руководителей учреждений, участвующих в проекте, современными и удобными средствами оперативного измерения и мониторинга изменений показателей. Также необходимо создать систему стимулирования как за выполнение проекта в целом, так и за достижение промежуточных значений оперативных показателей выполнения проекта. Это обеспечит для руководителей необходимое включение в проект, понятные правила игры и возможность влиять на достижение целей проекта.

Рассмотрим пример того, какие показатели, в том числе для оперативного управления, были введены при реализации приоритетного проекта Санкт-Петербурга «Электронное здравоохранение» в 2018-2020 годах. Была поставлена следующая задача: оперативно обеспечивать руководителей здравоохранения актуальной и достоверной информацией по оперативным ключевым показателям проекта, построенной автоматически из оперативных данных медицинской организации и региональной информационной системы здравоохранения по вопросам цифрового развития процессов и оказания медицинской помощи в Санкт-Петербурге в дополнение к имеющимся типовым отчётам.

Система должна была охватывать специалистов медицинских служб, сотрудников Медицинского информационно-аналитического центра Санкт-Петербурга (далее – МИАЦ) и главных врачей медицинских учреждений, то есть свыше 2000 пользователей и более чем 300 медицинских организаций города.

Предпосылками создания такой системы являлись:

- трудоёмкий ручной сбор и агрегация данных о деятельности свыше 300 МО СПб;
- потери данных на этапах сбора и агрегации информации;
- затруднённый поиск правильных цифр на аппаратных совещаниях с директорами городских МО – руководство города в сфере здравоохранения использовало свои данные, которые не были заранее

доступны для руководителей медицинских учреждений города, а последние приходили со своей отличающейся информацией.

Поэтому требовалось создать единое информационное пространство для руководителей здравоохранения и обеспечить принципиально другой подход к работе с управленческой информацией.

Практическая польза от создания системы заключалась в создании:

- единой системы контроля ключевых показателей деятельности для сотрудников МИАЦ, Комитета по Здравоохранению, главврачей, медицинских специалистов,
- подтягивании отстающих благодаря визуальным рейтингам районов и медицинских организаций,
- диалоге с пациентами посредством интерактивной графики на внешних порталах медицинской организации.

В качестве источника данных были использованы:

- сотни баз данных медицинских и лабораторных информационных систем посредством интеграционной шины Региональной государственной информационной системы в сфере здравоохранения Санкт-Петербурга (РЕГИЗ),
- хранилище данных со статистикой и нормативами для контроля,
- ручные выгрузки в формате csv и excel.

Аналитические панели обеспечивали визуализацию показателей по направлениям: запись-приём пациентов, причины обращений пациентов, диспансеризация, оборачиваемость койко-фонда, финансовые показатели, хозяйственная деятельность и др.

Данные собирались от учреждений здравоохранения на основании Распоряжения Комитета здравоохранения Санкт-Петербурга о создании и ведении «Электронной медицинской карты петербуржца» №88-р от 21.02.2018.

Примеры инфопанелей приведены на Рис. 70, Рис. 71, Рис. 72, Рис. 73.

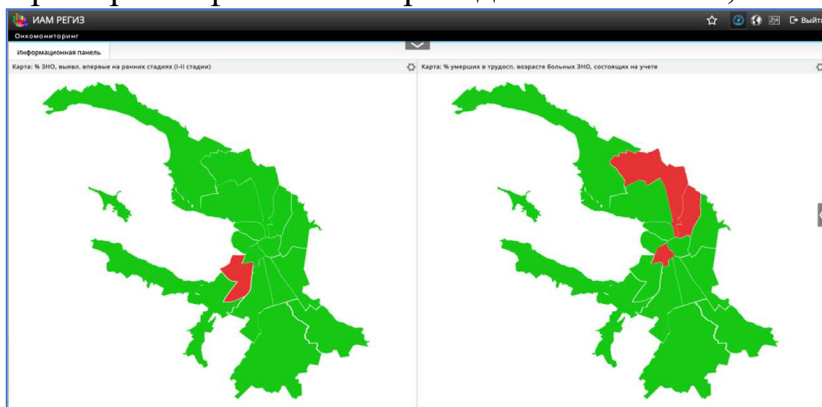


Рисунок 70 Инфопанель с картографическими элементами

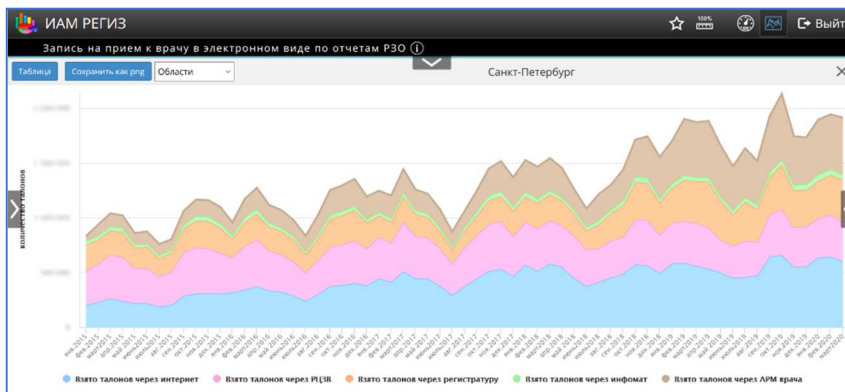


Рисунок 71 Инфопанель с графиками изменения показателей

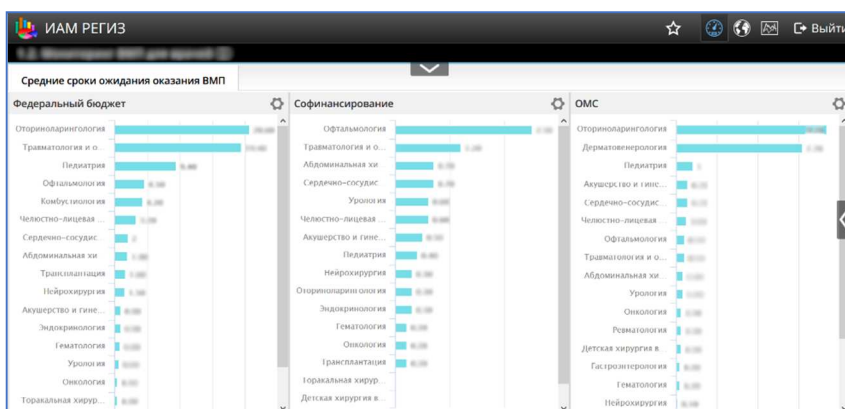


Рисунок 72 Инфопанель с рейтингованием по убыванию показателей

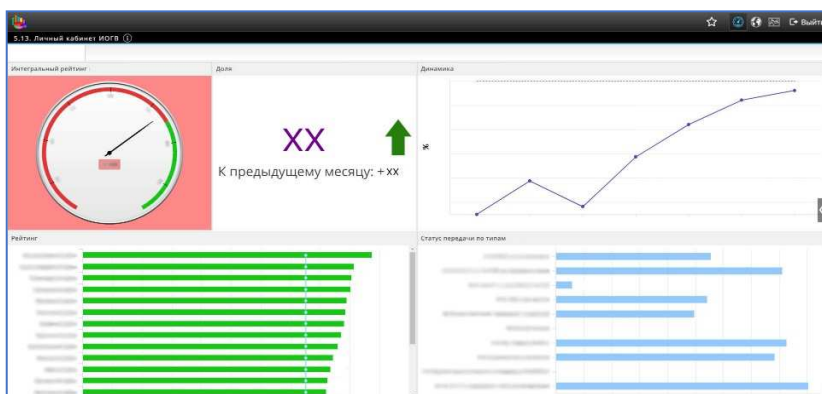


Рисунок 73 Дашборд руководителя медицинской организации с разными типами графических элементов

Был реализован **сервис личных кабинетов руководителей** - персональные интерактивные панели для ИОГВ, администраций районов, главврачей.

Оператором системы стал МИАЦ, который обеспечил организационно-методическую поддержку медицинских организаций, мониторинг исполнения подключения учреждений здравоохранения к ГИС РЕГИЗ и передаче данных в ЭМК, формирование отчётов и рейтингов учреждений, представление управленческой информации в Комитет здравоохранения для

принятия организационно-управленческих решений, стимулирования и наказания руководителей учреждений [43].

Также были сформированы «живые графики» показателей здравоохранения Санкт-Петербурга, см. Рис. 74, представлявшие собой интерактивные графики для граждан Санкт-Петербурга по данным о записи на приём к врачу, сроках ожидания медпомощи, диспансеризации на сайте МИАЦ СПб [50]: <http://spbmiac.ru/specialistam/zhivye-grafiki/>



Рисунок 74 «Живые графики» показателей здравоохранения Санкт-Петербурга

МИАЦ формировал рейтинги медицинских организаций - рейтинг подключения медицинских организаций к ГИС РЕГИЗ и передаче данных в ЭМК петербуржца и рейтинг полноты и качества передаваемой в ЭМК петербуржца информации [43].

Рейтинги доступны на сайте МИАЦ СПб [51]: <https://spbmiac.ru/ehlektronnoe-zdravookhranenie/rejtingi-e-zdravookhraneniya/rejtingi-mo-emk-peterburzhca/>

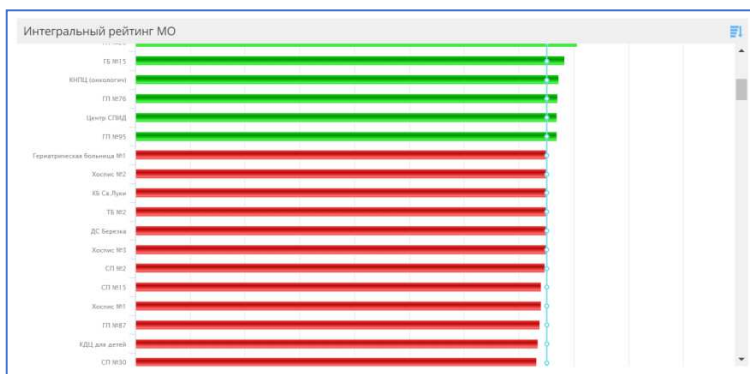


Рисунок 75 Интегральный рейтинг подключения медицинских организаций к ГИС РЕГИЗ и передаче данных в ЭМК петербуржца

Интегральный рейтинг медицинских организаций по полноте и качеству ведения ЭМК петербуржца приведен на Рис. 75.

2.6 Подходы к оценке цифровой зрелости систем социальной сферы. Индексы СИМИС и СЕЗАМ

Формирование дашбордов («приборных панелей») для руководителей по оперативным показателям, позволяющее выстроить таким образом прозрачную систему управления цифровизацией учреждений, взаимосвязано с задачей интегральной оценки цифровой зрелости процессов в учреждении. То же требуется и при интегральной оценке усилий руководителей по достижению целевых значений показателей ведущихся проектов, по оценке улучшения массово оказываемых гражданам услуг, таких как запись на прием к врачу. Все это задачи оценки реализации проектов развития.

Актуальность темы связана с ведущейся цифровой трансформацией социальной сферы экономики России, где обязательства государства обеспечены, главным образом, государственными учреждениями. Для реализации управляемого процесса цифровой трансформации государственных учреждений социальной сферы используется проектный подход, одним из элементов которого является определение целей и значений целевых показателей, утверждаемых в соответствующих проектных документах [52]. В июле 2020 был подписан Указ Президента Российской Федерации «О национальных целях развития Российской Федерации на период до 2030 года» [53], который определил «цифровую трансформацию» как одну из пяти национальных целей развития России. Целевым показателем установлено достижение «цифровой зрелости» ключевых отраслей экономики и социальной сферы, в том числе здравоохранения и образования. Для этого требуется разработка систем оценок «цифровой зрелости» социальной сферы в форме системы интегральных индексов, характеризующих как достигнутый функциональный уровень зрелости цифровой трансформации, так и объемные показатели внедрения цифровых технологий в структурных элементах организаций.

Обзор подходов к оценке уровня информатизации социальной сферы

Подходы к оценке уровня информатизации в России, применяемые в здравоохранении как важнейшей отрасли социальной сферы, систематически не рассматриваются в научной литературе - есть лишь незначительное количество публикаций. Здравоохранение характеризуется большим количеством типовых государственных учреждений здравоохранения, управление информатизацией которых определяется рядом федеральных и региональных требований. Подход по оценке уровня информатизации медицинских организаций и региона, предложенный в работах [54], [55], требует адаптации к принятому позднее ряду требований в условиях ведущихся федеральных проектов цифровой трансформации здравоохранения. Кроме этого, предложенный результат-

ориентированный подход учитывает функциональные показатели развития информатизации, но не позволяет учитывать объемные показатели внедрения, например, поэтапное включение подразделений медицинских организаций и врачей в процесс информатизации.

Уровень развития региональных систем записи на прием к врачу в субъектах России оценивается Минцифры России с помощью методики [56], которая относит такие системы к одному из четырех классов по фиксированной балльной системе от 0 до 3 баллов. Данная методика не позволяет оценивать текущее развитие систем записи на прием к врачу в условиях многопараметрической задачи и не позволяет построить систему управления развитием. Она применяется в целях усредненной оценки большого количества регионов в рамках оценки результатов федерального проекта создания единого цифрового контура здравоохранения.

Наиболее продвинутый подход применен в работе [57], где на основе систематизации международного и собственного практического опыта и «Digital Imaging Adoption Model (DIAM)» разработан инструмент для бенчмаркинга уровня цифровизации отделений лучевой диагностики. Инструмент включает: эталонную «Модель зрелости цифровизации медицинской визуализации «Медвиз»», базовый методический документ (унифицированную пятиуровневую стратегию цифровизации), инструмент оценки степени цифровизации, структурированный набор рекомендаций. В процессе бенчмаркинга производится классификация отделения лучевой диагностики в соответствии с авторской моделью «Медвиз». Однако метод применим к оценке одного подразделения и не рассматривает оценку субъекта Российской Федерации в целом.

Имеется большой спектр зарубежных публикаций по рассматриваемой теме. Так, в целях поддержки стратегии системы здравоохранения по продвижению цифровой трансформации в глобальной некоммерческой организации Healthcare Information and Management Systems Society, Inc. (HIMSS) в 2020 году был разработан инструмент для измерения и документирования прогресса в создании цифровой системы здравоохранения - Цифровой индикатор здоровья. HIMSS создала портфель из семи моделей цифровой зрелости, которые на сегодняшний день были приняты организациями здравоохранения в 50 странах. Набор моделей зрелости предлагает стратегию измерения для оценки конкретного аспекта цифровой зрелости, чтобы определять стратегические решения по продвижению цифровой зрелости. Инструменты зрелости используются на рынке глобальной системы здравоохранения для измерения цифровой зрелости ключевых областей, в том числе, цифровых медицинских записей (EMRAM, o-EMRAM), преемственности оказания медицинских услуг (CCMM), и ряда других. Все семь моделей измеряют цифровую зрелость с помощью ряда измерительных индикаторов, организованных по уровням зрелости от 0 до 7 [58].

Предлагаемый подход к оценке цифровой зрелости

Предлагается разрабатывать эталонные сервисные (функциональные) модели для отдельных процессов социальной сферы с последующей оценкой уровня реализации таких моделей в государственных информационных системах, используемых в учреждениях или субъекте России в целом. Кроме этого, предлагается оценивать объем внедрения или использования, основываясь на определении доли учреждений и соответствующих специалистов, участвующих в процессах, прошедших цифровую трансформацию. Также имеется возможность учесть для отдельных специалистов долю учитываемых ими в системе информационных операций. Для вычисления интегрального индекса на основе предложенных показателей предлагается использовать функции, гладко зависящие от изменения показателей. Это обеспечит возможность оперативного отслеживания динамики изменений интегрального индекса, в том числе для управления процессами информатизации, что особенно важно при оценке уровня цифровой зрелости субъекта РФ при большом количестве учреждений. Таким свойством обладает, например, среднее геометрическое из предложенных показателей, которое обеспечит большую чувствительность интегральной функции при изменении отдельного показателя, особенно в случае, если показатель имеет небольшие значения при существенно больших значениях прочих показателей, в отличие от традиционно применяемой формулы среднего арифметического с весовыми коэффициентами. Среднее геометрическое было выбрано на основе практического опыта расчета индекса «содержательного использования медицинских информационных систем», предложенного в 2015 году в СПб МИАЦ и утвержденного Председателем Комитета по здравоохранению Санкт-Петербурга, в котором в качестве основного слагаемого в расчете интегрального индекса использовалось произведение индикатора объема использования МИС на функциональный индикатор [59].

Таким образом, интегральный индекс зависит от двух показателей – от бальной оценки реализации сервисов эталонной модели (в формуле – отношение «ОценкаФункций» к максимальной оценке «МаксОценкаФункций») и от доли организаций или их специалистов, реализующих установленную модель цифровой трансформации (в формуле – «ОценкаДолиОрганизаций»), как среднее геометрическое этих величин:

$$\begin{array}{l} \text{Интегральный индекс цифровой зрелости} \\ = \sqrt{\frac{\text{ОценкаФункций}}{\text{МаксОценкаФункций}} \times \text{ОценкаДолиОрганизаций}} \end{array}$$

Оценка цифровой зрелости сервиса записи на прием к врачу

Рассмотрим для иллюстрации применения предложенного подхода один из

автоматизируемых процессов в рамках проекта создания единого цифрового контура здравоохранения - запись на прием к врачу. Для использования на уровне региона в целях управления развитием систем записи к врачу была предложена Сервисная модель региональной системы записи к врачу СЕЗАМ [60], состоящая из 25 эталонных сервисов, четырехуровневой классификации уровней развития, интегрального функционального индекса и 31 показателя функционирования. По аналогии с предложенным в статье подходом можно разработать эталонные сервисные модели для других подсистем региональной информационной системы в сфере здравоохранения. Интегральный индекс позволяет планировать развитие региональной системы записи к врачу, получать объективное сравнение развития сервисов своей системы с другими регионами.

Пример расчета СЕЗАМ по Санкт-Петербургу.

На Рис.76 приведен пример динамики изменения индекса СЕЗАМ по Санкт-Петербургу. Три графика характеризуют рост уровня цифровой зрелости системы записи на прием к врачу в Санкт-Петербурге в 2010-2020 годах (данные 2010-2014 годов смоделированы, 2015-2020 – реальные данные мониторинга), с применением различных интегральных функций:

график СЕЗАМ-СПб – рассчитывается как произведение ОценкаФункций/Макс и ОценкаДолиМО (такая формула используется в Санкт-Петербурге для расчета аналогичного Индекса СИ-МИС содержательного использования медицинских информационных систем [59]), отметим, что видимый ускоряющийся рост индекса является лишь следствием примененной функции и не характеризует истинный относительно равномерный рост цифровой зрелости;

график СЕЗАМ-Геом – рассчитывается как среднее геометрическое, а в графике СЕЗАМ-Ариф – как среднее арифметическое ОценкаФункций/Макс и ОценкаДолиМО.

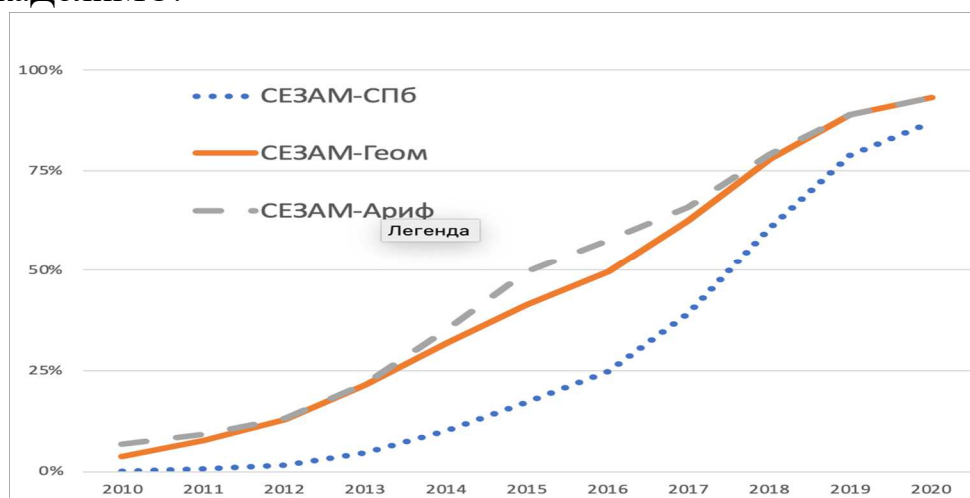


Рисунок 76 Варианты расчета Индекса СЕЗАМ

Вариант формулы индекса СЕЗАМ-Геом, а значит, и предложенный Интегральный индекс цифровой зрелости, наиболее адекватен реальной картине развития цифровой трансформации сервиса и предложенный подход можно рекомендовать к применению для расчетов цифровой зрелости социальной сферы.

На Рис. 77 приведен график роста Индекса СИ-МИС (содержательного использования МИС), который представляет собой оценку цифровой зрелости внедрения МИС в Санкт-Петербурге.

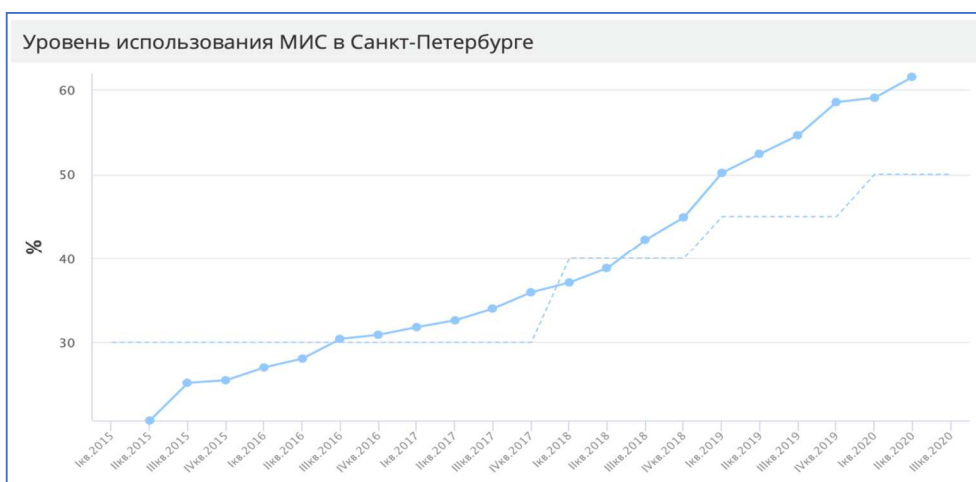


Рисунок 77 Уровень использования МИС в Санкт-Петербурге - Индекс СЕЗАМ

ПРИЛОЖЕНИЕ

Перечень примеров на внешних носителях

<i>Пример на внешних носителях 1. Паспорт цифрового сервиса Ситуационного Центра Губернатора</i>	86
<i>Пример на внешних носителях 2. Жизненный цикл цифрового сервиса Ситуационного Центра Губернатора</i>	86
<i>Пример на внешних носителях 3. Копилка идей для Ситуационного Центра Губернатора</i>	86
<i>Пример на внешних носителях 4. Информационно-логическая модель витрины данных ОДЛИ;</i>	90
<i>Пример на внешних носителях 5. Информационно-логическая модель витрины данных ИЭМК.</i>	90
<i>Пример на внешних носителях 6. API Учёт движения коечного фонда.</i>	96
<i>Пример на внешних носителях 7. Витрина данных Учёт движения коечного фонда.</i>	96
<i>Пример на внешних носителях 8. API Обмена инструментальными исследованиями.</i>	96
<i>Пример на внешних носителях 9. Часть проекта витрины данных ОДИИ.</i>	96
<i>Пример на внешних носителях 10. ER диаграмма сборки и анализа на размножение данных витрине ОДИИ.</i>	96

Перечень таблиц

<i>Таблица 1. Сравнение характеристик традиционных баз данных и технологий Big data</i>	13
<i>Таблица 2. Основные ограничения четырех методологических подходов в мониторинге политики (Источник: W.Dunn, 2004. p. 284)</i>	19
<i>Таблица 3 Матрица сопряженности в программе Блокнот/Notepad</i>	46
<i>Таблица 4. Список универсальных парсеров.</i>	65
<i>Таблица 5 Список универсальных парсеров для социальных сетей</i>	74
<i>Таблица 6 Варианты описания функционала ИАС</i>	79
<i>Таблица 7 Сценарий использования Drag&Drop показателей (метрик)</i>	82

Перечень рисунков

<i>Рисунок 1 Окно для работы в RStudio</i>	28
<i>Рисунок 2 Визуализация медианного значения использования источников альтернативной энергии.</i>	32

<i>Рисунок 3 Сопоставление среднего и медианного значения количества используемой альтернативной энергии.</i>	<i>33</i>
<i>Рисунок 4 Сопоставление среднего и медианного значения количества выбросов CO2.</i>	<i>34</i>
<i>Рисунок 5 Диаграмма Кливленда для независимой переменной age</i>	<i>35</i>
<i>Рисунок 6 Диаграмма Кливленда для зависимой переменной bwt.....</i>	<i>36</i>
<i>Рисунок 7 Квантильный график остатков.....</i>	<i>37</i>
<i>Рисунок 8 График расстояний Кука.....</i>	<i>38</i>
<i>Рисунок 9 График остатков в зависимости от предсказанных значений зависимой переменной.....</i>	<i>39</i>
<i>Рисунок 10 График остатков для непрерывного предиктора age</i>	<i>39</i>
<i>Рисунок 11 График остатков для дискретного предиктора smoke</i>	<i>40</i>
<i>Рисунок 12 График предсказанных значений зависимой переменной с указанием исходных значений</i>	<i>42</i>
<i>Рисунок 13 Визуализация уровней власти в стране Хэппилэнд.</i>	<i>48</i>
<i>Рисунок 14 Отношения между акторами на федеральном уровне власти в стране Хэппилэнд.</i>	<i>50</i>
<i>Рисунок 15 Отношения между уровнями власти в стране Хэппилэнд.....</i>	<i>50</i>
<i>Рисунок 16 Отношения между акторами внутри федерального уровня власти с другими уровнями власти.....</i>	<i>51</i>
<i>Рисунок 17 Пример интерфейса программы для сбора данных.....</i>	<i>59</i>
<i>Рисунок 18 Общая схема работы парсера.....</i>	<i>63</i>
<i>Рисунок 19 Постраничная пагинация.....</i>	<i>69</i>
<i>Рисунок 20 Пример генерации ссылок</i>	<i>69</i>
<i>Рисунок 21 Результат генерации ссылок.....</i>	<i>69</i>
<i>Рисунок 22 Границы парсинга.....</i>	<i>70</i>
<i>Рисунок 23 Как задать начало и конец парсинга</i>	<i>71</i>
<i>Рисунок 24 Настройка области</i>	<i>71</i>
<i>Рисунок 25 Настройка адреса.....</i>	<i>72</i>
<i>Рисунок 26 Настройка формата хранения данных</i>	<i>72</i>
<i>Рисунок 27 Границы парсинга.....</i>	<i>73</i>
<i>Рисунок 28 Доклад о проблемах в организации лабораторных исследований сотрудника, ведущего учёт инфекционных болезней.....</i>	<i>77</i>
<i>Рисунок 29 Показатели и метрики в пользовательском интерфейсе superset</i>	<i>81</i>
<i>Рисунок 30 Скриншот, описывающий требования к интерфейсу</i>	<i>85</i>
<i>Рисунок 31 Макет Аналитической панели по мониторингу карт маршрутизации пациентов с подозрением на злокачественные новообразования</i>	<i>86</i>

<i>Рисунок 32 Обозначения нотации «воронья лапка» ER диаграмм</i>	<i>96</i>
<i>Рисунок 33 Скриншот установки клиента DBeaver с официального сайта.....</i>	<i>99</i>
<i>Рисунок 34 Скриншот DBeaver с перечнем подключенных БД.....</i>	<i>99</i>
<i>Рисунок 35 Выбор типа нового соединения с БД.....</i>	<i>99</i>
<i>Рисунок 36 Настройки доступа к БД.....</i>	<i>100</i>
<i>Рисунок 37 Командная строка</i>	<i>100</i>
<i>Рисунок 38 Пример доступного ресурса</i>	<i>100</i>
<i>Рисунок 39 Пример недоступного ресурса</i>	<i>100</i>
<i>Рисунок 40 Базы данных DBeaver после подключения БД.....</i>	<i>101</i>
<i>Рисунок 41 Таблицы, их описание и данные БД</i>	<i>101</i>
<i>Рисунок 42 Описание перечня столбцов, их свойств, названий, уникальных ключей</i>	<i>102</i>
<i>Рисунок 43 Пример раздела данных, содержащихся в таблице.....</i>	<i>102</i>
<i>Рисунок 44 Пример раздела данных, содержащихся в таблице.....</i>	<i>103</i>
<i>Рисунок 45 Пример раздела данных, содержащихся в таблице.....</i>	<i>103</i>
<i>Рисунок 46 Пример раздела данных, содержащихся в таблице.....</i>	<i>103</i>
<i>Рисунок 47 Пример обращения к БД.....</i>	<i>104</i>
<i>Рисунок 48 Примеры результатов выполнения запросов</i>	<i>104</i>
<i>Рисунок 49 Комментарии в коде о решаемой бизнес-задаче.....</i>	<i>104</i>
<i>Рисунок 50 Пример различных комментариев в SQL запросе</i>	<i>104</i>
<i>Рисунок 51 Пример запроса с непересекающимися параметрами.....</i>	<i>110</i>
<i>Рисунок 52 Пример SQL-запроса 1 и результата</i>	<i>111</i>
<i>Рисунок 53 Пример SQL-запроса 2 и результата</i>	<i>112</i>
<i>Рисунок 54 Пример SQL-запроса 3 и результата</i>	<i>113</i>
<i>Рисунок 55 Пример SQL-запроса 4 и результата</i>	<i>114</i>
<i>Рисунок 56 Пример SQL-запроса 5 и результата</i>	<i>114</i>
<i>Рисунок 57 Пример SQL-запроса 6 и результата</i>	<i>115</i>
<i>Рисунок 58 Пример SQL-запроса 7 и результата</i>	<i>115</i>
<i>Рисунок 59 Основные виды визуализаций данных.....</i>	<i>117</i>
<i>Рисунок 60 Инфопанель по переданным лабораторным исследованиям на определение SARS coronavirus 2.....</i>	<i>118</i>
<i>Рисунок 61 Инфопанель по количеству результатов SARS coronavirus 2 РНК ..</i>	<i>119</i>
<i>Рисунок 62 Дашборд по показателям записи на приём к врачу.....</i>	<i>119</i>
<i>Рисунок 63 Пример отчёта с визуальным выделением особо важных данных...120</i>	
<i>Рисунок 64 Пример отчёта с искусственной группировкой.....</i>	<i>120</i>

<i>Рисунок 65 Пример 1 информационной панели по показателям Цифрового контура</i>	<i>122</i>
<i>Рисунок 66 Пример 2 информационной панели по показателям Цифрового контура</i>	<i>122</i>
<i>Рисунок 67 Пример 3 информационной панели по показателям Цифрового контура</i>	<i>123</i>
<i>Рисунок 68 Пример 4 информационной панели по показателям Цифрового контура</i>	<i>123</i>
<i>Рисунок 69 Пример 5 информационной панели по показателям Цифрового контура</i>	<i>123</i>
<i>Рисунок 70 Инфопанель с картографическими элементами</i>	<i>130</i>
<i>Рисунок 71 Инфопанель с графиками изменения показателей.....</i>	<i>131</i>
<i>Рисунок 72 Инфопанель с рейтингованием по убыванию показателей</i>	<i>131</i>
<i>Рисунок 73 Дашборд руководителя медицинской организации с разными типами графических элементов.....</i>	<i>131</i>
<i>Рисунок 74 «Живые графики» показателей здравоохранения Санкт-Петербурга</i>	<i>132</i>
<i>Рисунок 75 Интегральный рейтинг подключения медицинских организаций к ГИС РЕГИЗ и передаче данных в ЭМК петербуржца.....</i>	<i>133</i>
<i>Рисунок 76 Варианты расчета Индекса СЕЗАМ</i>	<i>137</i>
<i>Рисунок 77 Уровень использования МИС в Санкт-Петербурге - Индекс СЕЗАМ</i>	<i>138</i>

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Ермолаев О.Ю. Математическая статистика для психологов: учебник. М.: Московский психолого-социальный институт: Флинта, 2003. - 335 с.
2. Babbie E. The Basics of Social Research. Wadsworth: Cengage Learning, 2008. – 552 p.
3. Квале С. Исследовательское интервью. М.: Смысл, 2003. - 301 с.
4. Козлов М.В. (2014). Планирование экологических исследований: теория и практические рекомендации. М.: КМК. - 169 с.
5. Сергеева И.И., Чекулина Т.А., Тимофеева С.А. Статистика /И.И.Сергеева, Т.А. Чекулина, С.А. Тимофеева. М.: Инфра-М, 2016. - 227 с.
6. Ноэль Э. Массовые опросы: введение в методику демоскопии. М.: Прогресс, 1978. - 382 с.
7. Кун Т. Структура научных революций. М.: Издательство АСТ, 2003. - 605 с.
8. Масалков И.К., Семина М.В. Стратегия кейс стади: методология исследования и преподавания. М.: Академический Пропект; Альма Матер, 2011. - 443 с.
9. Сартори Дж. Искажение концептов в сравнительной политологии // Полис. Политические исследования. - 2003. - № 3. <https://doi.org/10.17976/jpps/2003.03.07>
10. Goktug M., Ivanova N. Methods Taught in Public Policy Programs: are Quantitative Methods still Prelevant? // Journal of Public Affairs Education. Vol. 16, no 2 (Spring 2010). - P. 255-277.
11. Dunn W. Public policy analysis. 3d ed. Pearson: Prentice Hall, 2004. - 510 p.
- 12.Брайман А., Белл Э. Методы социальных исследований. Группы, организации, бизнес / Пер. с англ. Харьков: Изд-во Гуманитарный центр, 2012. -774 с.
- 13.Поппер К. Логика научного исследования. М.: Республика, 2005. - 447 с.
- 14.Добренёв В.И., Кравченко А.И. Методы социологического исследования: учебник. М.: Инфра-М, 2016. - 767 с.
- 15.Козлов М.В. (2014). Планирование экологических исследований: теория и практические рекомендации. М.: КМК. - 169 с.
- 16.Бусыгина Н.П. Методология качественных исследований в психологии: учеб. пособие. М.: Инфра-М, 2014. - 304 с.
- 17.Бусыгина Н.П. Качественные и количественные методы исследования в психологии: учебник для бакалавриата и магистратуры. М.: Издательство

- Юрайт, 2017. - 423 с.
18. Наследов А.Д. Математические методы психологических исследований: анализ и интерпретация данных: учеб. пособие / А.Д. Наследов. СПб: Речь, 2004. - 384 с.
 19. R Core Team. (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. [Электронный ресурс]. - URL: <http://www.R-project.org/> (дата обращения 04.11.2020)
 20. RStudio (2018). RStudio: Integrated development environment for R (Version 1.1.453). Boston, MA. [Электронный ресурс]. - URL: <http://www.rstudio.org/> (дата обращения 14.11.2020)
 21. Кабаков Р.И. R в действии. Анализ и визуализация данных на языке R. М.: ДМК Пресс, 2014. - 572 с.
 22. Hand D.J. et al. A Handbook of Small Data Sets, N.Y: Chapman and Hall, 1994. - 392 p.
 23. Лонг Дж., Титор П. Р. Книга рецептов: Проверенные рецепты для статистики, анализа и визуализации. М.: ДМК, 2020. - 503 с.
 24. Низаметдинов Ш.У., Румянцев В.П. Анализ данных: учеб. Пособие. М.: НИЯУ МИФИ, 2012. - 288 с.
 25. Faraway J. Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models. London: CRC Press, 2016. - 399 p.
 26. Сморгунов Л.В., Шерстобитов А.С. Политические сети: Теория и методы анализа: Учебник для студентов вузов / Л. В. Сморгунов, А. С. Шерстобитов. М.: Издательство «Аспект Пресс», 2014. - 320 с.
 27. Мирошниченко И.В. Сетевой подход в политических исследованиях: содержание и направления развития // Человек. Сообщество. Управление. - 2013. - № 3.
 28. Rhodes R. Policy Network Analysis. In M. Moran, M. Rein and R. E. Goodin (Eds.) The Oxford Handbook of Public Policy. Oxford: Oxford University Press), 2006. - P. 423-45.
 29. Rhodes R. Understanding Governance: Ten Years On // Organization Studies, 2007. Vol. 28. no 08. - P.1-22.
 30. De Nooy W., Mrvar A., Batagelj V. Exploratory Social Network Analysis with Pajek: Revised and Expanded Edition for Updated Software. Third Edition. Cambridge: Cambridge University Press, 2018. - 455 p.
 31. Ядов В.А. Стратегия социологического исследования: описание, объяснение, понимание социальной реальности: учеб. пособие. М.:

Издательство «Омега-Л», 2009. - 567 с.

32. Knoke D., Pappi F., Broadbent J., Tsujinaka Y. Comparing Policy Networks. Labor Politics in the U.S., Germany and Japan. New York: Cambridge University Press, 1996. - 288 p.
33. Использование «VOSviewer» для библиометрического анализа [Электронный ресурс]. - URL: <https://news.egov.itmo.ru/research-11-23.html> (дата обращения 18.12.2020)
34. Researchers use Twitter to track the flu in real time [Электронный ресурс]. - URL: <http://news.northeastern.edu/2017/05/05/researchers-use-twitter-to-track-the-flu-in-real-time/> (дата обращения 20.12.2020)
35. Беген П.Н., Низомутдинов Б.А., Тропников А.С. Классификатор объектов городского хозяйства для данных из социальных сетей // Научный сервис в сети Интернет: труды XXII Всероссийской научной конференции (21-25 сентября 2020 г., онлайн). - М.: ИПМ им. М.В.Келдыша, 2020. - С. 101-115. <https://doi.org/10.20948/abrau-2020-41> [Электронный ресурс]. - URL: <https://keldysh.ru/abrau/2020/theses/41.pdf> (дата обращения 21.12.2020)
36. Информационная сеть «Техэксперт» [Электронный ресурс]. - URL: <http://docs.cntd.ru/document/1200139532> (дата обращения 20.12.2020)
37. ClickHouse: очень быстро и очень удобно [Электронный ресурс]. - URL: <https://habr.com/ru/post/322724/> (дата обращения 20.12.2020)
38. Алистер Коберн. Современные методы описания функциональных требований к системам, издательство «Лори», 2014 г.
39. Регламенты информационного взаимодействия с продуктами, разрабатываемыми компанией «Нетрика» [Электронный ресурс]. - URL: <http://api.n3zdrav.ru/> (дата обращения 20.12.2020)
40. Общие требования к информатизации с сайта СПб МИАЦ [Электронный ресурс]. - URL: <http://spbmiac.ru/ehlektronnoe-zdravookhranenie/normativno-pravovye-akty/obshhie-trebovaniya-k-informatizacii/> (дата обращения 20.12.2020)
41. Современное состояние НПА по информатизации здравоохранения. Блог компании «К-МИС» [Электронный ресурс]. - URL: <https://www.kmis.ru/blog/tekushchee-sostoianie-normativnogo-regulirovaniia-informatizatsii-zdravookhraneniia> (дата обращения 20.12.2020)
42. Федеральные информационные системы в сфере здравоохранения. Блог компании «К-МИС» [Электронный ресурс]. - URL: <https://www.kmis.ru/blog/federalnye-informatsionnye-sistemy-v-sfere-zdravookhraneniia> (дата обращения 20.12.2020)

43. Васин А.Г., Свиркин М.В., Балыкина Ю.Е., Акулин И.М. Развитие системы здравоохранения России: анализ внедрения электронной медицинской карты на примере Санкт-Петербурга // Дискуссия. — 2019. — Вып. 95. — С. 48—60 [Электронный ресурс]. - URL: <https://cyberleninka.ru/article/n/razvitie-sistemy-zdravoohraneniya-rossii-analiz-vnedreniya-elektronnoy-meditsinskoy-karty-na-primere-sankt-peterburga> (дата обращения 20.12.2020)
44. Создаем OLAP куб [Электронный ресурс]. - URL: <https://habr.com/ru/post/67272/> (дата обращения 20.12.2020)
45. Введение в Data Vault [Электронный ресурс]. - URL: <https://habr.com/ru/post/348188/> (дата обращения 20.12.2020)
46. Что такое ER-диаграмма и как ее создать? [Электронный ресурс]. - URL: <https://www.lucidchart.com/pages/ru/erd-диаграмма> (дата обращения 20.12.2020)
47. Документация СУБД Clickhouse, Yandex LLC [Электронный ресурс]. - URL: <https://clickhouse.tech/docs/ru/> (дата обращения 20.12.2020)
48. Датайога [Электронный ресурс]. - URL: <https://datayoga.ru/> (дата обращения 20.12.2020)
49. ГОСТ 34.* Стандарты ИТ технологий [Электронный ресурс]. - URL: http://www.rugost.com/index.php?option=com_content&view=category&id=22&Itemid=53 (дата обращения 20.12.2020)
50. «Живые графики» показателей здравоохранения Санкт-Петербурга [Электронный ресурс]. - URL: <http://spbmiac.ru/specialistam/zhivye-grafiki/> (дата обращения 20.12.2020)
51. Рейтинги медицинских организаций - рейтинг подключения медицинских организаций к ГИС РЕГИЗ и передаче данных в ЭМК петербуржца и рейтинг полноты и качества передаваемой в ЭМК петербуржца информации [Электронный ресурс]. - URL: <https://spbmiac.ru/ehlektronnoe-zdravookhranenie/rejtingi-e-zdravookhraneniya/rejtingi-mo-emk-peterburzhca/> (дата обращения 20.12.2020)
52. Информационный Портал «Будущее России. Национальные проекты» [Электронный ресурс]. - URL: <https://futureussia.gov.ru> (дата обращения 20.12.2020)
53. Указ Президента РФ «О национальных целях развития Российской Федерации на период до 2030 года», 21 июля 2020.
54. Стародубов В.И., Сидоров К. В., Зарубина Т. В., Швырёв С. Л., Королева Ю. И., Раузина С. Е. Методика оценки уровня информатизации медицинской организации // Менеджер здравоохранения. – 2017. – №8. – С. 39–52.

55. Стародубов В.И., Сидоров К.В., Зарубина Т.В., Алепко А.А. Формирование интегральных показателей оценки уровня информатизации медицинской организации // Врач и информационные технологии. 2018. №1. С. 5-23.
56. Проект Методики оценки оказания медицинскими организациями государственной и муниципальной систем здравоохранения субъектов российской федерации услуги по записи на прием к врачу на едином портале государственных и муниципальных услуг (функций), письмо Министерства цифрового развития, связи и массовых коммуникаций высшим должностным лицам субъектов Российской Федерации от 08.06.2020 №ОК-П9-070-15108.
57. Морозов С.П., Владзимирский А.В., Сафронов Д.С. Бенчмаркинг для оценки качества цифровизации отделений лучевой диагностики: разработка методологии // Врач и информационные технологии. 2019. №1. С.40-45.
58. Digital Health: A Framework for Healthcare Transformation, Anne Snowdon, RN, PhD, FAAN, Director of Clinical Research, Analytics, HIMSS [Электронный ресурс]. – URL: https://www.himssanalytics.org/dhi?utm_source=wp_dls&utm_medium=email_100120&utm_campaign=dhi_rapid (дата обращения: 07.09.2020)
59. Методика расчета уровня использования медицинских информационных систем в медицинских организациях. Утверждена Председателем Комитета по здравоохранению Санкт-Петербурга В.М. Колабутиным 30.03.2015. – Скан-копия документа (10 с.) [Электронный ресурс]. – URL: <https://spbmiac.ru/wp-content/uploads/2017/12/Metodika-rascheta-SI-MIS.pdf> (дата обращения: 07.09.2020)
60. Орлов Г.М. Метод измерения цифровой зрелости региональной системы записи к врачу на основе эталонной сервисной модели // INJOIT International Journal of Open Information Technologies. 2020. Т. 8, №11. С. 110-121. [Электронный ресурс]. – URL: <http://injoit.org/index.php/j1/article/download/1034/986>

Орлов Геннадий Михайлович
Игнатьева Ольга Анатольевна
Васин Алексей Геннадьевич
Низомутдинов Борис Абдуллохонович

Современные методы обработки и анализа данных

Учебное пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49, литер А