

**Федеральное агентство связи  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)**

Кафедра ПМиК  
Допустить к защите  
зав. кафедрой: проф., д.т.н.

\_\_\_\_\_ Фионов А.Н.

# **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

**Разработка программы для стандартизации  
статистических показателей из медицинских публикаций**

**Пояснительная записка**

Студент \_\_\_\_\_ Дряницын Алексей Андреевич /...../

Факультет \_\_\_\_\_ ИВТ \_\_\_\_\_ Группа \_\_\_\_\_ ИП-212

Руководитель \_\_\_\_\_ Лукинов Виталий Леонидович / .....

Новосибирск 2016г.

## Содержание

Введение.....	3
1 Постановка задачи.....	6
2 Статистика в медицине.....	7
2.1 О статистике .....	7
2.2 Нормальное распределение.....	7
2.3 Некоторые статистические показатели.....	8
2.4 Использование статистики в медицине .....	10
3 Выбор программных средств.....	11
3.1 Языки программирования .....	11
3.2 Программные средства.....	13
4 Реализация.....	25
4.1 Восстановление данных .....	25
4.2 Визуализация частоты использования медицинских показателей .....	25
4.3 Настройка Excel.....	27
Заключение .....	44
Список использованных источников .....	45

## Введение

Дипломная работа посвящена разработке программы для стандартизации статистических показателей из медицинских публикаций. Сбор и анализ данных играет одну из ключевых ролей в современной медицине. Использование собранных данных и их последующий анализ, значительно улучшает процесс лечения, профилактики и диагностики различных заболеваний.

Важным термином в медицине является «доказательная медицина» (ДМ). Он означает такой подход к медицинской практике, при котором каждое решение, относящееся к выбору метода лечения, должно иметь научное обоснование. Для научного обоснования, используется медицинская статистика, которая предоставляет показатели, на основании которых можно производить сравнение различных методологий лечения.

В статистике существует множество различных показателей: стандартное отклонение, размах, стандартная ошибка и многие другие. Каждый из таких показателей характеризует выборку данных с определенной стороны. Например, стандартное отклонение показывает, как распределены значения относительно среднего.

Для сбора и анализа таких статистических показателей используется методология, называемая метаанализ. Он позволяет объединить статистические показатели из разных публикаций и провести их сравнение. Существует два основных подхода к проведению метаанализа. В данной дипломной работе будет рассматриваться обобщение опубликованных результатов исследований. Метаанализ обладает следующими преимуществами:

- 1) Показывает, если выборка оказалась более разнообразна, чем предполагалось изначально
- 2) В результате применения методологии обобщаются несколько исследований
- 3) Позволяет контролировать разнообразие между исследованиями
- 4) Может обнаружить систематические ошибки

Любое исследование, проводимое с помощью метаанализа, делится на формулировку задачи, изучение литературных данных, отбор исследований, исключение показателей, которые не будут исследоваться, выбор модели и сам анализ.

Сбор данных из медицинских публикаций является достаточно трудоемким процессом, который основан на ряде критериев, сформированных на основании задач проведения метаанализа. Необходимо изучить статью, найти нужные статистические показатели и внести их в исследование для дальнейшей обработки.

После завершения сбора данных, выполняется сам анализ данных. На этом этапе возможно возникновение определенных трудностей. Например, в одной из статей, какой – показатель отсутствует, при этом остальные показатели,

выбранные для анализа, есть. Существует несколько вариантов решения данной проблемы:

- 1) исключить показатель из области исследования
- 2) исключить данную статью из выборки
- 3) попробовать восстановить данные

Для облегчения выполнения анализа существует множество инструментов таких как: пакет `metafor` для языка R, приложение STATISTICKA и другие. Каждый такой инструмент содержит наборы необходимых функций, для проведения метаанализа.

Решение затронутых проблем является целью данной дипломной работы.

## **Краткое описание разделов**

### **Раздел 1. Постановка задачи**

Выступает для определения целей и постановки задачи данной работы.

### **Раздел 2. Статистика в медицине**

Описывает статистику как науку и приводит некоторые статистические показатели.

### **Раздел 3. Выбор программных средств**

Многообразие различных инструментов, с помощью которых можно достичь целей работы, поражает воображение. Тем не менее, необходимо выбрать те, что будут наиболее эффективны.

### **Глава 1. Языки программирования**

Глава предназначена для общего ознакомления с языками программирования, которые будут использоваться при выполнении квалификационной работы. Рассматриваются такие языки программирования как C#, R и немного затрагивается VBA.

### **Глава 2. Программные средства**

В данной главе рассматриваются различные программы и компоненты, которые будут использоваться при выполнении работы. В ней можно найти описание таких программ как Visual Studio, RStudio, HelpNDoc, компонент InstallShield и clickOnce, а также некоторые пакеты языка R и VSTO.

### **Раздел 4. Реализация**

В этом разделе будут рассмотрены реализованные программы.

### **Глава 1. Восстановление данных**

Посвящена нелегкому процессу восстановления данных, собранных из медицинских публикаций. Приведены результаты выполнения восстановления по примеру данных

## Глава 2. Визуализация частоты использования медицинских показателей

В этой главе рассматривается выполнение второй задачи квалификационной работы. Рассматривается структура проекта markdown for R и графики, представленные в результате выполнения расчетов.

## Глава 3. Настройка Excel

В главе рассматривается полное описание надстройки, рассматриваются ее возможности и приводится пример ее работы.

## Заключение

Подводятся итоги выполнения работы, рассматривается эффективность созданных приложений, а также потенциал развития надстройки для MS Office Excel.

# 1 Постановка задачи

В данной работе преследуются несколько целей:

- Во-первых, это восстановление пропущенных данных медицинских публикаций. Во время написания статей, разные авторы практически всегда используют определенный набор показателей, который чаще всего отличается от тех, которые используют другие авторы данной области исследования. В следствие чего, затрудняется сравнение статистических показателей сходных по теме статей.
- Во-вторых, требуется провести анализ использования авторами различных статистических показателей и представить их в удобном для восприятия виде.
- В-третьих, упрощение работы с большими таблицами данных. Вся обрабатываемая статистическая информация из статей, заполняется в таблицу программы MS Office Excel. При большом объеме видов показателей, а также числа записей, затрудняется работа с последними.

Теперь, на основании этих целей, сформулируем задачи работы:

- 1) Написать программу, которая будет восстанавливать (если есть возможность) пропущенные медицинские показатели;
- 2) Написать программу, которая по заранее сформированной таблице использования медицинских показателей авторами статей, представит различные графики описательной статистики;
- 3) Создать приложение, которое будет реализовывать эффективное заполнение таблицы

## 2 Статистика в медицине

### 2.1 О статистике

Существует несколько определений понятия статистика и, прежде всего, выделяют статистику как отрасль знаний. Эта «отрасль» содержит общие вопросы по сбору, измерению и анализу массовых статистических данных.

Вообще, слово статистика произошло от латинского слова «status» - статус, состояние, положение дел. А сам термин «статистика» ввел немецкий ученый Готфрид Ахенвалль в 1746, положив тем самым начало развития новой науки.

Другое определение статистики, а точнее – математической статистики, говорит нам о том, что это наука, разрабатывающая математические методы систематизации и использования статистических данных для научных и практических выводов.

### 2.2 Нормальное распределение

Нормальное распределение (или распределение Гаусса) – это распределение вероятностей, совпадающее с функцией Гаусса:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Рисунок 2.1 - Функция Гаусса

где  $\mu$  — математическое ожидание (среднее значение), а  $\sigma$  — среднеквадратическое отклонение.

Нормальное распределение называется стандартным, если  $\mu = 0$ , а  $\sigma = 1$

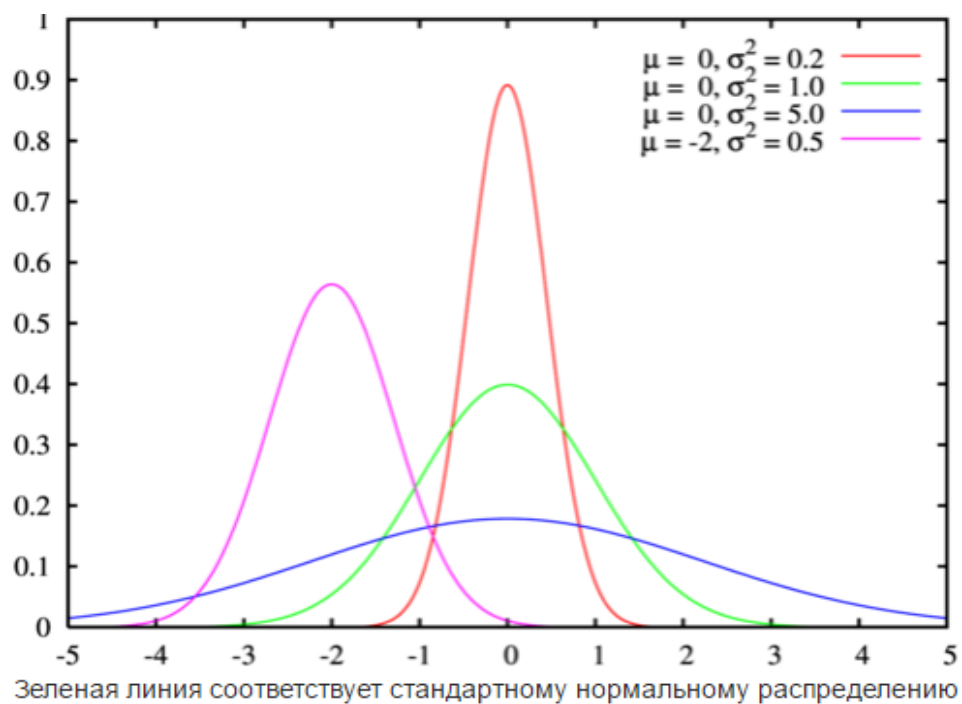


Рисунок 2.2 – График функции плотности нормального распределения

## 2.3 Некоторые статистические показатели

Range (размах) – диапазон значений. Показывает минимальное и максимальное значение выборки.

Mean – среднее арифметическое значение выборки.

Median – значение выборки, которое больше половины выборки и меньше остальной ее части. В более общем случае – элемент, который стоит на средней позиции в выборке, после ее сортировки.

Если распределение выборки является нормальным, то Mean и Median примерно равны.

SD – стандартное отклонение. Показывает рассеивание значения случайной величины, относительно ее математического ожидания. Стандартное отклонение можно найти по формуле:

$$SD = \sqrt{\frac{\sum(x_i - Mean)^2}{n-1}},$$

где  $x_i$  –  $i$ -ый элемент выборки,  
 $n$  – размер выборки

SE – стандартная ошибка. Характеризует стандартное отклонение выборочного среднего. Найти можно по формуле:

$$SE = \frac{SD}{\sqrt{n}}$$



Quantile – квантиль. Значение выборки, которое не превышает случайная величина с заданной вероятностью. По квантилю вычисляется связанный показатель – квартиль (Quartile).

Выделяют три основных квартили:

- 1) Первый (или нижний) квартиль. Он равен четверти от квантиля
- 2) Второй квартиль или медиана. Половина от значения квантиля
- 3) Третий (или верхний) квартиль. Равен 0.75 от квантиля

IQR – характеризует разброс распределения величины. Вычисляется как разность между первой и третьей квартилями.

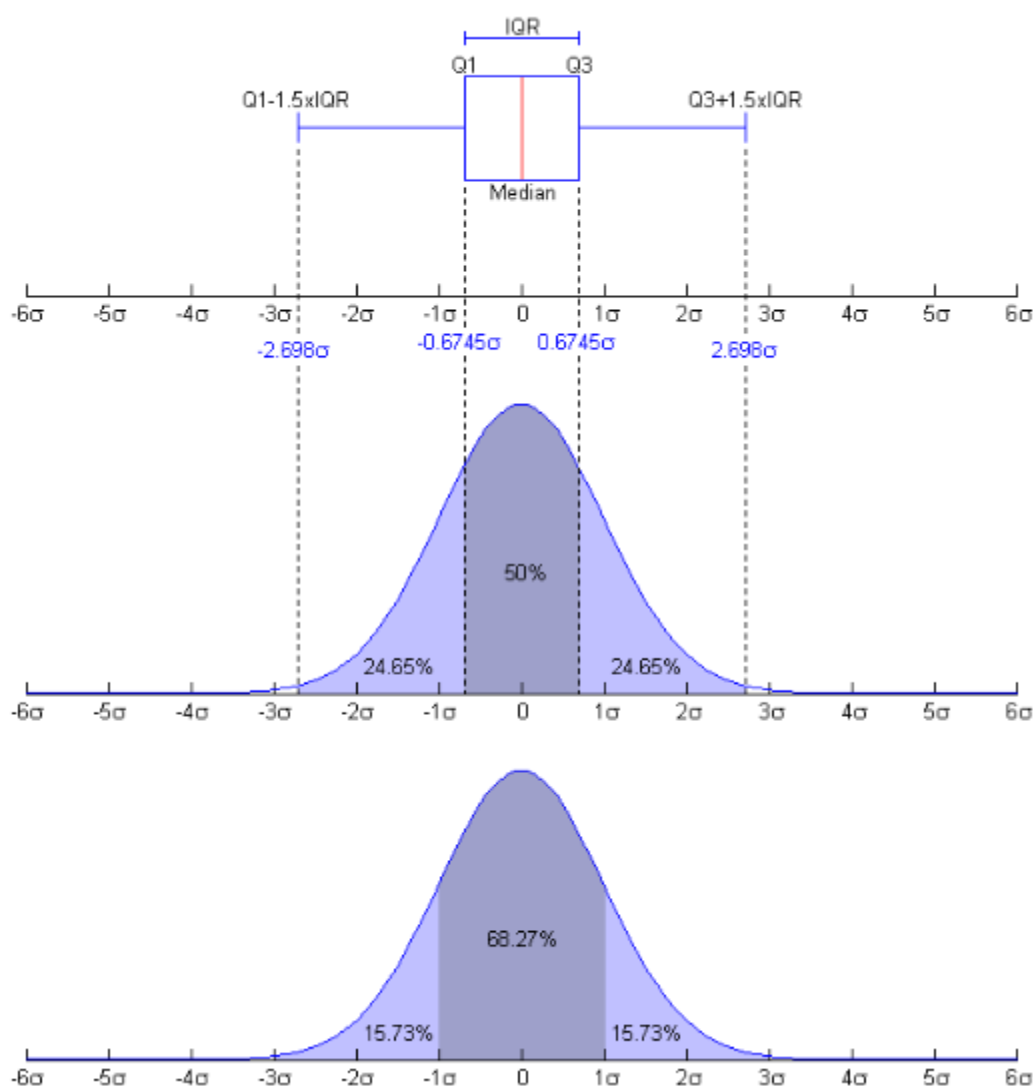


Рисунок 2.3 – Квантили нормального распределения

Для нормального распределения Range, первую и вторую квартили можно вычислить по следующим формулам:

$$Range: Mean - 3SD; Mean + 3SD$$

$$1Qt = Mean - 0.68SD$$

$$3Qt = Mean + 0.68SD$$

Для ненормальных распределений –

$$Range: Median - 3IQR; Mean + 3IQR$$

$$1Qt = Median - 0.5SD$$

$$3Qt = Median + 0.5SD$$

Формулы (номера) дают приближенное значение. Если значение выходит за диапазон допустимых – необходимо подобрать другой коэффициент.

## 2.4 Использование статистики в медицине

Существует отдельная отрасль статистики, которую называют медицинской (санитарной). Она направлена на изучение количественной стороны явлений и процессов, связанных с медициной.

Выделяют три раздела медицинской статистики:

- 1) Статистика здоровья населения
- 2) Статистика здравоохранения
- 3) Клиническая статистика

## 3 Выбор программных средств

Для выполнения задач дипломной работы, необходимо использовать наиболее эффективные инструменты. В ходе поиска и анализа доступных инструментов, мой выбор пал на компоненты, описание которых представлено ниже.

### 3.1 Языки программирования

Приложение, которое будет работать с таблицами Excel, можно сделать разных уровней: отдельное приложение, которое будет работать с csv форматом (универсальный формат представления табличных данных), либо надстройка для приложения MS Office Excel. Более практичным и грамотным решением в данном случае стала надстройка, т.к. это позволяет продолжать использовать Excel не создавая обработчик csv и уменьшает время разработки приложения.

Для написания надстройки Excel можно использовать два языка: C# и Visual Basic for Application (VBA). Конечно, при должном упорстве и приложении недюжинных усилий создать такое приложение можно с помощью практически любого языка, но это означает, что поддерживаемость такого решения будет слишком низкой, код специфичен, а сроки разработки будут огромными.

VBA – встроен в Excel и является общедоступным языком с исполняющей средой .NET. Программная оболочка позволяет разрабатывать приложения уровня документа, либо продукта Office, с использованием практически всех возможностей мощных языков программирования платформы .NET. Встроенность в Excel дает огромное количество готовых методов для обработки данных таблицы.

C# - еще один язык программирования, разработанный корпорацией Microsoft. Более универсальный, простой и понятный язык программирования с огромными возможностями.

Если бы задача была менее масштабной, не требовала бы дальнейшего расширения и развития – выбор пал бы на VBA. Но проект имеет достаточно интересный потенциал и может быть серьезно расширен. Поэтому – выбор пал именно на C#.

Две другие задачи данной работы связаны со статистикой и нет ничего лучше для выполнения работы кроме языка программирования R. Немаловажную роль играет то, что данный язык программирования и отличная среда разработки RStudio являются Open source проектами, при этом являясь мощным скриптовым языком с простым синтаксисом.

#### 3.1.1 Описание языка C#

C# (C Sharp, си-шарп) – объектно–ориентированный язык программирования, разработанный в 1998-2001 годах инженерами под

руководством Андерса Хейлсберга в компании Microsoft. Предназначался для разработки приложений на платформе .NET Framework.

C# разрабатывался на основе опыта использования других, более ранних языках программирования, которые оказались несколько проблематичными для разработки программных систем. Это позволило создать мощный язык программирования с C-подобным синтаксисом для разработки программных систем.

Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, свойства, обобщенные типы и методы, итераторы и многое другое.

Подобно Java, программы, написанные на C#, выполняются на своеобразной «виртуальной машине», предварительно переводясь в представление MSIL (Microsoft Intermediate Language) - CLR, что, с одной стороны, ограничивает возможности C# возможностями CLR, с другой – делает выполнение более безопасным и независимым в рамках платформы .NET Framework.

Последний выпущенный стандарт языка – C# 6.0, значительно упростил некоторые языковые конструкции, добавил новые возможности.

### 3.1.2 Описание языка R

**R – язык программирования, предназначенный для статистической обработки данных и работы с графикой. Так же является свободной программной средой вычислений с открытым исходным кодом в рамках проекта GNU.**

Был разработан сотрудниками статистического факультета Оклендского университета Россом Айхэкой и Робертом Джентлменом **на основе языка S**. С 2011 года поддерживается и развивается организацией R Foundation.

**R поддерживает множество статистических и численных методов и обладает хорошей расширяемостью с помощью пакетов. Каждый такой пакет представляет собой некую библиотеку, содержащую набор специфических функций. Так же, одной из особенностей языка является поддержка графических возможностей, которая позволяет визуализировать данные в виде различных графиков и диаграмм.**

Все пакеты и исполняющая среда распространяются через CRAN (Comprehensive R Archive Network). Существует множество зеркал для удобного распространения, но головной узел расположен в Вене (Австрия).

Пакеты языка R содержат отличную справку по любым функциям/пакетам языка. Для того, чтобы получить справку по интересующей теме, можно использовать `help(«имя команды»)`.

На текущий момент последней версией исполняющей среды R является версия 3.3

## 3.2 Программные средства

Для того, чтобы использовать приведенные языки программирования, потребуются соответствующие среды разработок.

У языка R существует несколько разработанных для него графических интерфейсов: JGR, RKWard, SciViews-R, Statistic Lab, R Commander, Rattle, RStudio.

Каждая из этих сред по-своему уникальна, отлично подходит для решения определенного круга задач, используя для этого язык R. Но наибольшее распространение получила среда RStudio – графический интерфейс с открытым исходным кодом.

Для языка C# так же существует несколько сред, поддерживающих данный язык программирования. Самой популярной является семейство продуктов, созданное корпорацией Microsoft. Речь, конечно же, о Visual Studio.

### 3.2.1 RStudio

RStudio – среда разработки для языка программирования R, предназначенная для статистической обработки данных и работы с графикой. Среда была написана на языке C++ и использует фреймворк Qt для графического интерфейса. Является open-source проектом (проект с открытым исходным кодом) и все модули этого приложения есть на github.

RStudio представлена двумя версиями: Desktop, для использования на локальной машине и Server, доступ к которой можно получить используя браузер.

Основные достоинства RStudio:

- Автоматическая установка пакетов для R с построением зависимостей
- Удобный графический интерфейс с редактором кода
- Интегрированная справка по языку R

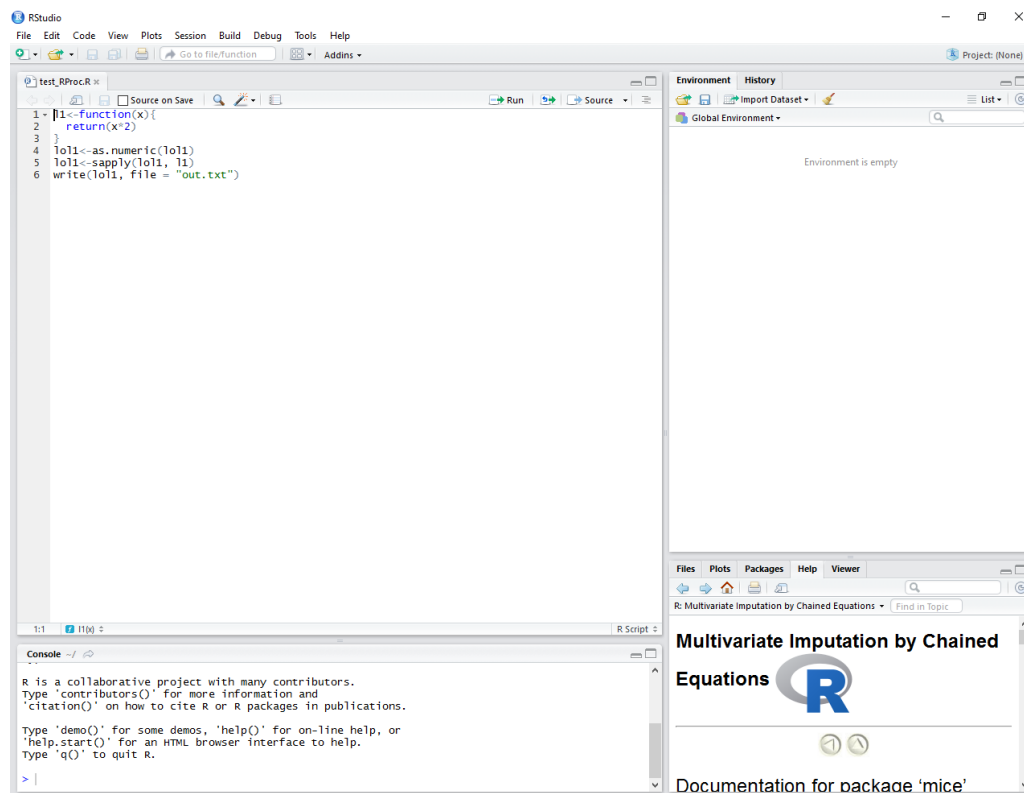


Рисунок 3.0.1 – Внешний вид главного окна RStudio

Помимо самой среды, потребуется несколько пакетов и дополнительных модулей для выполнения задач бакалаврской работы.

### 3.2.2 Пакеты для языка R

Пакет *metafor* – пакет, предназначенный для мета-анализа различных данных. В нем содержится набор функций для вычисления характеристик и преобразования данных. Так же в пакете содержится набор графических модулей для построения различных графиков.

Проект *R markdown* – проект, предназначенный для легкого создания, изменения динамической документации, презентаций и отчетов в формате PDF, HTML/HTML5, MS Word и других. Проект основан на *markdown*, облегченном варианте языка текстовой разметки с простым и понятным синтаксисом. Данный язык разметки был создан с целью написания максимально понятной и читабельно документации, с возможностью преобразовывать в языки для продвинутых публикаций.

Как уже говорилось, *markdown* имеет простой синтаксис. Например, если нужно выделить заголовок первого уровня, то можно сделать это следующим образом:

```
Заголовок 1, вариант 1
=====
#Заголовок 1, вариант 2
```

Рисунок 3.0.2 – Заголовки первого уровня

Если второго уровня, то:

```
Заголовок 1, вариант 1
-----
##Заголовок 1, вариант 2
```

Рисунок 3.3 – Заголовки второго уровня

Поддерживается всего 6 уровней заголовка.

Так же, поддерживаются списки:

1. Элемент 1 нумерованного списка
  2. Элемент 2 нумерованного списка
- \* Элемент маркированного списка
  - Элемент нумерованного списка
  - + буллеты элементов могут быть разными

Рисунок 3.4 – Заголовки первого уровня

Можно выделить текст, используя \*. При этом при \*какой-то текст\* будет курсивом, а при \*\*какой – то текст2\*\* будет полужирным. Использование ~~какой – то текст~~ сделает текст зачеркнутым. Также, можно вставлять ссылки, выделять текст HTML-тегами, вставлять код на любом языке программирования и многое-многое другое.

Возможность генерировать динамическую документацию достигается за счет возможности импортировать в разметку исполняемые участки кода на языке R.

Для создания нового проекта markdown, нужно перейти во вкладку File>New File>R Markdown...

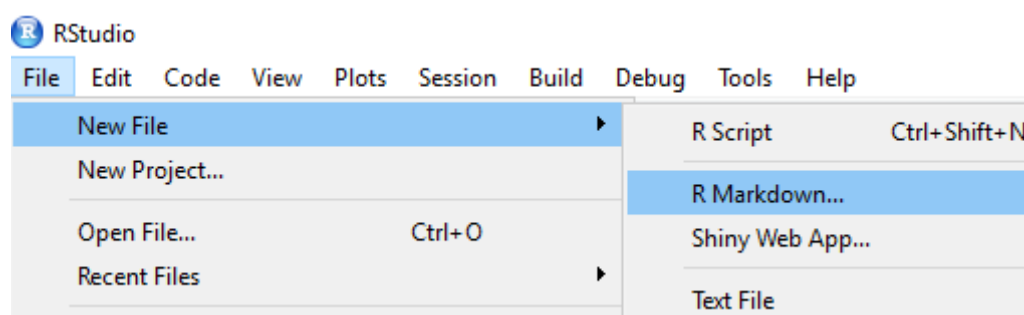


Рисунок 3.4 – Создание нового проекта R Markdown

При создании нового проекта markdown, автоматически создается тестовая страница с примерами выполнения кода, вставки графиков.

```
---
title: "testMD"
author: "legioner942"
date: "21 июня 2016 г."
output: html_document
---

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

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
```

Рисунок 3.5 – Вид нового проекта R Markdown



## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   : 2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

## Including Plots

You can also embed plots, for example:

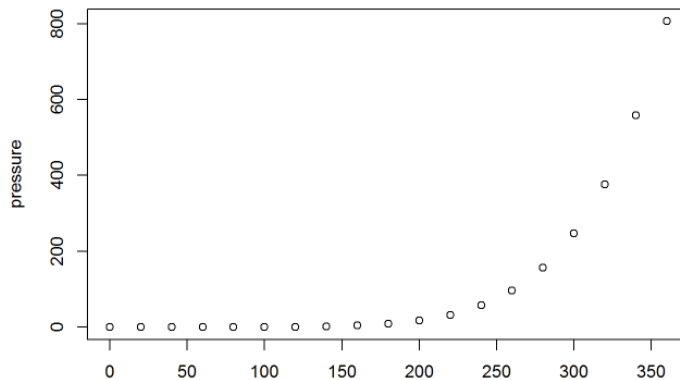


Рисунок 3.6 – Вид страницы нового проекта R Markdown

### 3.2.3 Microsoft Visual Studio

Microsoft Visual Studio – мощная среда для разработки различных по масштабам проектов, созданная корпорацией Microsoft. Среда поддерживает множество языков программирования, позволяет устанавливать дополнительные пакеты и многое другое.

Существуют целая линейка программных продуктов Visual Studio (точнее, выпусков), которые отличаются количеством доступных инструментов для разработки. В дипломной работе будет использоваться Visual Studio 2015 Professional, так как необходимо использовать InstallShield limited edition.

Основные достоинства VS:

- Удобный графический интерфейс
- Технология IntelliSense – технология автодополнения, разработанная корпорацией Microsoft, которая дописывает названия функций при вводе начальных букв. Так же используется для доступа к документации и для устранения неоднозначности в именах переменных, функций и методов, используя рефлексиию.
- Возможность проводить отладку программы

- Инструменты мониторинга использования системных ресурсов (начиная с версии 2015)
- Поддержка множества языков
- И многое другое...

Поддержка огромного количества инструментов открывает широкие возможности для создания огромного числа разных по уровню проектов. Одним из заслуживающих внимания инструментов, будет интегрированная система контроля версий.

Есть две такие системы, поддерживаемые Visual Studio:

Team Foundation – продукт, созданный для нужд Microsoft, обладает широкими возможностями по управлению, изменению и контролю над крупными проектами. Сервис предоставляется как для крупных предприятий, так и для частных лиц, которым требуется подобная система управления собственным проектом. Является условно бесплатным, при оплате подписки открываются дополнительные возможности по управлению проектами, возможность создавать неограниченное количество скрытых репозиторий.

GitHub – крупнейший веб-сервис, созданный на основе git, для хостинга IT-проектов и их совместной разработки. Создан компанией GitHub Inc. Как в случае и с Team Foundation – является условно бесплатным проектом, при оплате подписки предоставляется создавать скрытые репозитории проектов и множество других возможностей. Множество крупных IT-компаний, как например “ДГ-софт”, используют его как систему контроля версий при разработке своих продуктов. Простота, удобство и развитость данного веб-сервиса уже давно завоевала популярность среди программистов и других профессий IT-сферы.

Team Foundation встроена в Visual Studio, но GitHub является дополнительным пакетом, который необходимо установить. Установщик MS Visual Studio включает в себя пункт об установке необходимых компонентов. После установки, необходимо выполнить вход в свой аккаунт на GitHub, но допустимо использование локальных компонентов git. Можно успешно продолжать свою работу, не выкладывая исходный код проекта на удаленный сервис.

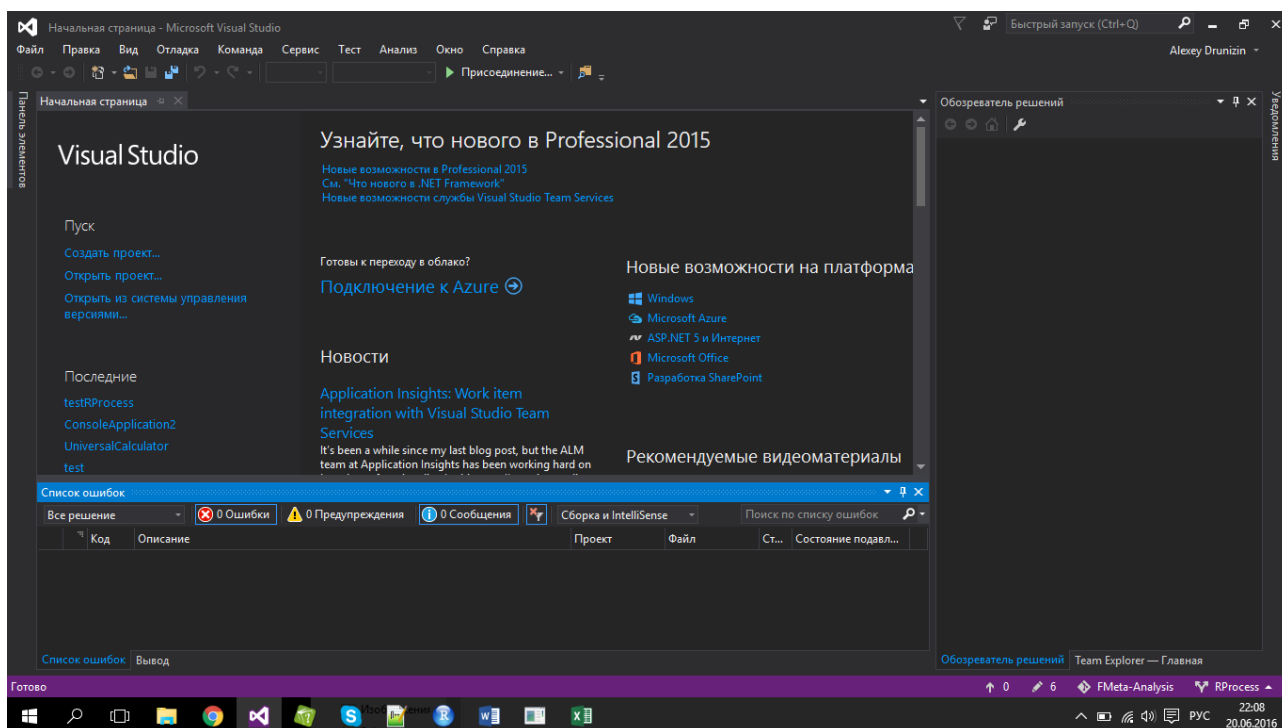


Рисунок 3.7 – Интерфейс Visual Studio

### 3.2.4 Visual Studio Tools for Office (VSTO)

VSTO – набор средств разработки, доступных в виде шаблонов проектов и runtime, что позволяет приложениям Office использовать функциональность .NET Framework Common Language Runtime (CLR) с помощью .NET type system. Как результат – приложения Office могут быть расширены с помощью CLI (Common Language Infrastructures) совместимого языка (C#, Visual Basic), а также использовать функциональные возможности и пользовательский интерфейс конструкций из приложений Office в .NET приложениях.

VSTO add-in (типы проектов и управления) позволяют разрабатывать надстройки для Microsoft Office различного уровня с использованием Visual Studio IDE.

Шаблоны проектов интегрированы в Visual Studio в выпусках, начиная с Professional. Возможно установить на версию Community.

### 3.2.5 InstallShield и ClickOnce

InstallShield – система, используемая для создания инсталлятора под операционные системы семейства Windows и Linux с закрытым исходным кодом.

Первая утилита была создана в 1992 году в Stirling Technologies, в дальнейшем права на утилиту переходили из рук в руки. В настоящий момент права принадлежат компании Flexera Software. С 2010 года интегрирована во все выпуски Visual Studio, кроме версии Express.

В 2015 году была введена поддержка Visual Studio 2015 для выпусков Professional, Premium и Ultimate.

Утилита является «условно» платной. Есть дополнительные возможности, за которые необходимо платить. Но базового пакета вполне достаточно.

Основные достоинства:

- Наличие подробной документации
- Большие возможности базового пакета
- Процесс настройки установщика прост и понятен
- Интеграция с Visual Studio

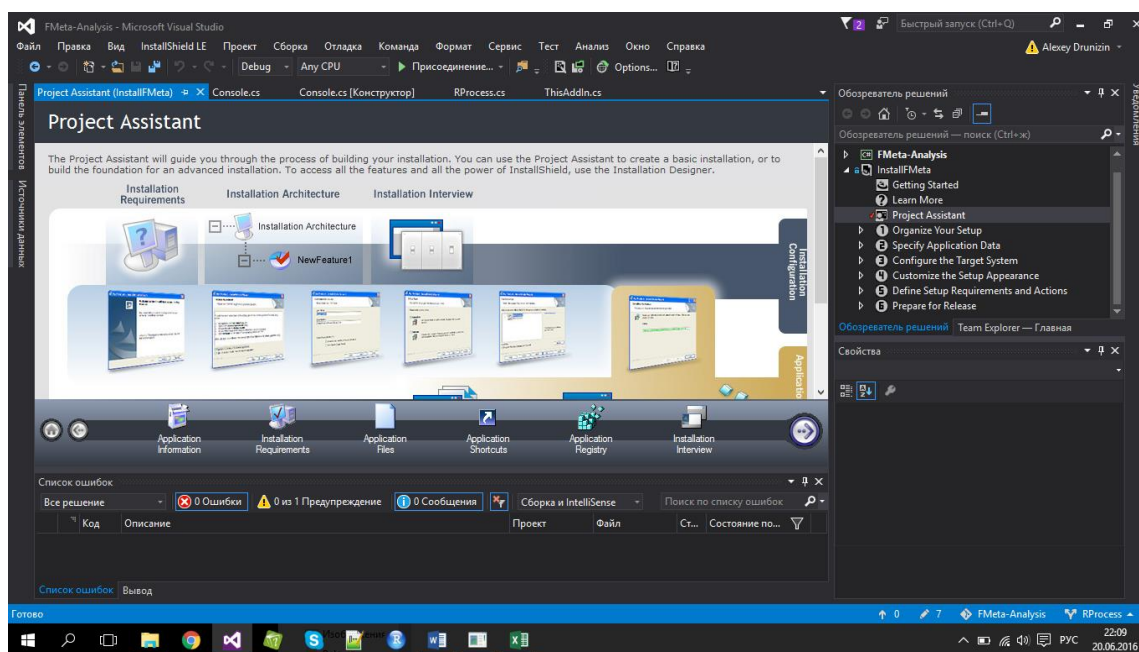


Рисунок 3.8 – Главное меню проекта InstallShield

Кроме того, особенно можно выделить возможность выполнения дополнительных сценариев (JScript, VBScript или .exe) после завершения процесса установки. Это позволяет сразу запустить установленное приложение, не затрачивая время пользователя на поиск установленной программы.

Помимо InstallShield, существует технология clickOnce, разработанная корпорацией Microsoft. Она предназначена для развертывания приложений различного уровня, основанных на фреймворках Windows Forms и WPF

(Windows Presentation Foundation) и доступна для проектов с версией .NET 2.0 и выше.

Ее главным преимуществом, помимо простоты использования, а также развертывания, является интегрированность во все версии Visual Studio. Кроме того, все необходимые компоненты собираются в автоматическом режиме и практически не требует особых усилий со стороны того, кто хочет опубликовать свой проект.

Данная технология позволяет устанавливать приложение либо через ссылку на веб-странице, либо через сетевое окружение. Кроме того, использование данной технологии позволяет избежать некоторых проблем, которые могут возникнуть при установке приложения. Например, права администратора для выполнения установки. Если при установке требуется получить права администратора – установщик автоматически запросит разрешение прав администратора.

Для того, чтобы собрать исполняемые файлы, достаточно в Visual Studio правой кнопкой мыши кликнуть на проект и выбрать пункт «Опубликовать». Затем, следуя простым инструкциям, мы достаточно быстро получаем сборку необходимых для установки файлов, собранных в отдельной папке.

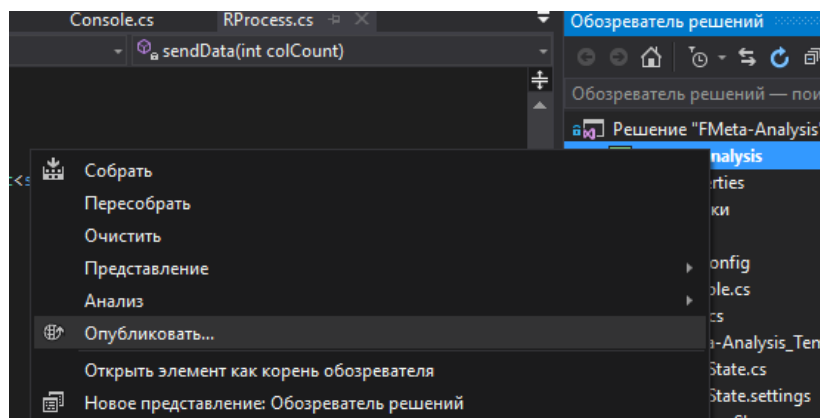


Рисунок 3.9 – Публикация проекта

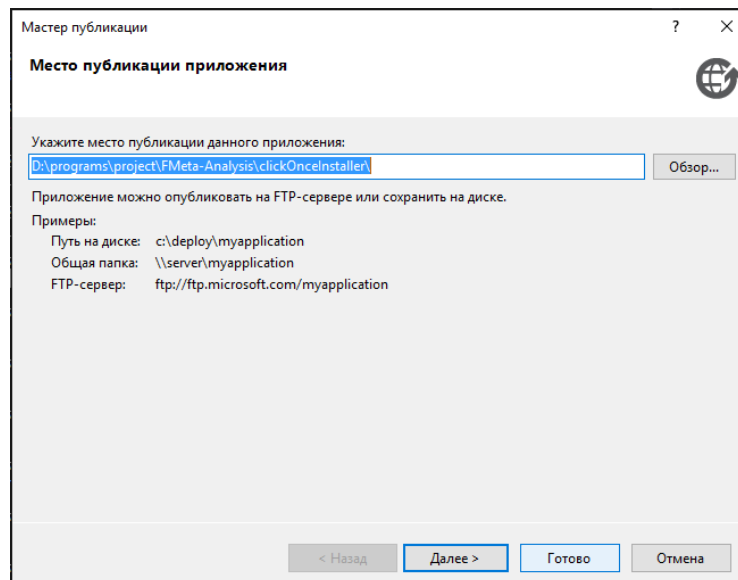


Рисунок 3.10 – Выбор папки для публикации

Нажимаем «Готово» и по указанному пути расположения появляются все необходимые компоненты для установки приложения на другом компьютере.

|                   |                  |                    |          |
|-------------------|------------------|--------------------|----------|
| Application Files | 21.06.2016 13:53 | Папка с файлами    |          |
| FMeta-Analysis    | 21.06.2016 13:53 | VSTO Deployment... | 6 КБ     |
| setup             | 21.06.2016 13:53 | Приложение         | 1 385 КБ |

Рисунок 3.11 – Папка, в которую выполнялась публикация

Выполнив setup.exe мы с легкостью установим надстройку.

Если же требуется настроить компоненты, изменить версию пакета или что – то другое – можно перейти в свойства проекта и выбрать пункт публикация. Открывшееся меню предоставляет больше возможностей по настройке.

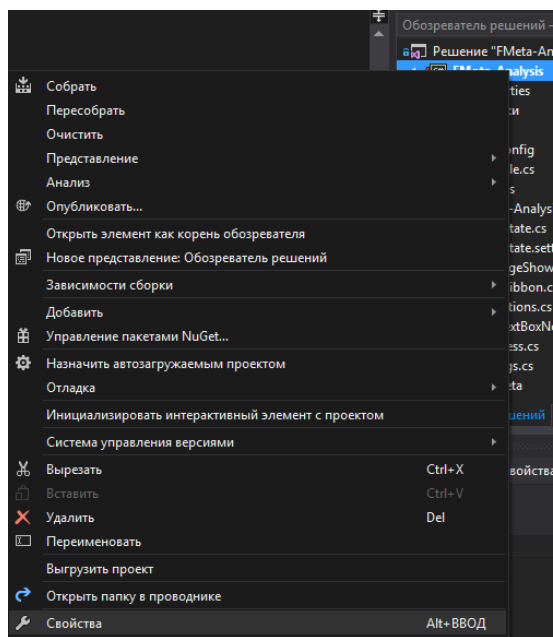


Рисунок 3.12 – Свойства проекта

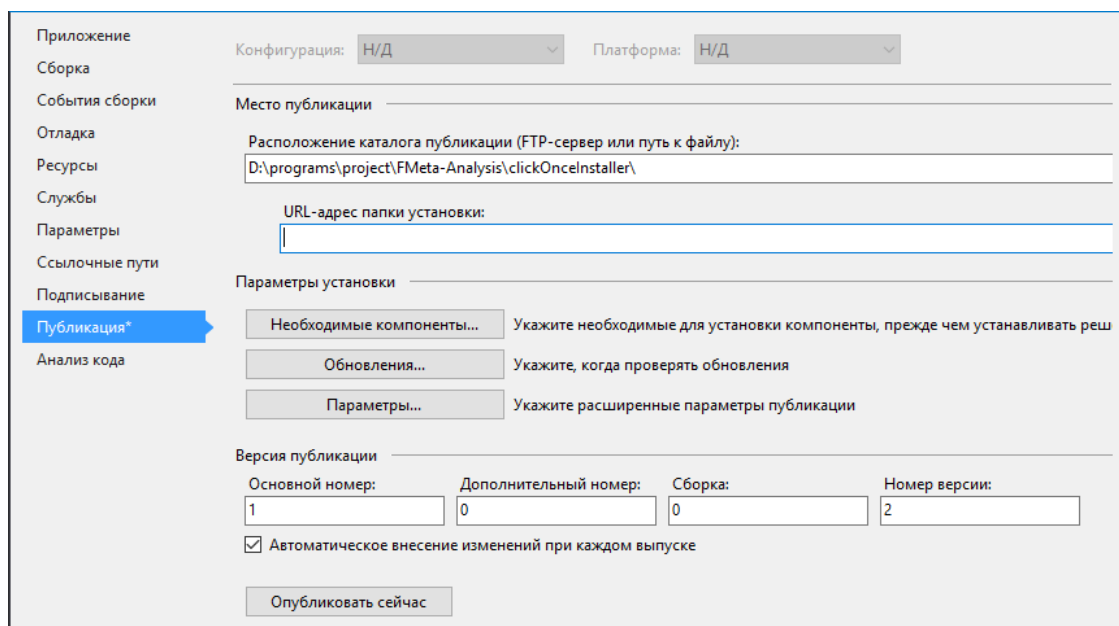


Рисунок 3.13 – Вкладка «Публикация»

Конечно, InstallShield предоставляет больше тонкой настройки, позволяет работать с реестром, создавать отдельные папки для проекта в разных расположениях, выполнять действия после завершения установки. Но в данном случае, для такого проекта проще всего использовать именно ClickOnce, за его простоту и удобство.

### 3.2.6 HelpNDoc

HelpNDoc – простая и мощная программа для создания документации в разнообразных форматах (CHM, HTML, PDF). Является бесплатной для частного использования.

Минусом является назойливая надпись, которая сообщает о том, что документация была сгенерирована в этой программе, но программа достаточно удобна и проста в использовании, к тому же, поддерживает проверку орфографии и многое другое.

#### Основные достоинства:

- Простая и мощная
- Бесплатная для частного пользования
- Проверка орфографии
- Поддержка основных форматов документации
- Генерация кода для различных языков программирования (C/C++, Delphi/Pascal, Visual Basic и другие)
- Поддержка шаблонов и другое

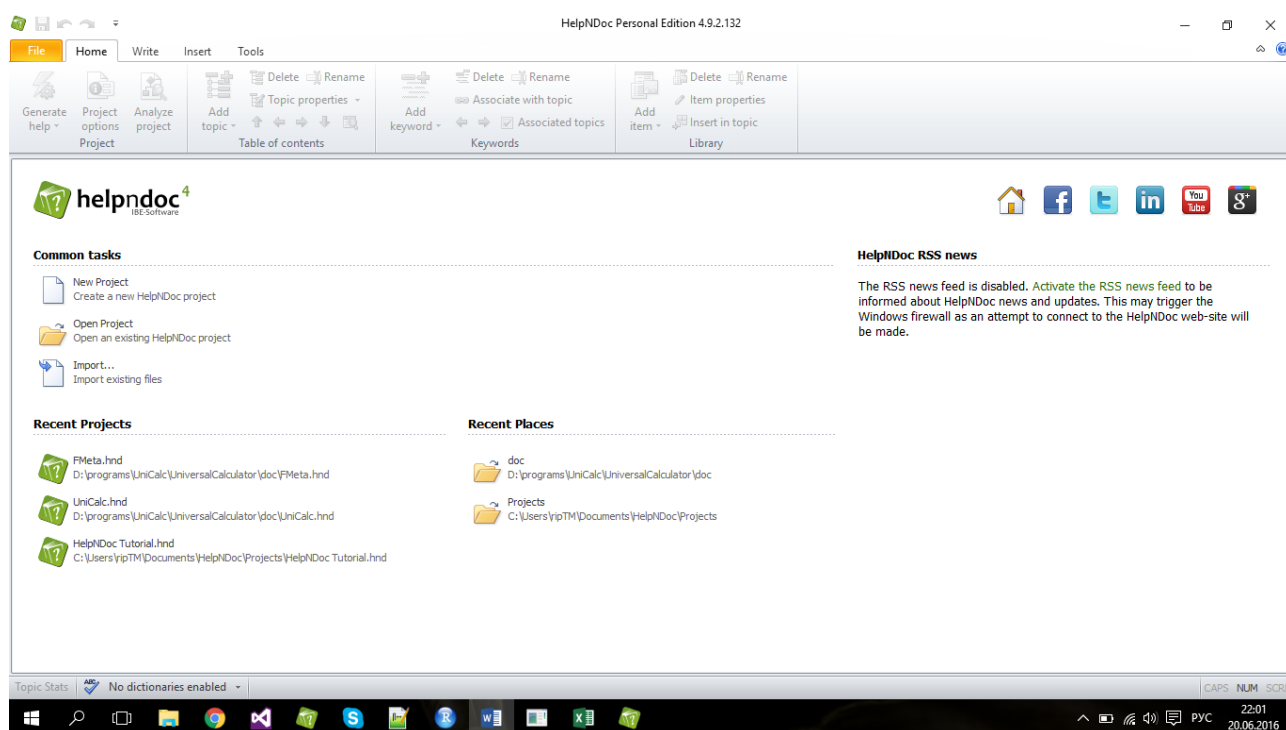


Рисунок 3.14 – Интерфейс HelpNDoc

Так же стоит отметить проблему, связанную с кодировкой. При создании категорий с русскими названиями, после сборки документации они отображаются вопросительными знаками. При этом текст справки самого раздела отображается корректно.

Для решения этой проблемы, нужно перейти в general settings (основные настройки), затем изменить project language (язык проекта) на русский и project charset (кодировка символов проекта) на cyrillic.



Рисунок 3.4 – Настройки для исправления кодировки



## 4 Реализация

### 4.1 Восстановление данных

Используя формулы, описанные в разделе 2.3, выполнение данной задачи достаточно просто. В созданном скрипте опишем функции, которые будут вычислять значения по данным формулам. Также, для подготовки таблицы к восстановлению, необходимо удалить записи, данные в которых не получится восстановить. А дальше – используя описанные функции восстанавливаем нужные показатели.

Для примера восстановим показатель Range.

| author    | author | cite | group.type | treatment                 | group.size | years.mean | years.sd | years.range |
|-----------|--------|------|------------|---------------------------|------------|------------|----------|-------------|
| Lee       | 2008   |      | treatment  | acupuncture               | 44         | 40,9       | 11       |             |
| Lee       | 2008   |      | control    | sham acupuncture          | 45         | 42,8       | 9,4      |             |
| Lee       | 2009   |      | treatment  | electroacupuncture        | 12         | 39,8       | 5,8      |             |
| Lee       | 2009   |      | control    | sham electroacupuncture   | 12         | 36,4       | 5,8      |             |
| Lee       | 2009   |      | treatment  | advice+exercise           | 12         | 38,2       | 6,9      |             |
| Tkachuk   | 2009   |      | treatment  | afala 8t/day              | 25         | 28,9       |          | 20-46       |
| Zimmerman | 2009   |      | control    | placebo                   | 30         | 43         |          | 34-61       |
| Zimmerman | 2009   |      | treatment  | extracorporeal shock wave | 30         | 42         |          | 22-52       |
| He        | 2010   |      | treatment  | biofeedback               | 21         |            |          |             |
| Pontari   | 2010   |      | treatment  | pregabalin                | 218        | 48         | 13       |             |
| Pontari   | 2010   |      | control    | placebo                   | 106        | 45,2       | 12,2     |             |

Рисунок 4.1 – Исходные данные

| author    | author.1 | cite | group.type | treatment                 | group.size | years.mean | years.sd | years.range |
|-----------|----------|------|------------|---------------------------|------------|------------|----------|-------------|
| Lee       | 2008     |      | treatment  | acupuncture               | 44         | 40,9       | 11       | 8-74        |
| Lee       | 2008     |      | control    | sham acupuncture          | 45         | 42,8       | 9,4      | 15-71       |
| Lee       | 2009     |      | treatment  | electroacupuncture        | 12         | 39,8       | 5,8      | 22-57       |
| Lee       | 2009     |      | control    | sham electroacupuncture   | 12         | 36,4       | 5,8      | 19-54       |
| Lee       | 2009     |      | treatment  | advice+exercise           | 12         | 38,2       | 6,9      | 18-59       |
| Tkachuk   | 2009     |      | treatment  | afala 8t/day              | 25         | 28,9       |          | 20-46       |
| Zimmerman | 2009     |      | control    | placebo                   | 30         | 43         |          | 34-61       |
| Zimmerman | 2009     |      | treatment  | extracorporeal shock wave | 30         | 42         |          | 22-52       |
| Pontari   | 2010     |      | treatment  | pregabalin                | 218        | 48         | 13       | 9-87        |
| Pontari   | 2010     |      | control    | placebo                   | 106        | 45,2       | 12,2     | 9-82        |

Рисунок 4.2 – Результат восстановления

### 4.2 Визуализация частоты использования медицинских показателей

Программа использует описанный в пункте 3.2.2 проект R markdown.

Весь алгоритм сводится к загрузке данных из csv файла и последовательной обработки столбцов для каждого показателя. На каждом шаге будет добавляться график к HTML-странице.

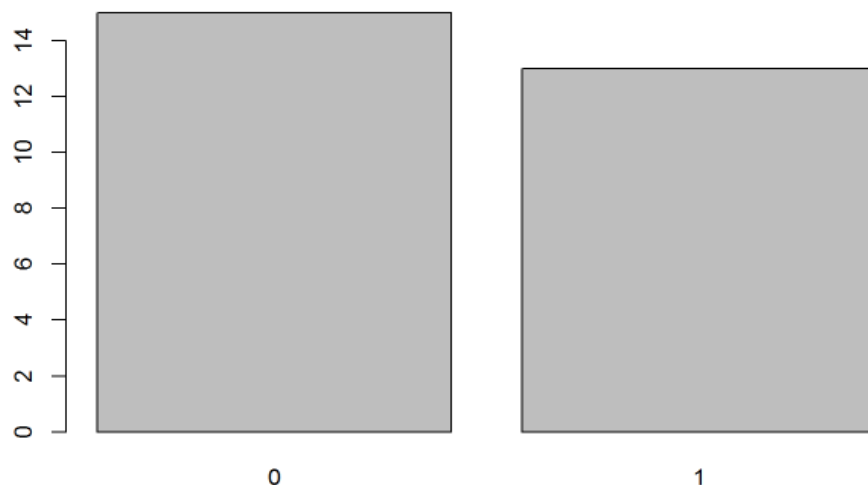
Для рисования графика, будет использоваться функция plot. Для того, чтобы она рисовала столбчатые диаграммы, на каждой итерации будем преобразовывать считанный вектор в двухуровневый фактор.

Так же, необходимо учесть, что необходимо выполнять проход не с самого первого столбца, т.к. там содержатся текстовые данные и нам они ни к чему.

Результат работы скрипта:

*legioner942*

```
## [1] 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 0 0 1
## Levels: 0 1
```



```
## chr "Graph"
## [1] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
## Levels: 0 1
```



Рисунок 4.3 – Страница с графиками статистических показателей

0 – авторы, которые не использовали данный параметр в своих работах

1 – авторы, которые использовали данный показатель в своей работе

Добавим к каждому графику название параметра, и изменим цвета линий графика:

```
## [1] "Param = randomized.controlled.trials"  
## [1] 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 0 0 1  
## Levels: 0 1
```

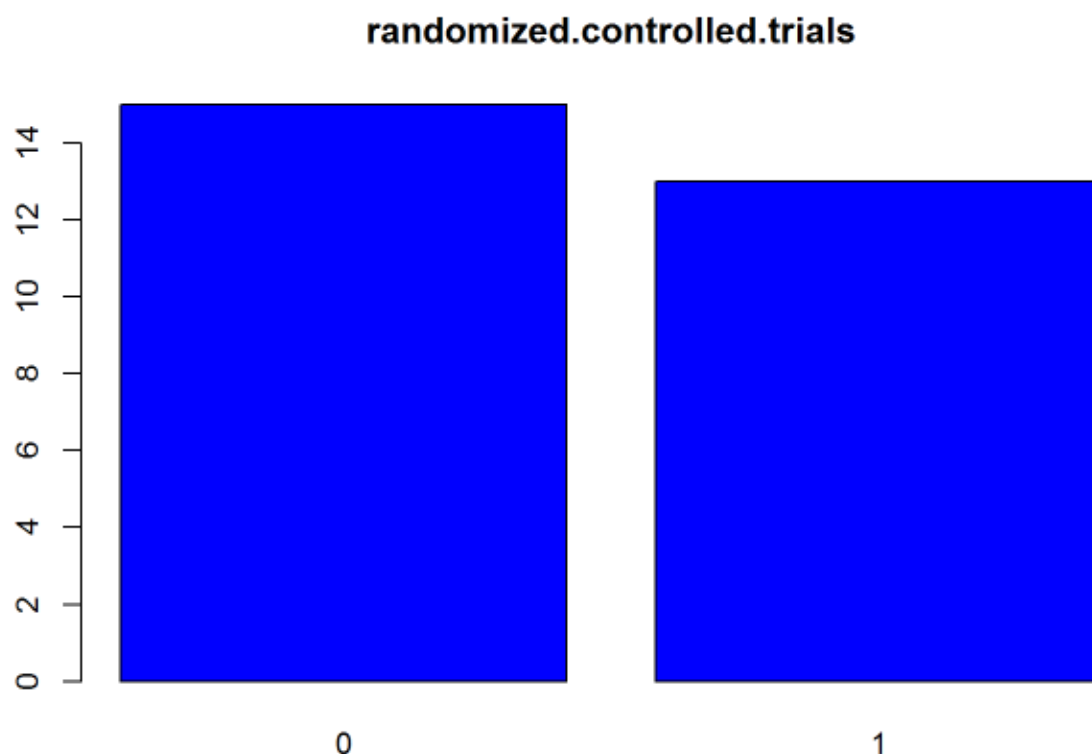


Рисунок 4.3 – Измененный график

## 4.3 Надстройка Excel

### 4.3.1 Общее описание и возможности

Надстройка предназначена для работы с крупными таблицами данных, позволяя немного ускорить и упростить процесс работы. Основной целью было минимизировать использование мыши для переключения между ячейками. Достигается это за счет использования консольного окна, с помощью которого можно взаимодействовать с таблицей.

Программа на данный момент поддерживает вставку, удаление, очистку, изменение значения строки, столбца или отдельной ячейки.

Снижение вероятности ошибки при использовании команд, достигается за счет использования нескольких уровней адресации:

- 1) Глобальный – номера строк и номера столбцов Excel
- 2) Внутренний – имена столбцов в заголовке (первая строка в таблице)

Для длинных названиях столбцов, можно устанавливать уникальные сокращения каждого и использовать их в качестве названий в некоторых командах.

Программа позволяет сохранять данные обо всех столбцах во внешний файл в формате json и наоборот, загружать готовые шаблоны. Так же, программа сохраняет данные об именах столбцов и сокращениях каждые 5 минут (периодичность сохранений можно изменить в настройках) и после выхода из программы Excel в файл dump.

Так же, программа может выполнять обработку данных, загрузив файл скрипта, написанного на языке R. Данные будут автоматически считаны, загружены в исполняющую среду и обработаны.

Пользователь данной надстройки может ознакомиться с полным описанием команд в разделе «Справка», настроить папку для сохранений файлов-шаблонов заголовка, изменить цвет шрифта для обычного текста и команд.

#### 4.3.2 Структура надстройки

Проект состоит из нескольких классов:

- 1) ThisAddIn
- 2) MetaRibbon
- 3) Console
- 4) Core
- 5) Column
- 6) Settings
- 7) FormState
- 8) RProcess
- 9) MessageShower

ThisAddIn – автоматически генерируется студией при создании проекта-надстройки для приложения Microsoft Office. Помимо сгенерированного кода, в нем записаны процедуры, которые отвечают за вставку, удаление, изменение и получение данных из таблицы Excel.

MetaRibbon – класс-компонент, отвечающий за отображение ленты в приложениях Microsoft Office. Класс не представляет особого интереса и содержит единственный метод, отвечающий за обработку нажатия на кнопку «Вызов консоли». Остальной код генерируется автоматически студией и изменений не требует.

Console – форма консоли, в ней обрабатываются вводимые пользователем команды, отображается история, открывается доступ к настройкам, справке, информации о программе, выполнение программы на языке R. Так же, есть метод, который отвечает за подсветку синтаксиса в истории и поле ввода. Введенная пользователем команда отправляется на обработку в Core.

Core – основной класс, отвечающий за обработку и выполнение команд. Так же, он хранит информацию о текущем заголовке, сокращениях и содержит методы для сохранения и загрузки файлов с информацией о заголовке.

Column – вспомогательный класс, у которого объявлены три поля: номер столбца, его название и сокращение.

Settings – форма для настройки консоли. Содержит методы восстановления настроек по умолчанию, изменение цвета шрифта, изменение пути для сохранения данных о заголовке.

FormState – компонент, который хранит текущие настройки приложения. По умолчанию содержит стандартные параметры.

RProcess – собирает данные из таблицы, вводит в исполняющую среду R, затем загружает команды из файла скрипта.

MessageShower – этот статический класс содержит методы, отображающие разные типы диалоговых окон с заданным текстом.

#### 4.3.3 Описание полей и методов Core

Класс так же содержит перечисление (enum) command, которое содержит коды операций.

Поля:

reduction – список строковых объектов, содержащий распознаваемые команды.

listOfCol – список объектов типа Column, содержащий текущие номера столбцов, их имена и сокращения.

paramLine – список строковых объектов, содержащий параметры для текущей команды.

current, subcommand – поля типа command, содержащие текущую операцию и подоперацию соответственно.

Init – поле логического типа (bool), которое показывает была ли проведена инициализация.

Методы:

Compiler() – метод, обрабатывающий входную строку и выделяющий команду, подкоманду и параметры выполнения. При этом, подкоманда или параметры могут отсутствовать. До начала разбора, входная строка обрабатывается вспомогательным методом RemoveSpaces(), который удаляет лишние пробелы из строки. После разбора записываются коды операции и подоперации (если подкоманда отсутствовала – то ей присваивается код -1, что соответствует команде «по») и список операндов. Если для команды нет операции, вызывается исключение, которое отображает окно с сообщением о том, что команда не была распознана. Если один из операндов команды равен

«nil», то такой параметр считается пустым. После окончания обработки вызывается метод Execute.

Execute() – анализирует записанные коды операций и вызывает соответствующие методы-обработчики. Так же метод проверяет число параметров и была ли выполнена инициализация. Если нарушается условие – вызывается исключение.

Методы set(), add(), init(), delete(), clear() выполняют соответствующие их названию операции с таблицей. Подробное описание команд можно найти в пункте 4.3.6

colNumtoAlph() и AlphtoColNum() – группа статичных методов, которые преобразуют номер столбца в ему эквивалентное буквенное представления (для Excel) и наоборот.

getListCol() – возвращает список, каждый элемент которого является объектом класса Column.

save\_column\_name\_toFile() – сохраняет файл, содержащий информацию о текущем рабочем пространстве (используемом заголовке) в формате json.

load\_from\_file() – загружает информацию о рабочем пространстве и выполняет инициализацию.

#### 4.3.4 Описание полей и методов ThisAddIn

Поля:

core, console – поля типа Core и Console.

Методы:

ThisAddIn\_Startup() – автоматически генерируемый метод, дополненный созданием объектов класса Core и Console и установкой папки по умолчанию для сохранений, если путь отсутствует. Метод вызывается автоматически, после старта приложения.

ThisAddIn\_Shutdown() – автоматически генерируемый метод, вызываемый перед закрытием приложения. Выполняет сохранение текущего состояния заголовка в файл dump, сбрасывает некоторые поля FormState.

setCell() – устанавливает переданное значение в заданную ячейку по номеру строки и названию/номеру столбца.

setRow() – метод записывает переданные значения в заданную строку таблицы.

getRow() – возвращает из указанной строки, указанное количество значений.

getColumnWithData() – возвращает все значения из указанного столбца.

getCell() – возвращает значение заданной номером строки и номером столбца ячейки.

clearRow() – выполняет очистку/удаление строки. При удалении все строки таблицы смещаются вверх.

`clearCol()` – выполняет очистку/удаление столбца. При удалении все столбцы таблицы смещаются влево.

`getRowCountWithData()` – возвращает число строк, в которых есть данные.

`getcolCountWithData()` – возвращает число столбцов, в которых есть данные.

#### 4.3.5 Описание полей и методов Console

Поля:

`lastEnter` – строковое поле, которое содержит последнюю введенную команду.

`proc` – поле типа `RProcess`.

Методы:

`getListFile()` – возвращает список файлов, которые находятся в папке для сохранений

`inpCons_KeyDown()` – обработчик нажатия клавиши в поле ввода. Захватывает для обработки две клавиши: стрелка вверх и `Enter`. При нажатии стрелки вверх восстанавливает последнюю введенную команду, при `Enter` – обрабатывает текст, введенный в поле ввода и отправляет их в `core`.

`setStatus()` – метод устанавливает состояние инициализации в индикаторе.

`Console_Load()` – метод, вызываемый при загрузке формы. Загружает сохраненную историю (только для текущего сеанса) и состояние инициализации в `FormState`.

`Console_FormClosing` – метод, вызываемый при закрытии формы. Сохраняет историю и состояние инициализации в `FormState`.

`logTextBox_TextChanged()` – метод подсветки ключевых слов в поле истории. Вызывается при изменении текста в `logTextBox`. Обрабатывает только последний добавленный кусок текста, что значительно улучшает скорость обработки.

`inpCons_TextChanged()` – метод подсветки ключевых слов в поле ввода.

`timerStart()` – устанавливает параметры таймера автосохранения и запускает его.

`Autosave_Tick()` – срабатывает при окончании работы таймера. Выполняет сохранение данных о столбцах в файл `dump` и вызывает метод `timerStart()`.

#### 4.3.6 Описание команд надстройки

Команды, обрабатываемые консолью:

- 1) `clear` – очищает окно истории
- 2) `list` – выводит список файлов-шаблонов, которые доступны в папке сохранений

Команда инициализации:

- 1) `init` – выполняет автоматическую инициализацию. Для выполнения требуется, чтобы строка заголовка была предварительно заполнена
- 2) `init <буквенное_обозначение_столбца>` - инициализирует для работы часть столбцов. Внимание! Команды, которые добавляют значения последовательно при выходе за пределы области будут так же срабатывать.
- 3) `init add <список_названий_столбцов>` - заполняет заголовок и производит инициализацию.

Необходимо чтобы названия столбцов были уникальны.

Команда вставки:

- 1) `add <номер_строки> <список_пар(имя/сокращение; значение)>` - вставляет в соответствующие именам/сокращениям ячейки строки, заданные значения
- 2) `add row <номер_строки> <перечисление_значений>` - последовательно вставляет в указанную строку заданные значения
- 3) `add col <буквенное_обозначение_столбца>/<номер_столбца> <значения>` - последовательно вставляет указанные значения в заданный столбец

Если вместо номера строки использовать модификатор «n» или «new», будет произведена вставка новой строки в конец таблицы.

Команды `add row` и `add col` перед вставкой очищают строку/столбец!

Команда установки:

- 1) `set nick <список_пар(имя_столбца; сокращение)>` - задает сокращение для заданного столбца
- 2) `set hcol <список_пар(имя_столбца; новое_имя_столбца)>` - изменяет имя заголовка столбца

Команда удаления:

- 1) `del row <номер_строки>` - удаляет строку со смещением остальных строк вверх
- 2) `del col <буквенное_обозначение_столбца>/<номер_столбца>` - удаляет заданный столбец со смещением остальных влево. При этом данный столбец так же удаляется из области

Команда очистки:

- 1) `clear row <номер_строки>` - очищает заданную строку



- 2) `clear col <буквенное_обозначение_столбца>/<номер_столбца>` - очищает заданный столбец. Заголовок при этом сохраняется

Команда сохранения/загрузки:

- 1) `load <имя_файла>` - загружает файл с заданным именем и выполняет инициализацию
- 2) `save <имя_файла>` - сохраняет данные о заголовке в виде файла, с заданным именем, в папку для сохранений

#### 4.3.7 Описание выполнения R-скриптов

Конструктор класса `RProcess`, с помощью класса `Process`, создает новый фоновый процесс, который является исполняющей средой R. Чтобы отсылать данные в этот процесс, требуется переопределить стандартный поток ввода для этого процесса. Так же, необходимо переопределить стандартный поток ошибок, отключить отображение окна процесса и задать параметры запуска среды без сохранений состояния. Последнее необходимо для того, чтобы процесс не вылетал при попытке переопределить поток ввода. Для того, чтобы задать параметры запуска, воспользуемся классом `ProcessStartInfo`:

```
info = new ProcessStartInfo()
{
    Arguments = "--vanilla --slave",
    UseShellExecute = false,
    CreateNoWindow = false,
    RedirectStandardError = true,
    RedirectStandardInput = true,
};
```

Рисунок 4.4 – Инициализация структуры `ProcessStartInfo`

Выполнение скрипта происходит в несколько этапов:

На первом этапе собираются данные из таблицы по столбцам и отправляются в виде команд создания списка:

`<имя_столбца><-c(<значение1>,...,<значениеN>)`

В результате, в исполняющей среде появляются все данные, которые есть в таблице на текущий момент.

Затем, загружается файл скрипта и построчно обрабатываясь, так же отправляясь в исполняющую среду и выполняясь.

Недостатками такого метода выполнения являются:

- 1) При больших объемах данных время обработки достаточно большое
- 2) Сложности с переводом типов данных. В данной работе все данные, загруженные в исполняющую среду, будут представлять собой векторы строковых значений
- 3) Отсутствие значения в исполняющей среде при написании скрипта. Повышается вероятность ошибки, т.к. работать приходится практически в слепую.

Решением этих проблем мог бы стать готовый API (application programming interfaces – интерфейс прикладного программирования), но, к сожалению, на данный момент такие интерфейсы пока не поддерживают версии исполняющей среды R выше, чем 3.0. И интерфейсы являются более громоздкими, тогда как задача выполнить небольшие скрипты, пусть и с большим объемом данных, требует минимальных ресурсов.

Проблему под номером 2 можно решить, добавив обработчик, отделяющий числовые значения от текстовых и вызывающий для числовых преобразование вектора к вектору с числовыми значениями. Команда, которая выполняет преобразование – `as.numeric(имя_вектора)`. Тогда, если в среду передается вектор, который содержит числа, он будет автоматически преобразовываться после добавления среду выполнения.

#### 4.3.8 Пример работы

Опишем пример работы с надстройкой.

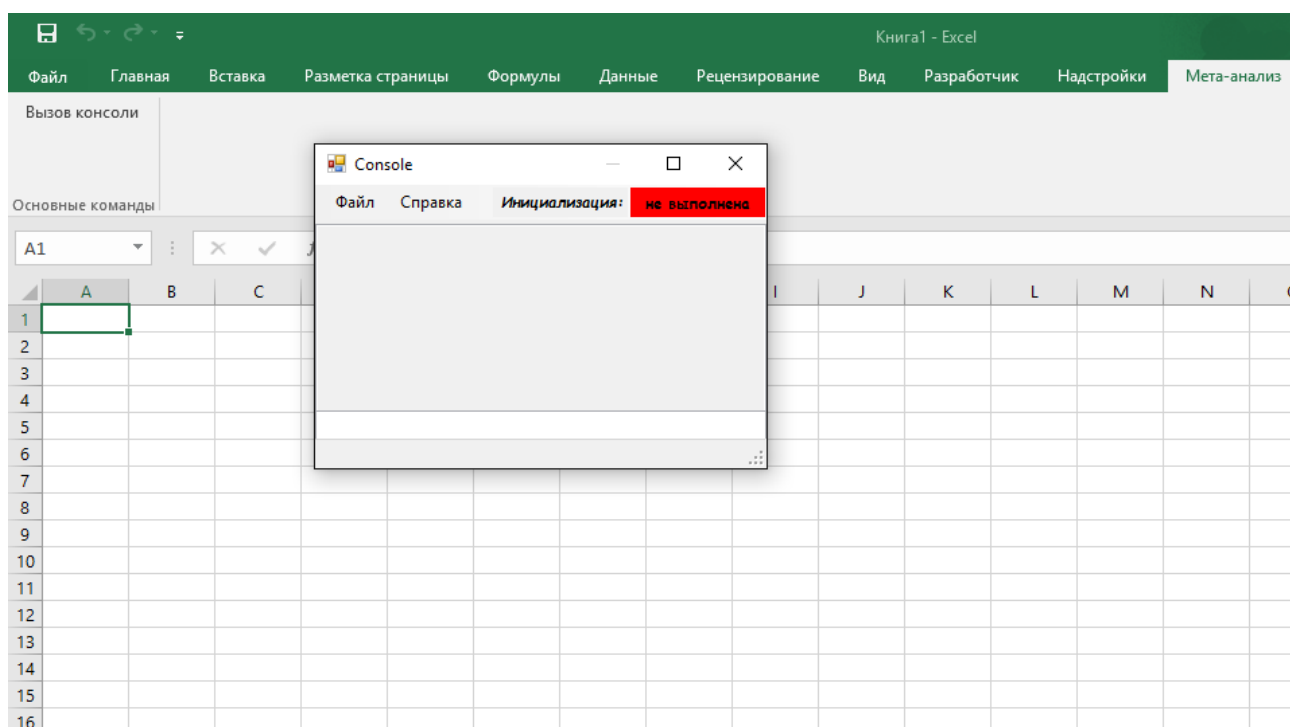


Рисунок 4.5 – Главное консольное окно

Для начала работы с надстройкой, необходимо провести начальную инициализацию будущей таблицы, т.е. заполнить заголовок. Сделать это можно двумя способами: использовать команду в консольном окне `init add` с последовательным перечислением названий столбцов, либо, заполнить названия вручную и выполнить обычную команду `init` с указанием номера столбца. Рекомендуется использовать глобальные имена столбцов пространства Excel (например, “A” или “BA”), хотя допускается использовать номер столбца.

После выполнения начальной инициализации, становятся доступны стандартные команды добавления, удаления, очистки. Так же, в консольном окне присутствует индикатор состояния инициализации.

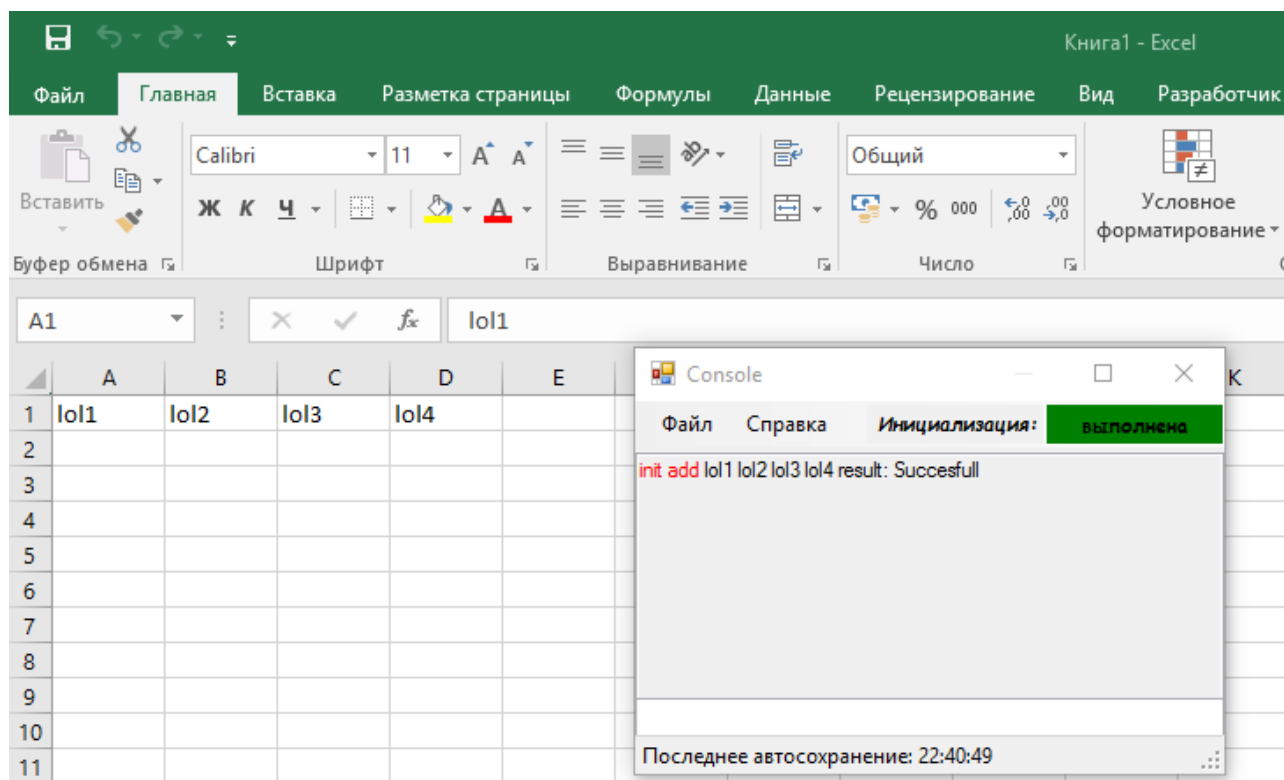


Рисунок 4.6 – Инициализация

Добавим новую строку в таблицу. Для добавления данных используется группа команд `add`, например, `add n lol1 1 lol2 2 lol3 3 lol4 4`.

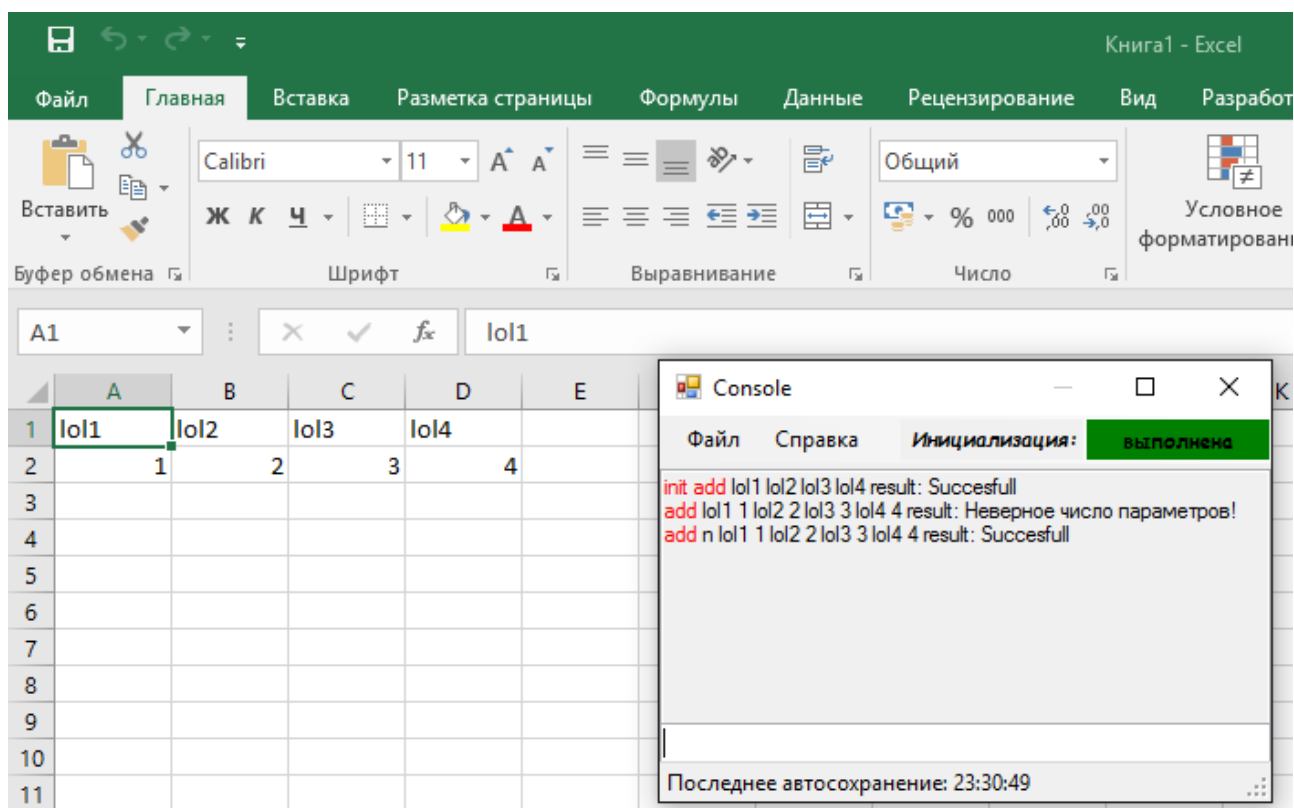


Рисунок 4.7 – Вставка новой строки

Предположим, что названия столбцов, используемые для примера, слишком длинные. Тогда было бы не так удобно заполнять данные. Установим сокращение для столбца с именем lol1 командой `set nick lol1 11` и добавим новую строку.

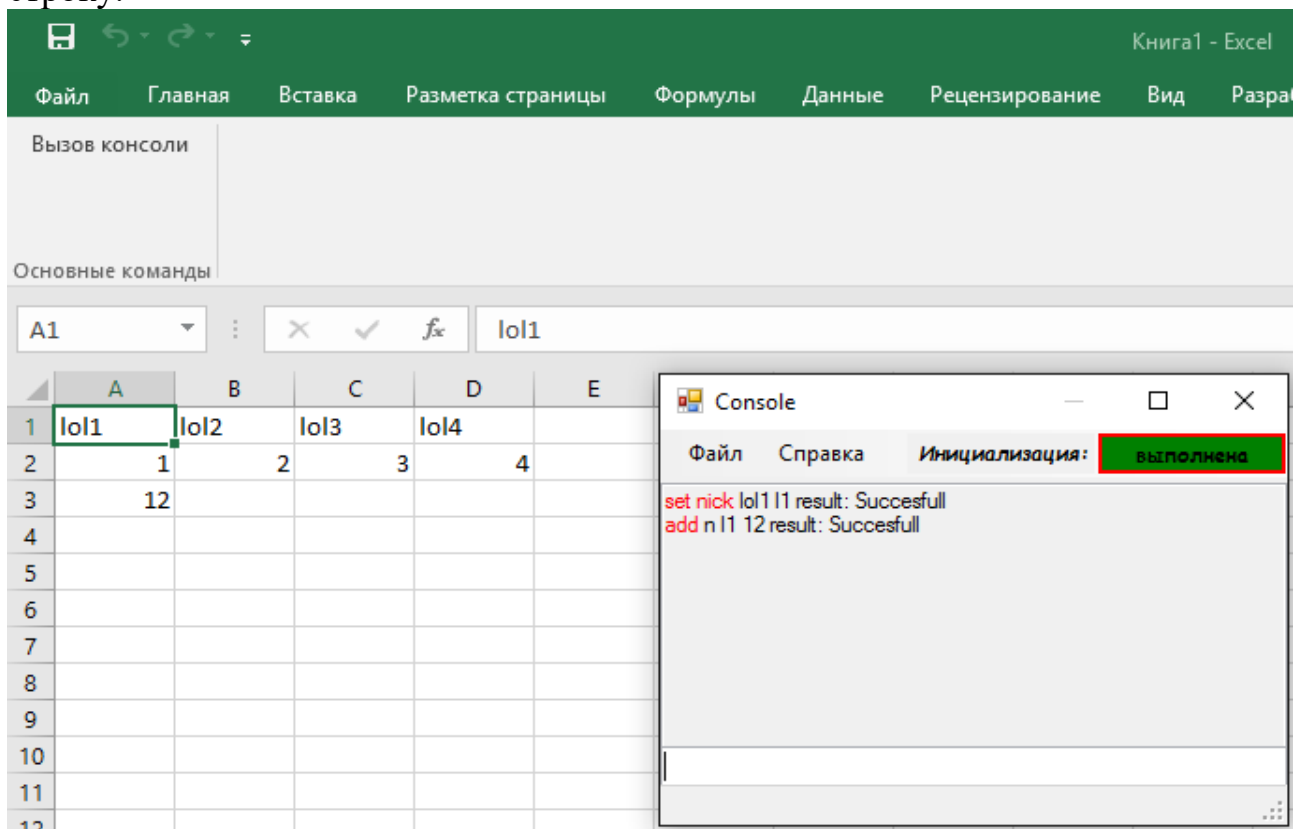


Рисунок 4.8 – Установка сокращения

Добавим несколько новых строк, затем удалим третью строку командой del

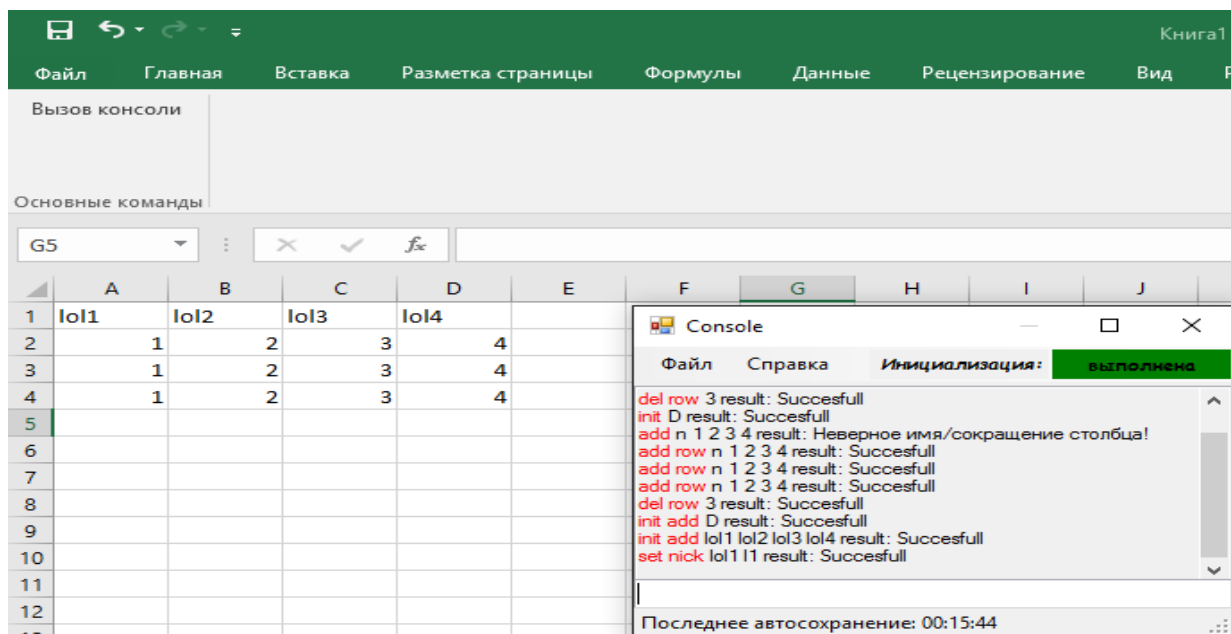


Рисунок 4.9 – Удаление строки

Изменим 2 строку командой add:

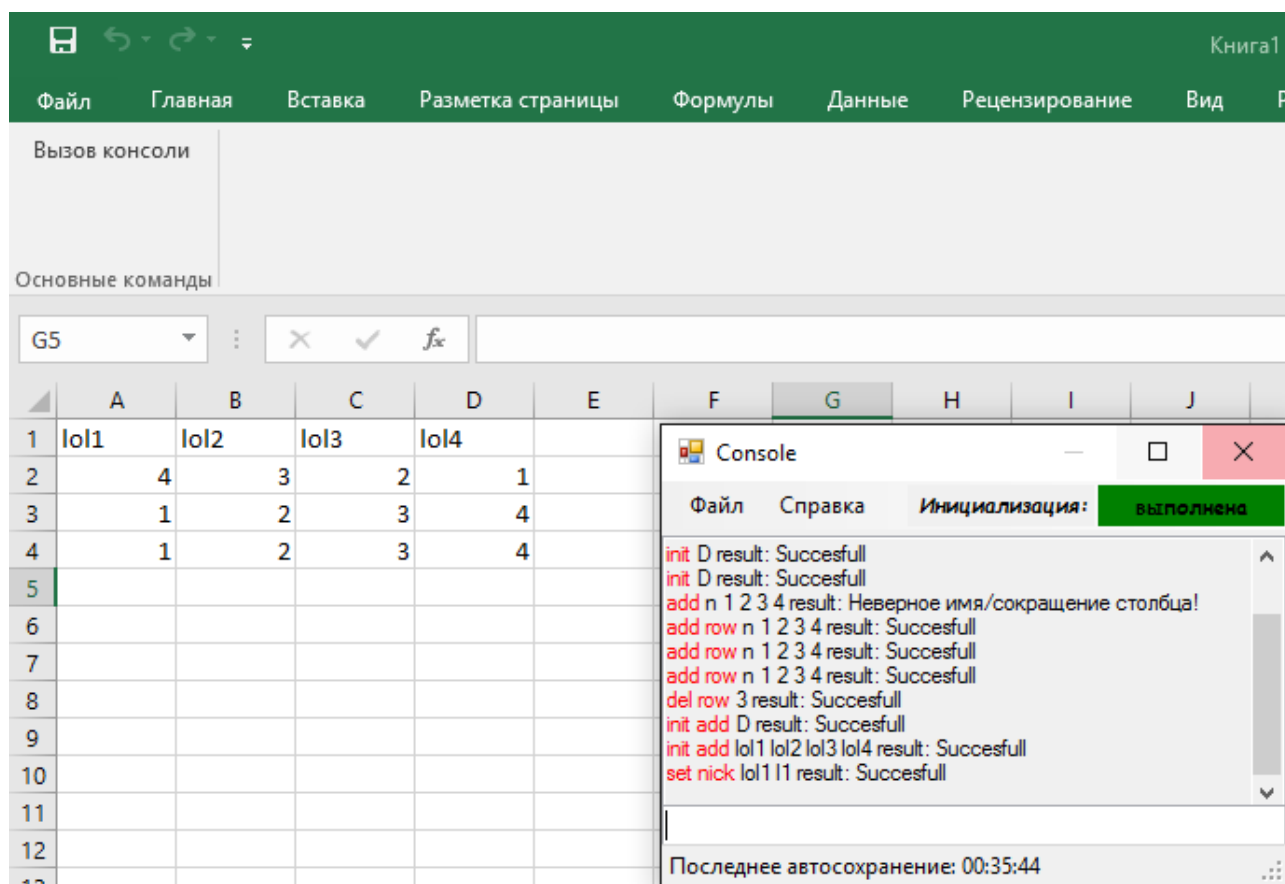


Рисунок 4.10 – Инициализация

Для сохранения текущего заголовка в виде шаблона, воспользуемся командой save и проверим создание нового файла командой list:

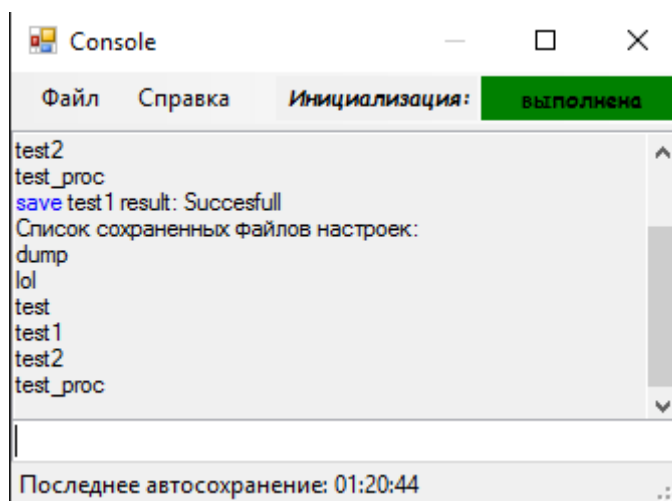


Рисунок 4.10 – Команда list

Создадим новый лист в текущей книге Excel и загрузим сохраненный шаблон командой load:

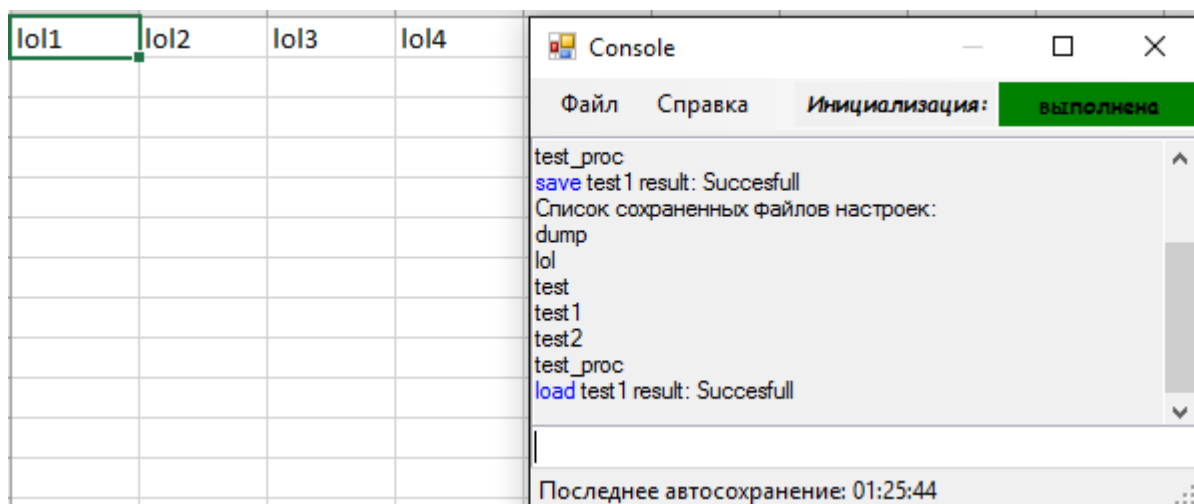


Рисунок 4.11 – Команда load

Можно изменить цвет подсветки команд. Для этого нужно перейти в настройки. Заменим цвет подсветки на синий.

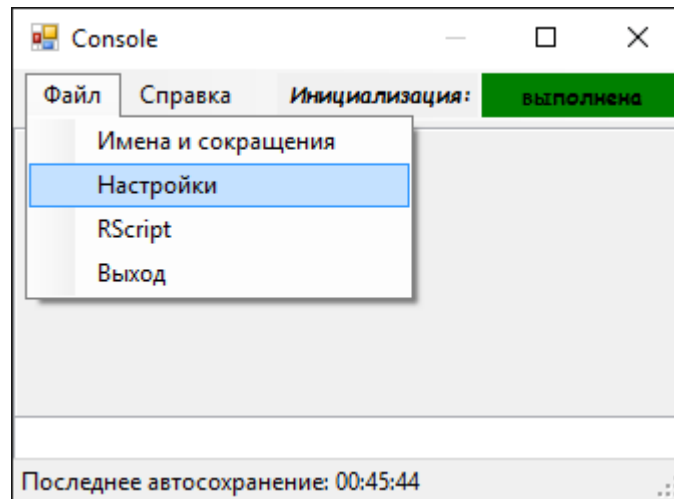


Рисунок 4.12 – Меню

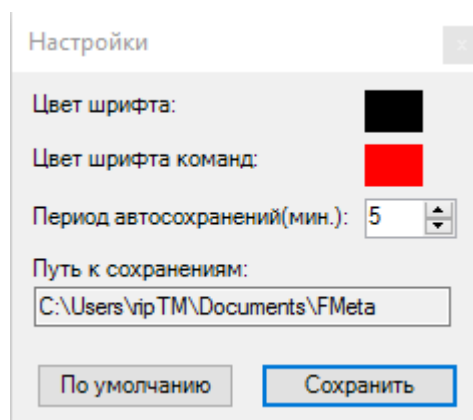


Рисунок 4.13 – Настройки приложения

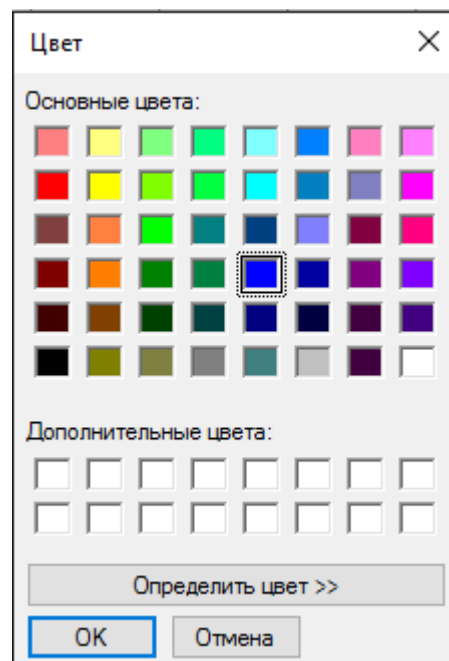


Рисунок 4.14 – Цветовая палитра

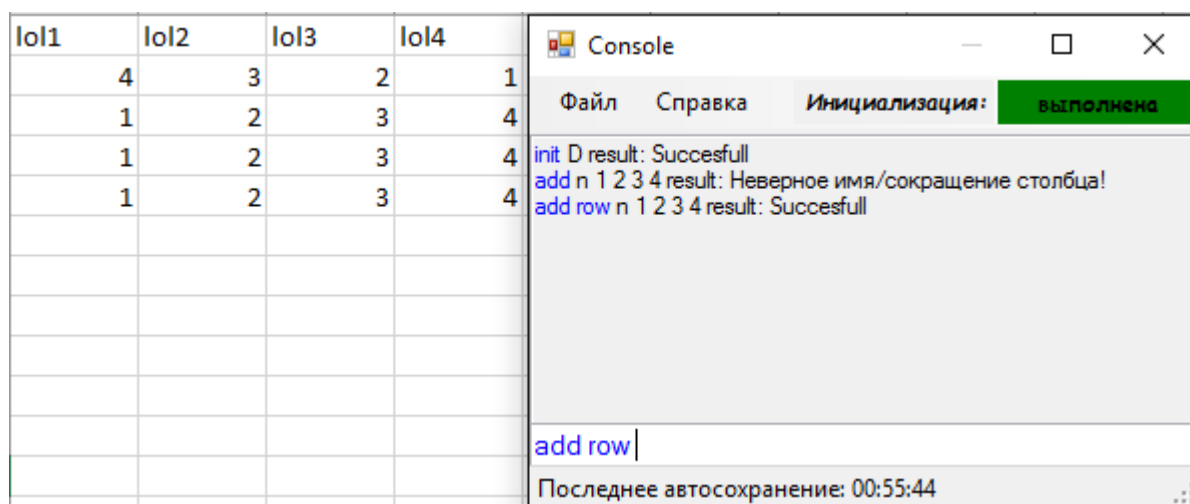


Рисунок 4.14 – Цвет синтаксиса

Выполним простой RScript test\_RProc следующего содержания:

```
l1<-function(x){
  return(x*2)
}
lol1<-as.numeric(lol1)
lol1<-sapply(lol1, l1)
write(lol1, file = "out.txt")
```

Рисунок 4.15 – Тестовый скрипт

Для следующих данных:

| C1 |      |   |
|----|------|---|
|    | A    | B |
| 1  | lol1 |   |
| 2  | 1    |   |
| 3  | 2    |   |
| 4  | 3    |   |
| 5  |      |   |

Рисунок 4.16 – Тестовые данные

Для этого, необходимо в меню «Файл» выбрать пункт RScript, затем, выбрать файл скрипта.



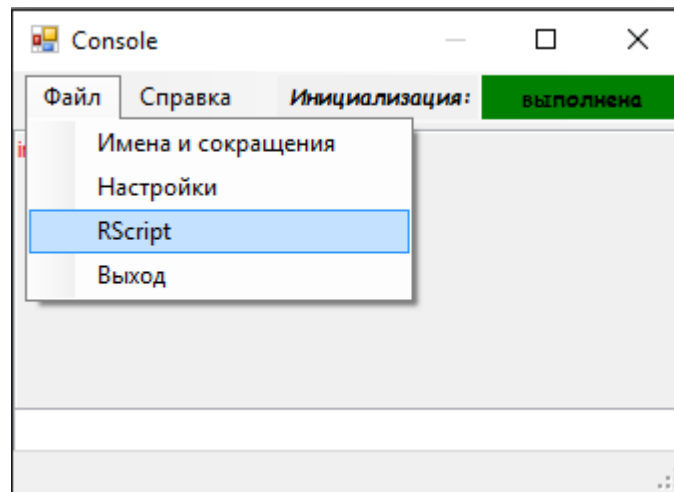


Рисунок 4.17 – Меню RScript

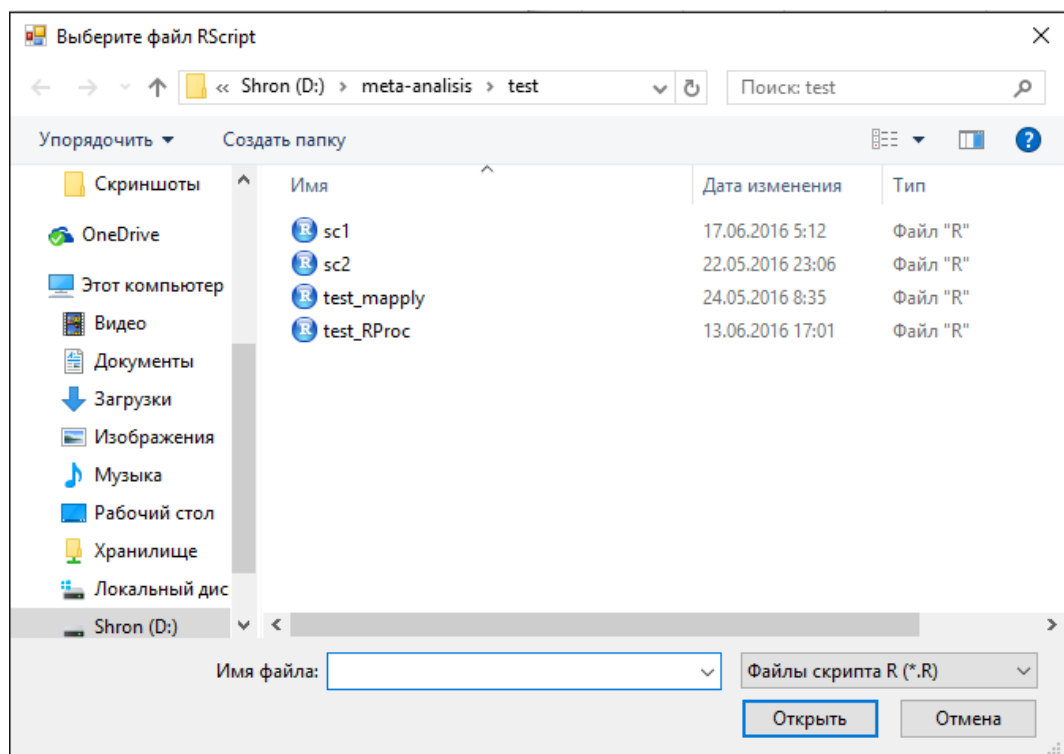


Рисунок 4.18 – Выбор файла скрипта

В результате выполнения, в папке скрипта, создается текстовый файл с результатами выполнения.

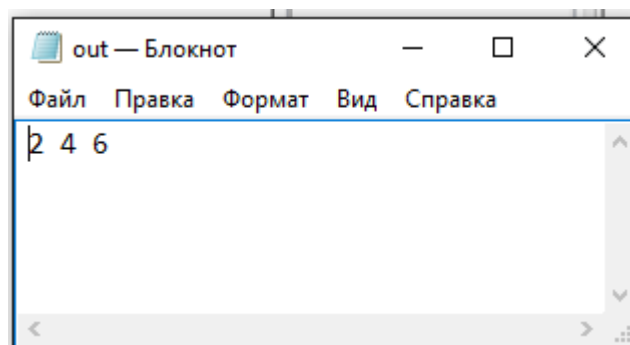


Рисунок 4.19 – Результат выполнения тестового скрипта

Для вызова справки, в которой содержатся все доступные команды и описания к ним, нужно выбрать в меню «Справка» пункт «Вызов справки»:

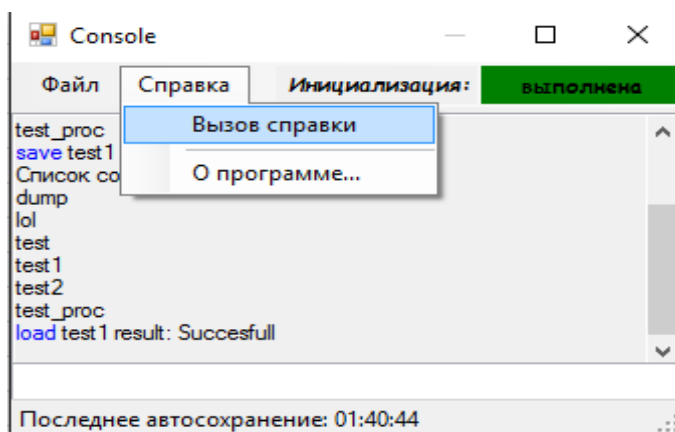


Рисунок 4.20 – Вызов справки

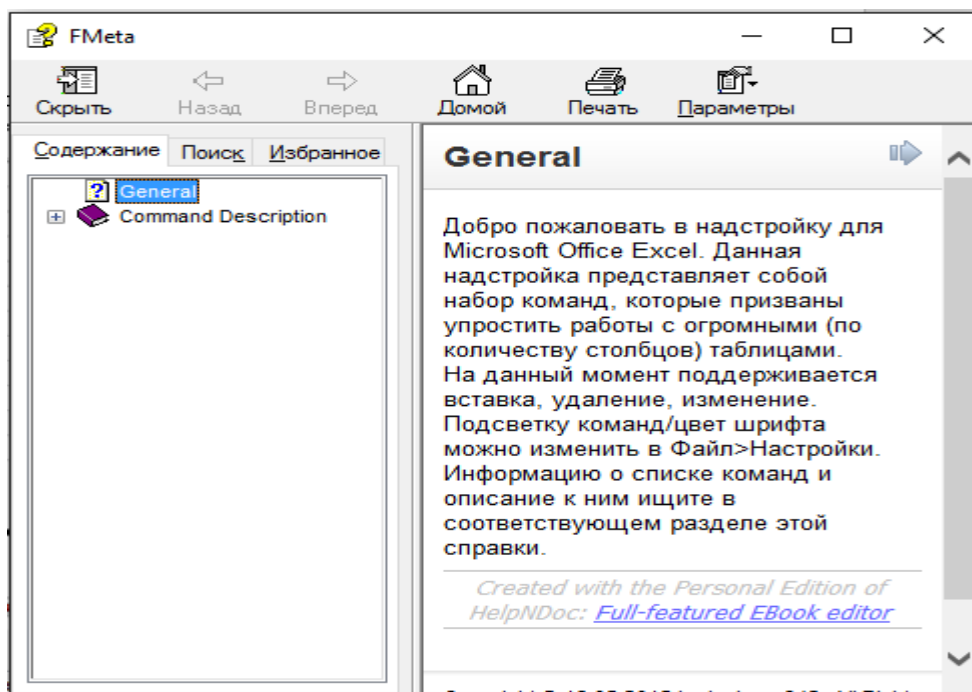


Рисунок 4.21 – Главное окно справки

Для того, чтобы узнать об авторе и версии программы, нужно выбрать в меню «Справка» пункт «О программе...».

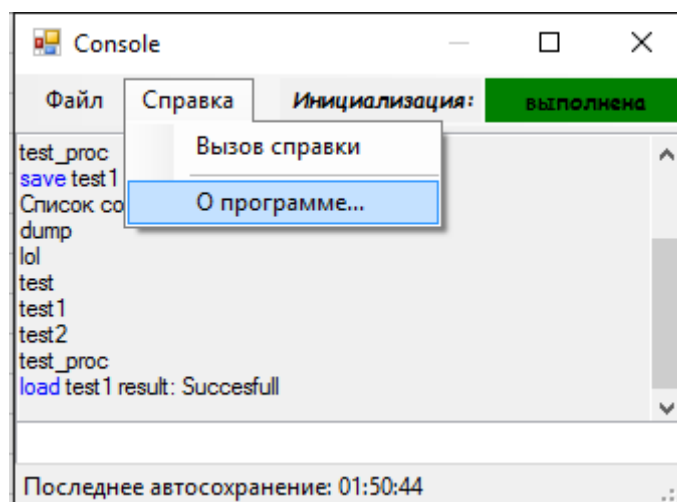


Рисунок 4.22 – Вызов информации о программе

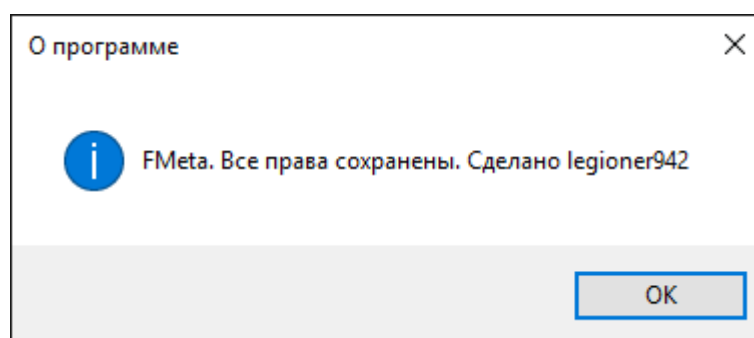


Рисунок 4.23 – Вид информационного окна

## Заключение

В результате выполнения квалификационной работы были достигнуты все поставленные задачи:

- 1) Написана программа на языке R, которая восстанавливает некоторые пропущенные данные, если есть возможность их восстановить;
- 2) Программа на языке R, которая визуализирует в виде столбчатых диаграмм используемые статистические показатели авторами разных статей;
- 3) Была написана надстройка Excel для облегчения заполнения таблицы данными, с возможностью обработать данные внешним скриптом;

Возможность восстанавливать данные, пусть и не всех, серьезно облегчает задачу сравнения эффективности методик лечения. Даже получая после восстановления неточные показатели (а они такими и получаются), уже можно судить с небольшой погрешностью об эффективности той или иной методике лечения, возрастной группе, которая принимает участие в испытаниях и многое, многое другое.

Визуализация данных об использовании статистических показателей разными авторами позволяет лучше представить текущее положение вещей.

Но более всего из всех задач выделяется приложение-надстройка, т.к. она имеет огромные перспективы для развития. Была сформирована идея этого приложения и идея была реализована. Но в дальнейшем, для развития данного приложения необходимо внести следующие изменения:

- 1) Изменить архитектуру. На данный момент архитектура не является достаточно гибкой и удобной
- 2) Изменить интерфейс консоли. Консоль должна стать частью приложения Excel
- 3) Расширить возможности надстройки (например, поиск и выделение ячейки по значению)
- 4) Разделить логику класса Core на отдельные компоненты
- 5) Добавление всплывающих подсказок
- 6) Форма редактирования сокращений и имен столбцов
- 7) Дополнить ленту, отображаемую в Excel

## Список использованных источников

- 1 Мостицкий С., Шитиков В. Статистический анализ и визуализация данных с помощью R. – электронная книга – <http://r-analytics.blogspot.com>
- 2 Троелсен Э., Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд.: Пер. с англ. – М. : ООО «И.Д. Вильямс», 2013. – 1312
- 3 Wikipedia, the free encyclopedia [Электронный ресурс] : Свободная общедоступная многоязычная универсальная энциклопедия. — Режим доступа: [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page). — Загл. с экрана. — яз. англ.
- 4 Stackoverflow на русском [Электронный ресурс] : Сайт вопросов и ответов для программистов. – Режим доступа: <http://ru.stackoverflow.com/>
- 5 MSDN – Developer Network [Электронный ресурс] : сеть разработчиков Microsoft. – Режим доступа: <https://msdn.microsoft.com/ru-ru/dn308572.aspx>