|  |
| --- |
| Министерство образования и науки Российской Федерации ГОУ ВО «северо-кавказский горно-металлургический институт (государственный технологический университет)»  Факультет Информационных технологий  Кафедра Автоматизированной обработки информации  Специальность Информатика и вычислительная техника  Пояснительная  записка  к выпускной квалификационной работе  на тему: Разработка системы оценки социально-экономических показателей СКФО с использованием различных статистических методов.  Студент Алборов Михаил Арсенович  Руководитель проекта Будаева Алина Алибековна  Проект рассмотрен кафедрой и допущен к защите в ГЭК  Заведующий кафедрой проф. Гроппен В.О.  г. Владикавказ 2017 г. |

Федеральное государственное образовательное учреждение высшего образования

**«Северо-Кавказский горно-металлургический институт**

**(государственный технологический университет)»**

Кафедра автоматизированной обработки информации

**УТВЕРЖДАЮ**

Заведующий кафедрой АОИ,

д.т.н., профессор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гроппен В.О.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

**ЗАДАНИЕ**

на выполнение выпускной квалификационной работы бакалавра

студента группы ИВб-13-2 Алборова Михаила Арсеновича,

обучающегося по направлению 09.03.01

«Информатика и вычислительная техника»

профиль «Автоматизированные системы управления»

1. **Тема работы**(утверждена приказом по СКГМИ (ГТУ) от «9» декабря 2016 г. №1632/ОБ)

Разработка системы оценки социально-экономических показателей СКФО с использованием различных статистических методов.

1. **Срок сдачи студентом законченной работы:** 21.06.2017 г.
2. **Исходные данные к работе:**
   1. Статистическая информация по регионам России из РОССТАТ.
3. **Содержание расчетно-пояснительной записки (перечень подлежащих разработкевопросов):**
   1. Введение.
   2. Глава 1. Аналитический обзор.
   3. Глава 2. Выбор и обоснование алгоритма.
   4. Глава 3. Выбор и обоснование программной реализации.
   5. Глава 4. Экспериментальная часть.
   6. Заключение.
   7. Список литературы.
   8. Приложение. Листинг программы.
4. **Перечень графического материала (с указанием обязательных чертежей):**
   1. Цель работы.
   2. Формальная постановка задачи.
   3. Алгоритм работы программы.
   4. Пример решения.
   5. Интерфейс программы (внешний вид).
   6. Результаты экспериментального тестирования: экспериментальные данные (графики).
5. **Дата выдачи задания:**«9» декабря 2016 г.

**Научный руководитель**

к.т.н., доц. кафедры АОИ Будаева А. А.

**Задание принял к исполнению** Алборов М.А.

Реферат

Пояснительная записка: 87 стр.,9 таб., 20 рис.,1 прил.

**Ключевые слова:** Социально-экономические показатели, метод эталонов, ранжирование, нормирование, расстояние между объектами.

Объект разработки: Программное обеспечение, визуально представляющее динамику изменения регионов со временем по различным показателям.

**Цель работы:** Разработка системы оценки социально-экономических показателей СКФО с использованием различных статистических методов.

Использованное прикладное и системное программное обеспечение:

* Операционная система MicrosoftWindows7 64-bit;
* Средаразработки Microsoft Visual Studio 2010;

Типы используемых вычислительных средств:при написании дипломного проекта использовался персональный компьютер со следующими характеристиками:

Intel (R) Celeron (R)1.50 GHz 1.50 GHz,ОЗУ 8ГБ, Intel (R) HD Graphics, ОС Windows 7 –64 bit.

Полученные результаты:

В ходе выпускной работы проведен аналитический обзор, предложен алгоритм метода эталонов, выбрана математическая модель, разработан программный комплекс, реализующий систему оценки социально-экономических показателей СКФО. Для оценки эффективности использования программного комплекса была проведена серия экспериментов, подтвердивших эффективность программного продукта.

**Оглавление**

[Введение 6](#_Toc485381355)

[Глава 1. Аналитический обзор 7](#_Toc485381356)

[1.1. Методы статистического исследования 9](#_Toc485381357)

[1.2. Метод статистического наблюдения 12](#_Toc485381358)

[1.3. Методика оценки эффективности социально-экономического развития региона 13](#_Toc485381359)

[1.4. Нормирование 22](#_Toc485381360)

[1.4.1. Нормировка показателей 22](#_Toc485381361)

[1.4.2. Типы показателей 23](#_Toc485381362)

[1.4.3.  Нормировка униполярного показателя 23](#_Toc485381363)

[1.4.4. Нормировка биполярного показателя 24](#_Toc485381364)

[1.4.5. Особенности балльных шкал 25](#_Toc485381365)

[1.5. Аналитические пакеты прикладных программ 27](#_Toc485381366)

[Глава 2. Выбор и обоснование алгоритма 31](#_Toc485381367)

[2.1. Содержательная постановка 31](#_Toc485381368)

[2.2. Формальная постановка задачи 33](#_Toc485381369)

[2.3. Метод эталонов 34](#_Toc485381370)

[2.4. Алгоритм 35](#_Toc485381371)

[2.5. Пример решения 36](#_Toc485381372)

[Глава 3. Выбор и обоснование программной реализации 44](#_Toc485381373)

[3.1. Язык программирования C# 44](#_Toc485381374)

[3.2. Интерфейс программы 45](#_Toc485381375)

[Глава 4. Экспериментальная часть 53](#_Toc485381376)

[Заключение 56](#_Toc485381377)

[Список использованной литературы 57](#_Toc485381378)

[Приложение. Листинг программы 58](#_Toc485381379)

# Введение

Модернизация системы управления региональными социоэкономическими системами неразрывно связана с определением оптимальных параметров, позволяющих проводить оценку социально-экономического состояния и динамики развития регионов. В условиях функционирования региона как открытой, динамичной социальной системы, изменений, происходящих во всех блоках-компонентах этих систем (институциональном, хозяйственно-экономическом, природно-экологическом, социальном) весьма актуализируется проблема разработки сбалансированной системы показателей и методики анализа и оценки социально-экономической эффективности развития региона.

Анализ теоретико-методологических подходов к определению эффективности позволяет сделать вывод о том, что, как правило, выделяют экономическую эффективность и социальную. Вместе с тем, самостоятельность данных видов эффективности относительна, так как они находятся в тесном единстве и взаимосвязи.

Поэтому актуальна цель работы –Разработка системы оценки социально-экономических показателей с использованием различных статистических методов.

Поставленная цель потребовала решения следующих задач:

* анализ существующих подходов к оценке социально-экономических показателей;
* выбор математической модели оценки и алгоритма для ее решения;
* программная реализация подсистемы;
* тестирование и отладка.

# Глава 1. Аналитический обзор

Полная и достоверная статистическая информация является тем необходимым основанием, на котором базируется процесс управления экономикой. Вся информация, имеющая народнохозяйственную значимость, в конечном счете, обрабатывается и анализируется с помощью статистики.

Именно статистические данные позволяют определить объемы валового внутреннего продукта и национального дохода, выявить основные тенденции развития отраслей экономики, оценить уровень инфляции, проанализировать состояние финансовых и товарных рынков, исследовать уровень жизни населения и другие социально-экономические явления и процессы. Овладение статистической методологией - одно из условий познания конъюнктуры рынка, изучения тенденций и прогнозирования, принятия оптимальных решений на всех уровнях деятельности.

Статистическая наука - это отрасль знаний, изучающая явления общественной жизни с их количественной стороны в неразрывной связи с их качественным содержанием в конкретных условиях места и времени. Статистическая практика - это деятельность по сбору, накоплению, обработке и анализу цифровых данных, характеризующих все явления в жизни общества.

Говоря о статистике следует помнить, что цифры в статистике не абстрактные, а выражают глубокий экономический смысл. Каждый экономист должен уметь пользоваться статистическими цифрами, анализировать их, уметь использовать для обоснования своих выводов.

Статистические законы действуют в пределах времени и места, в которых они обнаружены.

Окружающий мир состоит из массовых явлений. Если отдельный факт зависит от законов случая, то масса явлений подчиняется закономерностям. Для обнаружения этих закономерностей используется закон больших чисел.

Для получения статистической информации органы государственной и ведомственной статистики, а также коммерческие структуры проводят различного рода статистические исследования. Процесс статистического исследования включает три основные стадии: сбор данных, их сводка и группировка, анализ и расчет обобщающих показателей.

От того, как собран первичный статистический материал, как он обработан и сгруппирован в значительной степени зависят результаты и качество всей последующей работы, а в конечном итоге при нарушениях могут привести к абсолютно ошибочным выводам.

Сложной, трудоемкой и ответственной является заключительная, аналитическая стадия исследования. На этой стадии рассчитываются средние показатели и показатели распределения, анализируется структура совокупности, исследуется динамика и взаимосвязь между изучаемыми явлениями и процессами.

На всех стадиях исследования статистика использует различные методы. Методы статистики - это особые примы и способы изучения массовых общественных явлений.

На первой стадии исследования применяются методы массового наблюдения, собирается первичный статистический материал. Основное условие - массовость, т.к. закономерности общественной жизни проявляются в достаточно большом массиве данных в силу действия закона больших чисел, т.е. в сводных статистических характеристиках случайности взаимопогашаются.

На второй стадии исследования, когда собранная информация подвергается статистической обработке, используется метод группировок. Применение метода группировок требует непременного условия - качественной однородности совокупности.

На третьей стадии исследования проводится анализ статистической информации с помощью таких методов как метод обобщающих показателей, табличный и графический методы, методы оценки вариации, балансовый метод, индексный метод.

Аналитическая работа должна содержать элементы предвидения, указывать на возможные последствия складывающихся ситуаций.

## Методы статистического исследования

Между наукой-статистикой и практикой существует тесная взаимосвязь: статистика использует данные практики, обобщает и разрабатывает методы проведения статистических исследований. В свою очередь в практической деятельности применяются теоретические положения статистической науки для решения конкретных управленческих задач. Знание статистики необходимо современному специалисту для принятия решений в условиях стохастики (когда анализируемые явления подвержены влиянию случайностей), для анализа элементов рыночной экономики, в сборе информации, в связи с увеличением числа хозяйственных единиц и их типов, аудите, финансовом менеджменте, прогнозировании.

Для изучения предмета статистики разработаны и применяются специфические приемы, совокупность которых образует методологию статистики (методы массовых наблюдений, группировок, обобщающих показателей, динамических рядов, индексный метод и др.). Применение в статистике конкретных методов предопределяется поставленными задачами и зависит от характера исходной информации. При этом статистика опирается на такие диалектические категории, как количество и качество, необходимость и случайность, причинность, закономерность, единичное и массовое, индивидуальное и общее. Статистические методы используются комплексно (системно). Это обусловлено сложностью процесса экономико-статистического исследования, состоящего из трех основных стадий: • первая - сбор первичной статистической информации; • вторая - статистическая сводка и обработка первичной информации; • третья - обобщение и интерпретация статистической информации.

Общей методологией изучения статистических совокупностей является использование основных принципов которыми руководствуются в любой науке. К этим принципам, как к своего рода началам относятся следующие:

1. объективность изучаемых явлений и процессов;

2. выявление взаимосвязи и системности в которых проявляется содержание изучаемых факторов;

3. целеполагание, т.е. достижение поставленных целей со стороны исследователя, изучающего соответствующие статистические данные.

Это выражается в получении сведений о тенденциях, закономерностях и возможных последствиях развития изучаемых процессов. Знание закономерностей развития социально-экономических процессов, интересующих общество, имеет важное практическое значение.

К числу особенностей статистического анализа данных следует отнести метод массового наблюдения, научной обоснованности качественного содержания группировок и его результатов, вычисление и анализ обобщенных и обобщающих показателей изучаемых объектов.

Что касается конкретных методов экономической, промышленной или статистики культуры, населения, национального богатства и т.п., то здесь могут быть свои специфические методы сбора, группировки и анализа соответствующих совокупностей (суммы фактов).

В экономической статистике, например, широко применяется балансовый метод как наиболее распространенный метод взаимной увязки отдельных показателей в единой системе экономических связей в общественном производстве. К методам применяемым в экономической статистике также относятся составление группировок, исчисление относительных показателей (процентное соотношение), сравнения, исчисление различных видов средних величин, индексов и т.п.

Метод связующих звеньев состоит в том, что два объемных, т.е. количественных показателя сопоставляются на основе существующего между ними отношения. Например, производительность труда в натуральных показателях и отработанного времени, или объем перевозок в тоннах и средней дальности перевозок в км.

При анализе динамики развития народного хозяйства основным методом выявления этой динамики (движения) является индексный метод, методы анализа временных рядов.

При статистическом анализе основных экономических закономерностей развития народного хозяйства важным методом статистики является вычисление тесноты связей между показателями с помощью корреляционного и дисперсионного анализа и др.

Кроме названных методов широкое распространение получили математико-статистические методы исследования которые расширяются по мере движения масштабов применения ЭВМ и создания автоматизированных систем.

Этапы статистического исследования:

1. Статистическое наблюдение – массовый научно организованный сбор первичной информации об отдельных единицах изучаемого явления.

2. Группировка и сводка материала – обобщение данных наблюдения для получения абсолютных величин (учетно-оценочных показателей) явления.

3. Обработка статистических данных и анализ результатов для получения обоснованных выводов о состоянии изучаемого явления и закономерностях его развития.

Все этапы статистического исследования тесно связаны друг с другом и одинаково важны. Недостатки и ошибки, возникающие на каждой стадии, сказываются на все исследовании в целом. Поэтому правильное использование специальных методов статистической науки на каждом этапе позволяет получить достоверную информацию в результате статистического исследования.

Методы статистического исследования:

1. Статистическое наблюдение

2. Сводка и группировка данных

3. Расчет обобщающих показателей (абсолютные, относительные и средние величины)

4. Статистические распределения (вариационные ряды)

5. Выборочный метод

6. Корреляционно-регрессионный анализ

7. Ряды динамики

8. Индексы

Задача статистики – исчисление статистических показателей и их анализ, благодаря чему управляющие органы получают всестороннюю характеристику управляемого объекта, будь то вся национальная экономика или отдельные ее отрасли, предприятия и их подразделения. Управлять социально-экономическими системами нельзя, не располагая оперативной, достоверной и полной статистической информацией.

## Метод статистического наблюдения

Статистическое наблюдение - это планомерный, научно-организованный и, как правило, систематический сбор данных о явлениях общественной жизни. Оно осуществляется путем регистрации заранее намеченных существенных признаков с целью получения в дальнейшем обобщающих характеристик этих явлений.

Например, при проведении переписи населения о каждом жителе страны записываются сведения о его поле, возрасте, семейном положении, образовании и др., а затем статистические органы определяют на основе этих сведений численность населения страны, его возрастную структуру, размещение по территории страны, семейный состав и другие показатели.

К статистическому наблюдению предъявляются следующие требования: полнота охвата изучаемой совокупности, достоверность и точность данных, их однообразие и сопоставимость.

## Методика оценки эффективности социально-экономического развития региона

В современных условиях целью исследования в области оценки эффективности социально-экономического развития субъектов Российской Федерации является поиск наиболее эффективных методов расчета, позволяющих с максимальной точностью и минимальными затратами оценить различные варианты социально-экономического развития региона.

При оценке эффективности социально-экономического развития региона систем региональных процессов должен учитываться весь комплекс взаимоотношений внутри и между региональными системами, все существующие взаимосвязи.

В условиях трансформирующейся социально-экономической системы Российской Федерации существует необходимость оценки эффективности и адекватности управляющих воздействий федеральных, региональных муниципальных органов власти на региональные процессы. При этом социально-экономическую систему следует рассматривать как систему, включающую элементы экономики и функционирующую с участием людей. В числе таких форм общественного устройства - мировое хозяйство, социалистическая и капиталистическая системы, системы хозяйствования регионального и национального уровня; в качестве подобных систем могут рассматриваться также территориально-производственные комплексы, отрасли, объединения, предприятия и т.п.

Необходимо помнить, что социально-экономическая система представляет собой большую систему, характеризующуюся многообразием составляющих ее элементов, которые можно объединить в достаточно широкие подсистемы - экономическую (производственно-технологическую) и социальную. Ввиду глобальности подсистем в литературе они, как правило, называются системами. Необходимо отметить, что границы между этими подсистемами несколько размыты. В различных социально-экономических системах могут быть серьезные отличия по форме их взаимоотношений, мотиваций, потребностей и целей, оценочной меры выделенных приоритетов и степени важности.

Таким образом, проблемы социально-экономического развития регионов находятся в тесной зависимости от экономических, социальных, политических и других процессов, происходящих как в самих субъектах, так и в Российской Федерации в целом.

Методика оценки эффективности социально-экономического развития региона должна быть сформирована с учетом подхода экономико-математического моделирования социально-экономических систем и общих принципов системного анализа, при котором функциональные взаимосвязи субъекта Российской Федерации рассматриваются как подсистемные (в свою очередь, сами подсистемы можно рассматривать как самостоятельные системы) взаимодействия общей социально-экономической инфраструктуры региона.

При формировании методики оценки эффективности социально-экономического развития региона и в дальнейшем выбора эффективных методов воздействия на социально-экономическое развитие региона необходимо объективно определиться с приоритетным направлением развития субъекта Российской Федерации. Определение приоритетов в социально-экономическом развитии региона позволит качественно определить структуру оцениваемых показателей.

Необходимо отметить, что определением направления развития региона и обеспечением его эффективного развития занимаются исследователи во многих странах. Цели и методология этих исследований, в сущности, не зависят от уровня развития страны, т.к. рассматривается эффективное использование ограниченных ресурсов.

М. Уандыкова утверждает, что «...большинство авторов рассматривают экономическую систему в узком смысле и поэтому считают основной проблемой, решаемой ею, оптимальное распределение ограниченных ресурсов с целью максимального удовлетворения неограниченных потребностей людей и общества в целом».

Необходимо учитывать, что за последнее время в общественном сознании произошел большой скачок, который характеризуется глубоким осознанием того, что именно социальные цели объединяют людей в рамках страны (государства), тогда как экономические цели могут разъединять, а не объединять людей.

Таким образом, следует рассматривать не отдельно экономическую систему, а социально-экономическую систему как симбиоз социальной и производственной сфер жизнедеятельности страны.

Исходя из вышеизложенного, при формировании методики оценки эффективности социально-экономического развития региона необходимо определиться с рядом принципов, позволяющих рассматривать регион как систему.

Основными принципами оценки эффективности социально-экономического развития региона, по нашему мнению, являются:

1) принцип системности - необходимость анализа социально-экономического потенциала региона как взаимосвязанных элементов: коммерческих и некоммерческих организаций [5], органов федеральной, региональной власти и местного самоуправления и т.д. Принцип системности выступает как одна из граней диалектической философии, как конкретизация и развитие диалектического метода. Так, по мнению В. И. Ленина (1870-1924), «.чтобы действительно знать предмет, надо охватить, изучить все его стороны, все связи и опосредствования. Мы никогда не достигаем этого полностью, но требование всесторонности предостерегают нас от ошибок...»;

2) принцип комплексности - оценка эффективности социальноэкономического развития региона, осуществляется с учетом всех внешних и внутренних факторов;

3) принцип долговременности - результаты оценки должны учитывать принципы стратегического развития региона и определять его положение в будущем;

4) принцип сопряженности - рассматривать социально-экономическое развитие региона как составную часть социально-экономического развития страны, с учетом федеративных особенностей;

5) принцип непрерывной изменчивости - результаты оценки эффективности социально-экономического развития региона применимы только на ограниченном временном промежутке;

6) принцип контринтуитивного поведения сложных систем Форрестера -результаты оценки эффективности социально-экономического развития региона, осуществляется до тех пор, пока полностью не согласуется с идеями автора;

7) принцип устойчивого неравновесия - точность оценки эффективности социально-экономического развития региона зависит от степени динамичности отдельных параметров, обеспечивающих данное развитие;

8) принцип достаточности - выбор оцениваемых показателей осуществляется с учетом степени влияния на уровень экономического роста региона. Данный принцип согласуется с методом Альфреда Маршалла (1842-1924) -методом частичного равновесия: «. мы исключаем влияние всех других факторов оговоркой «при прочих равных условиях», хотя и не считаем их инертными, а лишь временно игнорируем их действие». При этом уместно вспомнить закон достаточного основания, сформулированный Готфридом Лейбницем (1646-1716): «.достаточное основание есть положение (или совокупность положений), которое является заведомо истинным и из которого логически вытекает обосновываемое положение»;

9) принцип соответствия - обусловлен применением принципа достаточности и позволяет использовать выявленную закономерность изменения достаточных показателей в целом для всей системы «регион»;

10) принцип оптимальности - «задача заключается не в том, чтобы найти решение лучше существующего, а в том, чтобы найти самое лучшее решение из всех возможных» [9];

11) принцип эмерджентности - принцип системного анализа, выражает следующее важное свойство системы: чем больше система и чем больше различие в размерах между частью и целым, тем выше вероятность того, что свойства целого могут сильно отличаться от свойств частей. Данный принцип подчеркивает возможность несовпадения локальных оптимумов целей отдельных частей с глобальным оптимумом цели системы [10];

12) принцип формализации (формальный - относящийся к форме, в противоположность сущности, т.е. несущественный), нацелен на получение количественных и комплексных характеристик.

В соответствии с вышеизложенными принципами, при анализе эффективности социально-экономического развития региона необходимо выделить две взаимосвязанные и взаимодополняющие подсистемы - донорную и акцепторную. Первая подсистема формирует инфраструктуру для эффективного использования ограниченных ресурсов и производства регионального совокупного общественного продукта, а вторая -обеспечение условий для развития первой подсистемы. При этом необходимо учитывать системный принцип двух взаимосвязанных и взаимодополняющих подсистем, при котором невозможно (и не имеет смысла) определить наиболее значимую подсистему. Донорная подсистема (материальное производство) формирует условия для развития акцепторной подсистемы - социальной инфраструктуры, обеспечивающей качество жизни человеческого общества.

В этом взаимодействии двух подсистем, среди которых синергетические связи имеют решающее значение, происходит качественная трансформация территориального образования в новую форму рыночного субъекта. При этом возникает необходимость оценки системной эффективности взаимодействия двух подсистем.

Рассматривая эффективность социально-экономического развития региона с позиции системы, необходимо проанализировать содержание затрат и результатов производства регионального совокупного общественного продукта, а также учитывать факторы, обеспечивающие качественные условия жизнедеятельности населения данного территориального образования. Таким образом, эффективность социально-экономического развития региона выражает качество общественно-производственных отношений между обществом и регионом, а также отношения внутри региона между двумя подсистемами, донорной и акцепторной, по поводу создания материальной и духовной основы для эффективного социальноэкономического развития региона.

Следует отметить, что эффективность социально-экономического развития региона отражает уровень вовлечения экономических и социальных ресурсов в хозяйственный оборот, т.е. степень использования социально-экономического потенциала. Для оценки эффективности социально-экономического развития региона могут быть использованы системообразующие показатели: объем произведенных товаров и услуг, валовой региональный продукт, показатели эффективности производственнохозяйственной деятельности, характеристики уровня жизни населения и демографической ситуации.

Необходимо отметить, что оценка эффективности социально-экономического развития регионов по полной системе показателей является сложной и трудоемкой задачей, т.к. регион является сложной и большой социальноэкономической системой.

Таким образом, использование формализованных методов оценки эффективности социально-экономического развития региона является основой для систематизации факторов, обеспечивающих синергетический эффект при взаимодействии с региональной инфраструктурой.

Методика предлагает комплексное исследование социально-экономического состояния регионов России и предполагает последовательное прохождение следующих этапов:

1.         Выявление факторов, характеризующих различные сферы социально-экономического положения регионов: производственный, финансовый, инновационный, демографический и человеческий потенциалы развития;

2.         Отбор показателей, входящих в состав потенциалов развития;

3.         Рассмотрение региональной дифференциации на основе кластерного анализа, либо других методов оценки.

Вследствие рассмотрения большого числа признаков и территорий в исследовании социально-экономического состояния регионов, для наиболее подробной интерпретации результатов использовался метод многомерного анализа — кластерный анализ, предназначенный для группировки совокупности элементов, которые характеризуются многими факторами, и получения однородных групп (кластеров). Решение задачи заключается в определении естественного расслоения статистических характеристик на чётко выраженные кластеры, лежащие друг от друга на некотором расстоянии (условно рассчитываемый показатель – евклидово расстояние) и содержащие в себе члены, находящиеся на различном (евклидовом) расстоянии от условного центра кластера. Эффективное решение задачи поиска кластеров, требующей выполнения большого числа вычислительных и логических операций, стало возможным с развитием программного обеспечения, поэтому инструментом реализации выступала программа «STATISTICA» .

В результате кластерного анализа регионы были разделены на однородные группы по уровню благополучия социально-экономического положения и определены территории - «ключи». С использованием картографического метода с помощью ГИС-программы «MapInfo Professional» провели преобразование и отображение результатов полученной типологии. В итоге, был построен комплекс карт, отражающих современное состояние социально-экономического положения регионов России.

Несмотря на то, что используемый метод оценки социально-экономического развития регионов является комплексным, т. е. рассматривает развитие как сложную социально-экономическую категорию, состоящую из блоков, характеризующих отдельные сферы жизнедеятельности человека. Однако в полной мере отразить реальное социально-экономическое положение не удается, так как некоторые данные, получаемые от органов официальной статистики, вызывают недоверие. Поэтому наше исследование потребовало использования социологических методов анализа изучаемого процесса, для сглаживания таких неточностей и получения достаточно объективной картины.

В рамках социологического исследования была разработана анкета, для опроса населения, основной целью, которой было выявление уровня социальной, экономической, политической и др. активности жителей регионов России, так как, на наш взгляд, именно активная жизненная позиция граждан может послужить условием эффективного развития территории.

Генеральную совокупность исследования составило трудоспособное население в регионах — «ключах», выявленных методом кластерного анализа. Выборка респондентов осуществлялась методом «снежного кома», т. е. путём выявления следующих опрашиваемых через уже опрошенных, их знакомых или знакомых интервьюеров и т. д.

Кроме того, был использован метод экспертной оценки. В качестве экспертов были привлечены высококвалифицированные специалисты в области экономики, политики и других сфер деятельности. Исследование, направленное на определение мнения экспертов об уровне социально-экономического положения в том или ином субъекте, проводилось методом анкетирования. Респонденту предлагалось ответить на перечень вопросов о современном состоянии, о проблемах и перспективах развития различных регионов страны. На наш взгляд, данный способ дает дополнительные сведения об изучаемом процессе, помогает понять и интерпретировать результаты, полученные при использовании объективных методов оценки социально-экономического состояния регионов на основе официальной статистики.

Полученные данные в ходе социологического опроса были проанализированы с помощью автоматизированного комплекса обработки результатов социологических исследований «Анкета», созданного авторским коллективом географического факультета Ставропольского государственного университета . Преимущество использования данного комплекса заключается в возможности формирования системы вопросов и перечня ответов к ним, обеспечении безопасного хранения анкетных данных, а также в получении всестороннего анализа обрабатываемых материалов.

В итоге, проведенный анализ позволил определить наиболее острые проблемы в социально-экономическом развитии субъектов Российской Федерации и наметить некоторые пути их решения. Составить рекомендации по развитию территории с учетом территориальных особенностей, направленные на решение самых насущных проблем.

## Нормирование

### 1.4.1. [Нормировка показателей](https://psytest.wordpress.com/data_treatment/normalization_indicator/)

Обычно выраженность некоторого качества пытаются описать числом. Чаще всего такое число х формируется как сумма баллов. Насколько это правомерно — вопрос другой. Мы же предположим, что такое число х получено и осмысленно.

Обычно х меняется от некоторого минимального значения xmin (отражающего отсутствие качества) до некоторого максимального значения xmax (крайняя степень проявления, наличия, выраженности).

Его получение решает проблему сравнения двух объектов, но только по этому показателю. Впрочем, и здесь дело не очень хорошо. Надо всегда помнить, в каких пределах меняется показатель. А эти диапазоны — самые разнообразные.

Конечно, нельзя сравнивать качества непосредственно. Для этого сравниваемые числа должны быть безразмерными. Показатель обычно интерпретируется как **степень выраженности**некоторого качества. И вот это сравнивать можно. Но для этого их следует привести к одной шкале так, чтобы начала и концы двух шкал совпадали.

Такое преобразование для всех показателей и называется нормировкой . После этого мы можем сравнивать разнообразные показатели, полученные различными методиками.

### 1.4.2. Типы показателей

При всем разнообразии числовых характеристик объектов (или респондентов) из них можно выделить два широких класса:

* **униполярные**, выражающие только степень наличия (интенсивность, выраженность, …) некоторого качества;
* **биполярные**, отражающие не только степень наличия качества, но и его «направленность».

### 1.4.3. Нормировка униполярного показателя

Давно сложилось в науке так, что величины нормируются на диапазон от 0 до 1.

Для этого функция преобразования y=f(x) должна обладать следующими свойствами:

y(xmin)=0; y(xmax)=1; dy/dx>0   (1)

Любая функция с такими свойствами м.б. использована для нормировки. Например, если xmax Infin, то можно выбрать функцию

Expmax

Легко видеть, что за счёт выбора соответствующей функции можно учесть разнообразные эффекты искажения оценок. Например, склонность респондента к крайним оценкам. При этом, возможно, следует применять для различных респондентов и различные функции преобразования, учитывающие особенности их личности, статуса и т.п. Примерные графики таких функций — на рис. 1.

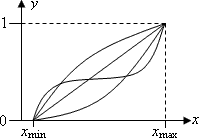


Рис. 1.1. **Графики функции нормировки**

Наиболее часто применяется линейное преобразование:

Lin1   (2)

Если полагать, что увеличение х описывает как возрастание выраженности качества А, так и убывание степени некоторого другого качества В, то нормированной мерой качества В может служить просто разность y´=1–y. Таковы, например, родственные по смыслу качества ‘близость’ и ‘дистанция’. Их метризация выявляет плохо осознаваемую ранее, но вполне четкую дополнительность и даже противоположность.

### 1.4.4. Нормировка биполярного показателя

Обычно такой показатель представляет собой ‘склейку’ двух взаимопредполагающих и антонимичных униполярных качеств А и В.

Часто В есть просто отрицание А и наоборот. По такому принципу построены, например, шкалы семантического дифференциала. Однако, пары для такого дифференциала следует проверить по словарю антонимов (например, два антонима к слову «веселый» – «грустный» и «мрачный» – вовсе не являются синонимами).

Нормировка соответствующей величины предполагает выбор «положительного» направления оси y. В качестве такового **произвольно** выбирается тот из полюсов шкалы, увеличение интенсивности которого **принимается как возрастание** y. Противоположный полюс автоматически становится «отрицательным». Подчеркнем, что никакой модальности (аксиологической оценки) за этим нет — играть роль могут только сложившиеся смысловые стереотипы, но не более того.

Пусть величина х оценивает степень выраженности обоих качеств (с соответствующим обозначением, например, ‘очень люблю’ или ‘слегка ненавижу’). Нормировку можно проводить при помощи любой функции, удовлетворяющей условиям (1). В частности, это м.б. и линейное преобразование:

Lin2(3)

Очевидно, что yPrin[–1; +1].

Обе формулы (2) и (3) описывают линейное преобразование вида y=k·x+b. Поэтому все **статистические выводы** относительно величин x и y **полностью совпадают**.

### 1.4.5. Особенности балльных шкал

При использовании балльной шкалы имеется несколько тонкостей, которые часто упускаются из виду:

* Иногда нет ответов на все вопросы, относящиеся к данному показателю. Причины разные — ответ просто не дан, ошибка при внесении ответа или его кодировке, … Короче — имеются пропуски ответов.
* Практически всегда балл приравнивается к номеру ответа среди прочих. И наименьший балл становится равным 1.
* Хотелось бы использовать для некоторых вопросов ответ с числом градаций, отличающимся от остальных. Но тогда его вклад надо учитывать как-то по-другому.

При нормировке балльной шкалы надо всего лишь принять, что х = S, где S сумма набранных баллов по полученным ответам (а не заданных вопросов!). Соответственно, Smin и Smax — минимальная и максимальная суммы баллов, которые можно набрать при полученных ответах.

Если же градации для всех вопросов одинаковы, то число N — это как раз и есть число таких ответов, за которые начислены баллы. Тогда формула (2) примет простой вид:

Lin1b(4)

Здесь bmin и bmax — наименьшее и наибольшее значения баллов. При этом у меняется в диапазоне от 0 до 1. Границы ‘0’ он достигает при всех ответах, равных bmin, а ‘1’ — равных bmax.

Для нормировки балльного показателя на дипазон [-1; 1] надо пользоваться формулой:

Lin2b(5)

## Аналитические пакеты прикладных программ

Учитывая большую потребность в статистической данных, разработано множество аналитических пакетов прикладных программ, таких, как SPSS, Statistica, SAS, STATA и другие. Как раньше – на счетах, логарифмической линейке или калькуляторах сейчас уже никто не считает. Зато гораздо больше внимания уделяется постановке задач, сбору и оценке качества информации, построению моделей и интерпретации результатов статистических расчетов.

**1) SPSS Statistics** (аббревиатура [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *«Statistical Package for the Social Sciences»* — «статистический пакет для социальных наук») — [компьютерная программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) для [статистической обработки](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7) данных, один из лидеров рынка в области коммерческих статистических продуктов, предназначенных для проведения прикладных исследований в [социальных науках](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%BD%D0%B0%D1%83%D0%BA%D0%B8).

Между 2009 и 2010 название программного обеспечения SPSS было изменено на PASW (Predictive Analytics SoftWare) Statistics.

28 июля 2009 компания объявила, что она была приобретена компанией [IBM](https://ru.wikipedia.org/wiki/IBM) за 1,2 млрд долл. США. По состоянию на январь 2010 года компания стала называться «SPSS: An IBM Company».

По мнению некоторых авторов, SPSS «занимает ведущее положение среди программ, предназначенных для статистической обработки информации».

**История**

[Норман Най](https://ru.wikipedia.org/w/index.php?title=%D0%9D%D0%B0%D0%B9,_%D0%9D%D0%BE%D1%80%D0%BC%D0%B0%D0%BD&action=edit&redlink=1), [Хедли Халл](https://ru.wikipedia.org/w/index.php?title=%D0%A5%D0%B0%D0%BB%D0%BB,_%D0%A5%D0%B5%D0%B4%D0%BB%D0%B8&action=edit&redlink=1) и [Дейл Бент](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%B5%D0%BD%D1%82,_%D0%94%D0%B5%D0%B9%D0%BB&action=edit&redlink=1) разработали первую версию системы в [1968 году](https://ru.wikipedia.org/wiki/1968_%D0%B3%D0%BE%D0%B4), затем этот пакет развивался в рамках [Чикагского университета](https://ru.wikipedia.org/wiki/%D0%A3%D0%BD%D0%B8%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%82%D0%B5%D1%82_%D0%A7%D0%B8%D0%BA%D0%B0%D0%B3%D0%BE). Первое пользовательское руководство вышло в [1970 году](https://ru.wikipedia.org/wiki/1970_%D0%B3%D0%BE%D0%B4) в издательстве [McGraw-Hill](https://ru.wikipedia.org/wiki/McGraw-Hill), а с [1975 года](https://ru.wikipedia.org/wiki/1975_%D0%B3%D0%BE%D0%B4) проект выделился в отдельную компанию *SPSS Inc.* Первая версия пакета под [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows) вышла в [1992 году](https://ru.wikipedia.org/wiki/1992_%D0%B3%D0%BE%D0%B4). На данный момент также существуют версии под Mac OS X и Linux.

В [2009 году](https://ru.wikipedia.org/wiki/2009_%D0%B3%D0%BE%D0%B4) компания SPSS произвела [ребрендинг](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B1%D1%80%D0%B5%D0%BD%D0%B4%D0%B8%D0%BD%D0%B3) своего статистического пакета, который теперь стал называться PASW Statistics (Predictive Analytics SoftWare). 29 июля 2009 года компания SPSS объявила о том, что она приобретается фирмой [IBM](https://ru.wikipedia.org/wiki/IBM).

**Возможности**

* Ввод и хранение данных.
* Возможность использования переменных разных типов.
* Частотность признаков, таблицы, графики, таблицы сопряжённости, диаграммы.
* Первичная описательная статистика.
* Маркетинговые исследования
* Анализ данных маркетинговых исследований

**Модули**

* IBM SPSS Statistics Base
* IBM SPSS Advanced Statistics
* IBM SPSS Bootstrapping
* IBM SPSS Categories
* IBM SPSS Complex Samples
* IBM SPSS Conjoint
* IBM SPSS Custom Tables
* IBM SPSS Data Preparation
* IBM SPSS Decision Trees
* IBM SPSS Direct Marketing
* IBM SPSS Exact Tests
* IBM SPSS Forecasting
* IBM SPSS Missing Values
* IBM SPSS Neural Networks
* IBM SPSS Regression

**2) Statistica** — [программный пакет](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BA%D0%B5%D1%82_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D1%8B%D1%85_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC) для статистического анализа, разработанный компанией StatSoft, реализующий функции [анализа данных](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), [управления данных](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8), [добычи данных](https://ru.wikipedia.org/wiki/Data_mining), [визуализации данных](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) с привлечением [статистических методов](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0).

**Состав**

Существуют различные варианты пакета в зависимости от целей и задач пользователя:

1. однопользовательская версия (*single-user*);
2. сетевая версия (*concurrent network*) — для использования в [локальных вычислительных сетях](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C);
3. enterprise-версия — для использования в крупных организациях;
4. веб-версия — для использования в крупных сетях через [веб-браузер](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80).

Также существуют различные комплекты поставки в зависимости от включённых функций:

* *Base* — набор основных [статистик](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA) и методов для разведочного анализа.
* *Advanced* — включает все возможности продукта *Base*, а также модули углубленных линейных и нелинейных моделей, многомерных технологий анализа данных, анализа мощности и интервального оценивания.
* *Quality Control* (контроль качества) — включает методы [управления качеством данных](https://ru.wikipedia.org/w/index.php?title=%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BA%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE%D0%BC_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85&action=edit&redlink=1), а также [контрольные карты](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%BA%D0%B0%D1%80%D1%82%D1%8B) презентационного качества.
* *Automated Neural Networks* — включает методы для [нейросетевых исследований](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0).
* *Data Miner* — включает методы [добычи данных](https://ru.wikipedia.org/wiki/Data_mining).
* *Text Miner* — дополнительная возможность для добычи данных над текстами.
* *Process Optimization* — возможности проводить мониторинг процессов, идентифицировать и предотвращать проблемы, относящиеся к контролю качества на производстве.
* *Monitoring and Alerting Server (MAS)* (сервер мониторинга и предупреждений) — средства для централизованного автоматизированного мониторинга различных процессов и параметров продуктов.

**Графика**

Пакет обладает широкими графическими возможностями, позволяет выводить информацию в виде различных типов графиков (включая научные, деловые, трёхмерные и двухмерные графики в различных системах координат, специализированные статистические графики — гистограммы, матричные, категорированные графики и др.), все компоненты графиков настраиваются.

# Глава 2. Выбор и обоснование алгоритма

## 2.1. Содержательная постановка

Как было сказано ранее, основная цель работы – разработка системы оценки социально-экономических показателей СКФО с использованием различных статистических методов. Фактически, данная задача сводится к определению рейтинга регионов СКФО. Все методики определения рейтингов основаны на выборе оценочных показателей, которые, впоследствии, обрабатываются некоторым образом. Данные критерии могут быть разнородны: для одних – наилучшим значением является max, для других – min.

Такими критериями, в моей работе, служат следующие:

Наилучшим значением является min:

• Число детей, умерших в возрасте до 1 года, на1000 родившихся живыми;

• Уровень безработицы;

• Заболеваемость на 1000 человек населения по основным классам болезней;

• Число зарегистрированных преступлений на 100 000 человек населения;

• Число зарегистрированных убийств и покушений на убийство;

• Структура потребительских расходов домашних хозяйств;

Наилучшим значением является max:

• Ожидаемая продолжительность жизни при рождении;

• Коэффициенты миграционного прироста на 10 000 человек населения;

• Потребление мяса и мясопродуктов на душу населения;

• Валовой региональный продукт;

• Индексы промышленного производства;

• Индексы производства продукции сельского хозяйства.

Для упрощения работы все критерии первой группы преобразуются к формату второй группы – наилучшим значением является max.

Все данные взяты с сайта Федеральной службы государственной статистики **[[1]](#footnote-2)** за период с 1990 по 2015гг.

Существуют различные методики определения рейтинга:

• *простые методы:* метод Кондорсе (наилучшим является объект, которые лучше других по правилу большинства рейтинговых показателей), метод Борда (наилучшим считается объект, с максимальной суммой баллов, полученных в результате сортировки объектов экспертамипо каждому критерию), метод Копленда (рейтинг определяется по числу объектов, которых он лучше и хуже соответственно, по мнению экспертов), метод Симпсона (соотносит мнения экспертов и показателей, указывающих , что один объект лучше другого; все объекты упорядочиваются в порядке убывания полученных оценок);

•*метод линейного ранжирования:* состоит из четырёх этапов; сводит все критерии в один посредством некоторой функции свертки нормированных показателей с последующим упорядочиванием; всем показателям назначают весовые характеристики в зависимости от их степени важности по мнению экспертов;

• *кластер-методики*: основаны на разбиении всех объектов на группы однотипных; окончательные оценки группам дают эксперты.

На мой взгляд, общий недостаток приведённых методов состоит в необходимости привлечения экспертов для оценки объектов. Мнения экспертов часто противоречивы, вследствие чего полученные решения могут быть неоднозначными. Альтернативой данного подходу является использование метода эталонов, который позволяет учитывать различные показатели (в том числе и противоречивые), при минимальном участии экспертов. Метод основан на выделении объектов эталонов (возможно гипотетических) – абсолютно лучшего, либо абсолютно худшего, которые будут иметь, соответственно, наилучшие и наихудшие значения.

От правильности выбора этих объектов зависит однозначность и адекватность получаемых решений. Объекты-эталоны должны быть определены на том же множестве параметров, что и исходныеобъекты.

Эталонные значения параметров выбираются как граничные, в имеющейся выборке данных (maxили minв зависимости от типа параметра). Формальная постановка задачи представлена ниже.

## 2.2. Формальная постановка задачи

**Обозначения:**

**n –** кол-во критериев;

**m–** кол-во регионов;

**–** исходные значения;

**–** нормированные значения j-ого критерия у *i*-ого региона;

**B–** худший эталон, все характеристики которого нули (0).

## 2.3. Метод эталонов

Применение метода эталонов с одной стороны позволяет получать Парето-оптимальные решения, с другой стороны – не требует экспертной оценки исходной информации. Суть метода состоит в выделении эталонного решения: наилучшего или наихудшего. Наилучший эталон будет иметь в качестве своих характеристик наилучшие значения целевых критериев задачи (1), а наихудший, соответственно, наихудшие значения.

Графическое представление:

Рис.2.1

* Два способа преобразования исходной многокритериальной задачи в однокритериальную:
  + с учетом близости с наилучшим эталоном
  + с учетом удаленности от наихудшего эталона.

Рис. 2.1

## 2.4. Алгоритм

Шаг 1. Загрузка исходных данных.

Шаг 2. Выбор критериев для оценки.

Шаг3.Обработка данных.

Шаг 3.1. = для всех , при которых

min.

Шаг 3.2. Нормирование данных

Шаг 3.3. Приведение показателей к виду

Шаг 4. Вычисление расстояний между регионами и лучшим эталоном.

Шаг 5. Ранжирование регионов по степени удалённости от лучшего эталона.

Шаг 6. Вывод результатов.

## 2.5. Пример решения

**3 региона:**

1) Северная Осетия

2) Ставрополье

3) Дагестан

**3 критерия:**

1) Число зарегистрированных убийств и покушений на убийство

2) Валовой региональный продукт

3) Коэффициенты миграционного прироста на 10 000 человек населения

**Промежуток времени:**

2005-2010 года

**Таблицы:**

**Северная Осетия:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **2005** | **2006** | **2007** | **2008** | **2009** | **2010** |
| Число зарегистрированных убийств | 96 | 112 | 72 | 78 | 70 | 50 |
| Валовой региональный продукт | 31182,2 | 43341,2 | 52804,8 | 57867,6 | 65140,7 | 75327,4 |
| Коэффициенты миграционного прироста | 18 | 24 | 14 | -26 | -28 | -25 |

**Ставрополье:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **2005** | **2006** | **2007** | **2008** | **2009** | **2010** |
| Число зарегистрированных убийств | 325 | 351 | 280 | 238 | 211 | 221 |
| Валовой региональный продукт | 146569,3 | 181675,1 | 222239,6 | 275084,2 | 277466,8 | 330790,8 |
| Коэффициенты миграционного прироста | 64 | 41 | 66 | 40 | 47 | 34 |

**Дагестан:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **2005** | **2006** | | **2007** | | **2008** | **2009** | **2010** |
| Число зарегистрированных убийств | 200 | 243 | | 170 | | 172 | 204 | 296 |
| Валовой региональный продукт | 90442,6 | 124153,5 | 156928,8 | | 211260,2 | | 265092,1 | 274354,2 |
| Коэффициенты миграционного прироста | 55 | 68 | 82 | | 15 | | 30 | 36 |

**Шаг 1:**

Находим значения, которые меньше нуля (0). В нашем случае это значения из критерия «Коэффициенты миграционного прироста».

Min = -28 (Северная Осетия [3;5])

= + 28 + 1

Северная Осетия: 47, 53, 43, 3, 1, 4

Ставрополье: 93, 70, 95, 69, 76, 63

Дагестан: 84, 97, 111, 44, 59, 65

**Шаг 2:**

Нормируем данные:

Min/Max

1) min = 0, max = 351

2) min = 0, max = 330790,8

3) min = 0, max = 111

Таблицы нормирования:

**Северная Осетия:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **2005** | **2006** | **2007** | **2008** | **2009** | **2010** |
| Число зарегистрированных убийств | 0,273 | 0,319 | 0,205 | 0,222 | 0,199 | 0,142 |
| Валовой региональный продукт | 0,094 | 0,131 | 0,159 | 0,174 | 0,196 | 0,227 |
| Коэффициенты миграционного прироста | 0,423 | 0,477 | 0,387 | 0,027 | 0,009 | 0,036 |

**Ставрополье:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **2005** | **2006** | **2007** | **2008** | **2009** | **2010** |
| Число зарегистрированных убийств | 0,925 | 1 | 0,797 | 0,678 | 0,601 | 0,629 |
| Валовой региональный продукт | 0,443 | 0,549 | 0,671 | 0,831 | 0,838 | 1 |
| Коэффициенты миграционного прироста | 0,837 | 0,630 | 0,855 | 0,621 | 0,684 | 0,567 |

**Дагестан:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **2005** | **2006** | **2007** | **2008** | **2009** | **2010** |
| Число зарегистрированных убийств | 0,569 | 0,692 | 0,484 | 0,490 | 0,581 | 0,843 |
| Валовой региональный продукт | 0,273 | 0,375 | 0,474 | 0,638 | 0,801 | 0,829 |
| Коэффициенты миграционного прироста | 0,756 | 0,873 | 1 | 0,396 | 0,531 | 0,585 |

**Шаг3:**

В нашем случае критерий «Число зарегистрированных убийств» имеет эталон ноль (0). Нам же надо привести их к одному типу, что бы эталоном был один (1). Для этого используем формулу , что бы эквивалентно изменить значение с точностью наоборот и тогда этот критерий будет стремиться к max, а именно к единице (1).

Северная Осетия:

1-0,273 = 0,727; 1 – 0,319 = 0,681; 1 – 0,205 = 0,795; 1 – 0,222 = 0,778; 1 – 199 = 0,801; 1-142 = 0,858;

Ставрополье:

1-0,925 = 0,075; 1 – 1 = 0; 1 – 0,797 = 0,203; 1 – 0,678 = 0,322; 1 – 0,601 = 0,399; 1 - 629 = 0,371;

Дагестан:

1-0,569 = 0,431; 1 – 0,692 = 0,308; 1 – 0, 484 = 0,516; 1 – 0,490 = 0,51; 1 – 581 = 0,419; 1 - 843 = 0,157;

**Шаг 4:**

Находим расстояние до худшего эталона за все года с 2005 по 2010гг по формуле -

**2005 год:**

Северная Осетия:

**= = 0,846;**

Ставрополье:  **= = 0,950;**

Дагестан:  **= = 0,912;**

**2006 год:**

Северная Осетия:

**= = 0,842;**

Ставрополье:  **= = 0,836;**

Дагестан:  **= = 0,999;**

**2007 год:**

Северная Осетия:  **= = 0,898;**

Ставрополье:  **= = 1,106;**

Дагестан:  **= = 1,221;**

**2008 год:**

Северная Осетия:

**= = 0,798;**

Ставрополье:  **= = 1,086;**

Дагестан:  **= = 0,908;**

**2009 год:**

Северная Осетия:

**= = 0,825;**

Ставрополье:  **= = 1,153;**

Дагестан:  **= = 1,048;**

**2010 год:**

Северная Осетия:

**= = 0,888;**

Ставрополье:  **= = 1,208;**

Дагестан:  **= 1,027;**

Таблица:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **2005** | **2006** | **2007** | **2008** | **2009** | **2010** |
| Северная Осетия | 0,846 | 0,842 | 0,898 | 0,798 | 0,825 | 0,888 |
| Ставрополье | 0,950 | 0,836 | 1,106 | 1,086 | 1,153 | 1,208 |
| Дагестан | 0,912 | 0,999 | 1,221 | 0,908 | 1,048 | 1,027 |

**Места:**

2005 – Ставрополье, Дагестан, Северная Осетия;

2006 – Дагестан, Северная Осетия, Ставрополье;

2007 – Дагестан, Ставрополье, Северная Осетия;

2008 – Ставрополье, Дагестан, Северная Осетия;

2009 – Ставрополье, Дагестан, Северная Осетия;

2010 – Ставрополье, Дагестан, Северная Осетия;

# Глава 3. Выбор и обоснование программной реализации

## 3.1. Язык программирования C#

**C#** — [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Разработан в [1998](https://ru.wikipedia.org/wiki/1998)—[2001 годах](https://ru.wikipedia.org/wiki/2001_%D0%B3%D0%BE%D0%B4) группой инженеров под руководством [Андерса Хейлсберга](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%B9%D0%BB%D1%81%D0%B1%D0%B5%D1%80%D0%B3,_%D0%90%D0%BD%D0%B4%D0%B5%D1%80%D1%81) в компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) как язык разработки приложений для платформы [Microsoft .NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) и впоследствии был стандартизирован как [ECMA](https://ru.wikipedia.org/wiki/ECMA)-334 и [ISO](https://ru.wikipedia.org/wiki/ISO)/[IEC](https://ru.wikipedia.org/wiki/IEC) 23270.

C# относится к семье языков с [C-подобным синтаксисом](https://ru.wikipedia.org/wiki/C-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81), из них его синтаксис наиболее близок к [C++](https://ru.wikipedia.org/wiki/C%2B%2B) и [Java](https://ru.wikipedia.org/wiki/Java). Язык имеет [статическую типизацию](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), поддерживает [полиморфизм](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [перегрузку операторов](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BE%D0%B2) (в том числе операторов явного и неявного приведения типа), [делегаты](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%B3%D0%B0%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), атрибуты, [события](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [свойства](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [обобщённые](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) типы и методы, [итераторы](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#.D0.93.D0.B5.D0.BD.D0.B5.D1.80.D0.B0.D1.82.D0.BE.D1.80.D1.8B), [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) с поддержкой [замыканий](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [LINQ](https://ru.wikipedia.org/wiki/Language_Integrated_Query), [исключения](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9), [комментарии](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B8_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) в формате [XML](https://ru.wikipedia.org/wiki/XML).

Переняв многое  от своих предшественников, языков [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [Pascal](https://ru.wikipedia.org/wiki/Pascal_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), [Модула](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B0-2), [Smalltalk](https://ru.wikipedia.org/wiki/Smalltalk) и, в особенности, [Java](https://ru.wikipedia.org/wiki/Java_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает [множественное наследование](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BD%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) классов (между тем допускается [множественное наследование интерфейсов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#.D0.9C.D0.BD.D0.BE.D0.B6.D0.B5.D1.81.D1.82.D0.B2.D0.B5.D0.BD.D0.BD.D0.BE.D0.B5_.D0.BD.D0.B0.D1.81.D0.BB.D0.B5.D0.B4.D0.BE.D0.B2.D0.B0.D0.BD.D0.B8.D0.B5_.D0.B8_.D1.80.D0.B5.D0.B0.D0.BB.D0.B8.D0.B7.D0.B0.D1.86.D0.B8.D1.8F_.D0.B8.D0.BD.D1.82.D0.B5.D1.80.D1.8)).

C# разрабатывался как язык программирования прикладного уровня для [CLR](https://ru.wikipedia.org/wiki/Common_Language_Runtime) и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает [BCL](https://ru.wikipedia.org/wiki/Base_Class_Library). Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим [.NET](https://ru.wikipedia.org/wiki/.NET_Framework)-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, [Сборка мусора](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на [VB.NET](https://ru.wikipedia.org/wiki/Visual_Basic_.NET), [J#](https://ru.wikipedia.org/wiki/Visual_J%E2%99%AF) и др.

Я выбрал C#, так как он предназначен для разработки самых разнообразных приложений, предназначенных для выполнения в среде .NET Framework. Язык C# прост, типобезопасен и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C.Поддержка Visual C# в Visual Studio обеспечивается с помощью полнофункционального редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, мощного и удобного отладчика и многих других средств. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и другим полезным, правильным классам, что существенно ускоряет цикл разработки.

## 3.2. Интерфейс программы

**Алгоритм** включает следующие шаги:

1 шаг: Обработка данных;

2 шаг: Нормирование полученных значений;

3 шаг: Ранжирование регионов;

4 шаг: Построение графиков.

5 шаг: Так же есть функция наблюдения за динамикой изменения расстояния между двумя регионами со временем.

Данные по регионам хранятся в Excel-файле («123.xlsx») для удобства и произвольного, быстрого, а так же не сложного доступа к данным для их собственного редактирования, либо добавления. Так же данный способ занимает гораздо меньше памяти, имея вес 32,5 кб.

При запуске программы открывается главное окно (Рис. 1). В начале работы почти все компоненты будут недоступны. В верхней левой части будет доступна кнопка «Открыть файл»

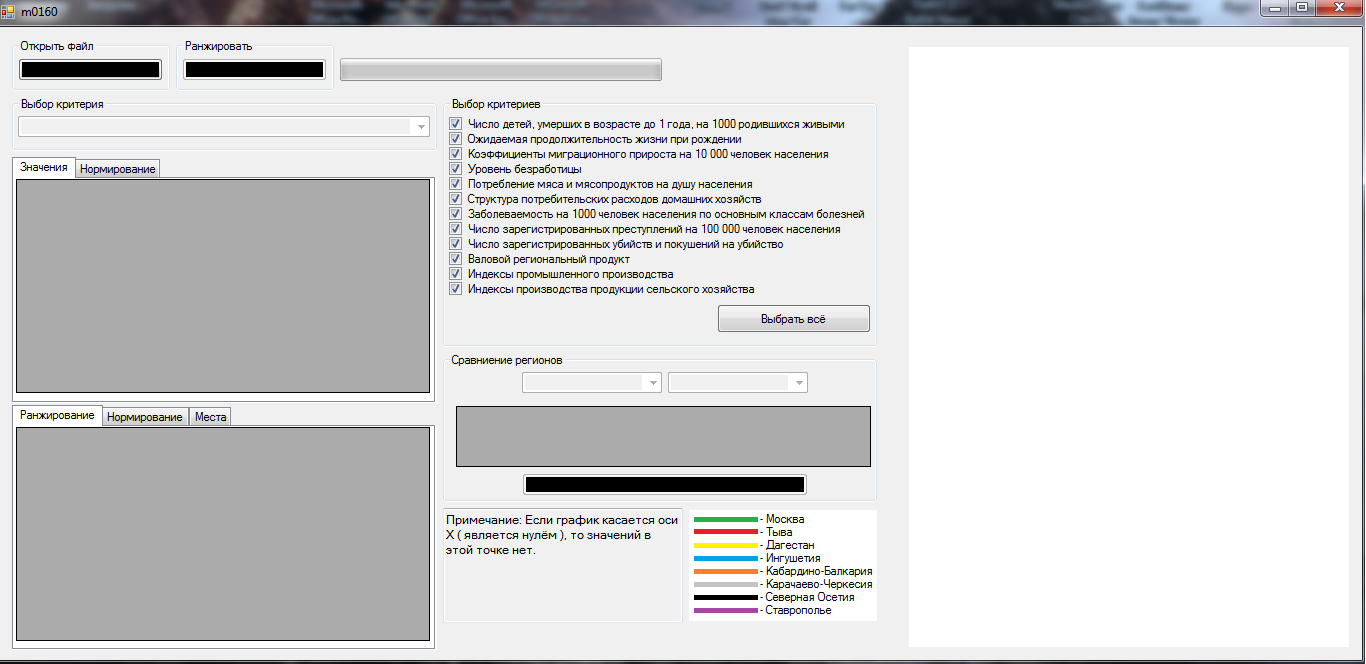


Рис. 3.1

При открытии файла пользователю будут доступны для открытия только файлы формата .xlsx (Excel), выбираем его и загружаем в программу (Рис. 2). Начнётся загрузка из файла, при этом будет заполняться полоса загрузки для визуального изображения текущего состояния. В конце программа уведомит пользователя об окончании загрузки.

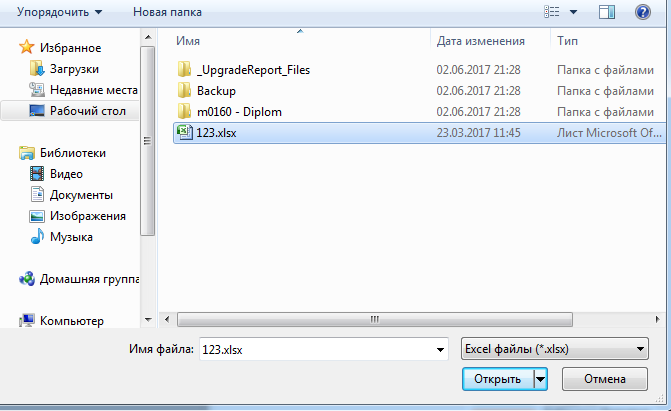


Рис. 3.2

После загрузки пользователь сможет выбирать критерии (Рис. 3), после чего заполнятся таблицы «Значения» (Рис. 4) , «Нормирование» (Рис. 5) и построится график (Рис. 6)

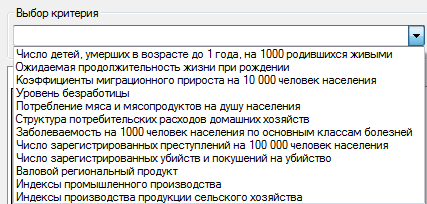


Рис. 3.3



Рис. 3.4



Рис. 3.5

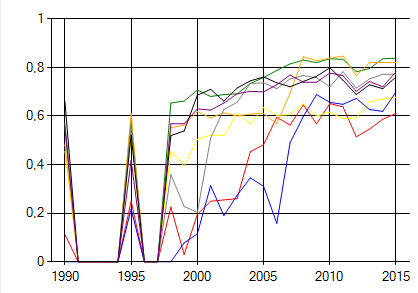


Рис. 3.6

Так же будет доступна кнопка «Ранжировать» (Рис. 7). При её нажатии регионы просуммируются по выбранным критериям (рис. 8) и заполнятся таблицы «Ранжирование» (Рис. 9), «Нормирование» (Рис. 10), «Места» (Рис. 11) и построится график (Рис. 12).

7.png

Рис. 3.7

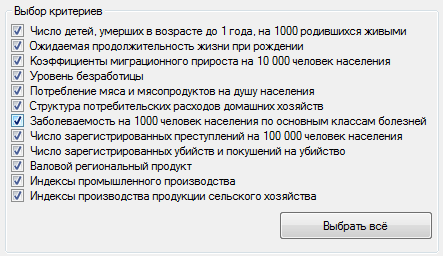


Рис.3.8



Рис.3.9

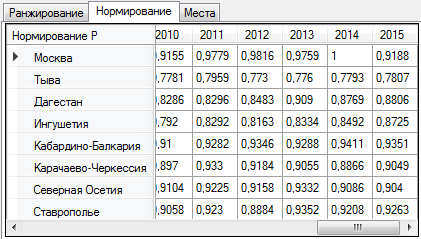


Рис.3.10



Рис.3.11

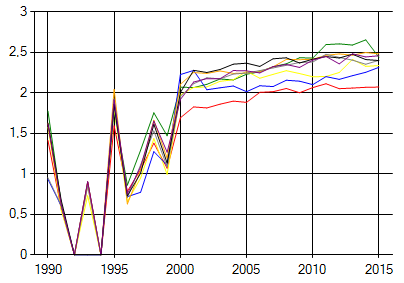


Рис.3.12

При двойном нажатии на один из графиков, он откроется в новом окне в увеличенном размере (Рис. 13). Пока не закроется это окно, главное окно будет недоступно.

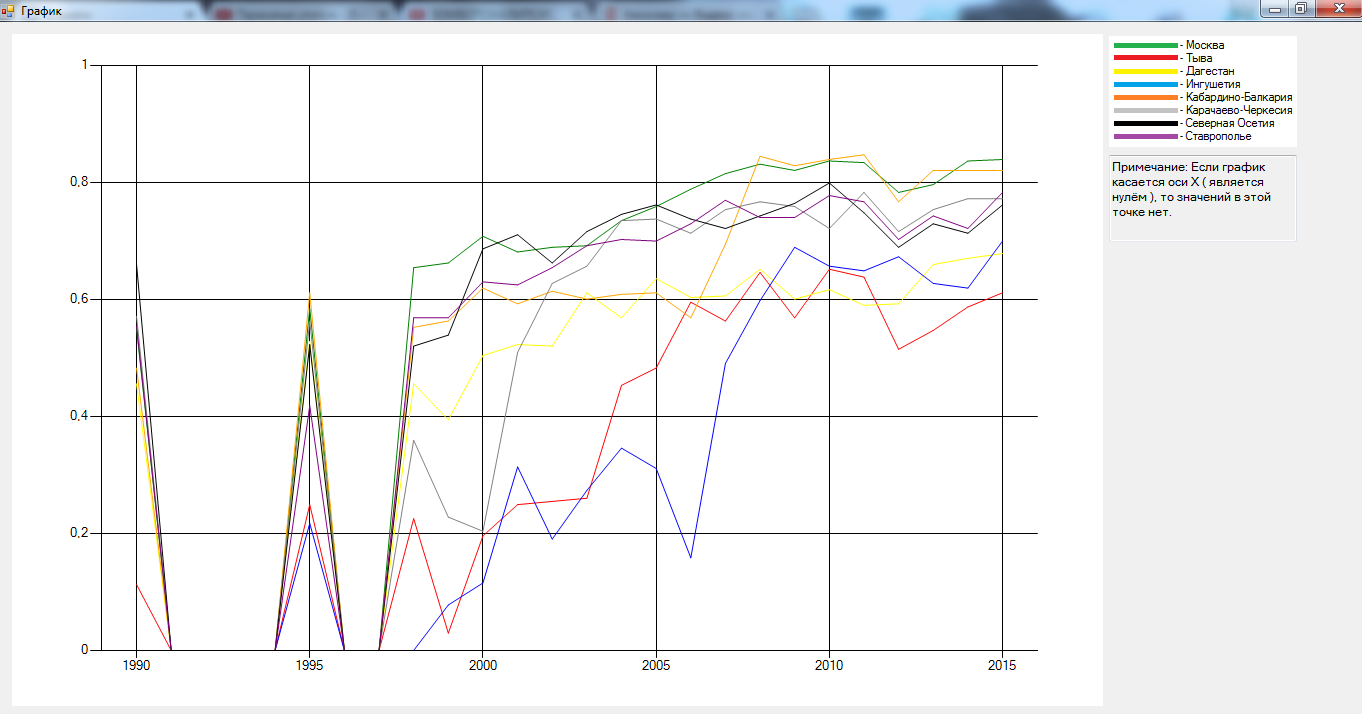


Рис. 3.13

В окне есть дополнительная функция для определения расстояния между двумя регионами с годами по выбранным критериям (Рис. 8), для просмотра динамики изменения расстояния со временем.

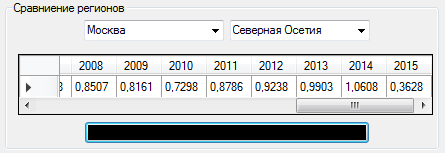


Рис. 3.14

Так же присутствует изображение с подсказкой для определения региона на графике (Рис. 15).

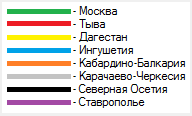


Рис. 3.15

# Глава 4. Экспериментальная часть

**Целью эксперимента является следующее:**

* анализ зависимости времени счета от количества регионов и количества критериев.

Эксперимент был поставлен на компьютере со следующими характеристиками:

* + ПроцессорIntel (R) Celeron (R) 1.50 GHz 1.50 GHz;
  + 8ГБ ОЗУ;
  + Microsoft Windows 7.

**Эксперимент №1**

Целью эксперимента является выявление зависимости времени счёта от количества регионов при фиксированном числе критериев – 12.

Таблица 1. Исходные данные для эксперимента №1

|  |  |
| --- | --- |
| **Кол-во регионов** | **Время выполнения** |
| 8 | 4 |
| 10 | 6 |
| 13 | 7 |
| 15 | 9 |
| 25 | 15 |
| 35 | 21 |
| 50 | 30 |
| 60 | 35 |
| 70 | 41 |
| 80 | 47 |
| 90 | 52 |
| 100 | 59 |

Рис. 4.1. Зависимость времени от числа регионов

**Эксперимент №2**

Целью эксперимента является выявление зависимости времени счёта от количества критериев при фиксированном числе регионов – 8.

Таблица 2. Исходные данные для эксперимента №2

|  |  |
| --- | --- |
| **Кол-во критерий** | **Время выполнения** |
| 12 | 4 |
| 15 | 6 |
| 18 | 7 |
| 25 | 10 |
| 30 | 11 |
| 40 | 14 |
| 50 | 18 |
| 60 | 22 |
| 70 | 25 |
| 80 | 28 |
| 90 | 33 |
| 100 | 36 |

Рис. 4.2. Зависимость времени от числа критериев

Таким образом, рассмотренные в Главе 4 эксперименты позволили визуально выявить зависимость количества регионов и количества критериев ко времени.

# Заключение

В результате выполнения выпускной квалификационной работы была полностью достигнута поставленная цель работы –разработка системы оценки социально-экономических показателей СКФО с использованием различных статистических методов. Все поставленные задачи были выполнены:

* проведен анализ существующих подходов к оценке социально-экономических показателей;
* выбрана математическая модель оценки и алгоритма для ее решения;
* осуществлена программная реализация системы на языке С#;
* проведены тестирование и отладка.

# Список использованной литературы

1. Глинский В.В., Ионин В.Г. Статистический анализ. Учебное пособие.- М.: ФИЛИНЪ, 1998 г.-264 с.

2. Ефимова М.Р., Петрова Е.В., Румянцев В.Н. Общая теория статистики. Учебник.-М.: ИНФРА-М, 1996 г.-416 с.

3. Костина Л.В. Методика построения статистических графиков. Методическое пособие.- Казань, ТИСБИ, 2000 г.-49 с.

4. Ларичев О.И. Теория и методы принятия решений. М: Логос, 2002. 392 с.

5. Будаева А.А., Гроппен В.О. Принятие решений: теория, технология, приложения. Владикавказ: Фламинго, 2009. 184 с.

# Приложение. Листинг программы

usingSystem;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

using Excel = Microsoft.Office.Interop.Excel;

namespace m0160\_\_\_Diplom

{

public partial class Form1 : Form

{

System.Windows.Forms.DataVisualization.Charting.Chart chart1;

System.Windows.Forms.DataVisualization.Charting.Chart chart2;

Microsoft.Office.Interop.Excel.Application ObjExcel;

Microsoft.Office.Interop.Excel.Workbook ObjWorkBook;

Microsoft.Office.Interop.Excel.Worksheet ObjWorkSheet;

double[,,]table;

double[,,]normirovenie;

int ChisloStolbcov = 0;

bool tt = true;

public Form1()

{

InitializeComponent();

}

private void comboBox1\_TextChanged(object sender, EventArgs e)

{

#region Выбор критерия

int Bukva = 0;

if (comboBox1.Text == "Число детей, умерших в возрасте до 1 года, на 1000 родившихся живыми") { Bukva = 0; }

if (comboBox1.Text == "Ожидаемая продолжительность жизни при рождении") { Bukva = 1; }

if (comboBox1.Text == "Коэффициенты миграционного прироста на 10 000 человек населения") { Bukva = 2; }

if (comboBox1.Text == "Уровень безработицы") { Bukva = 3; }

if (comboBox1.Text == "Потребление мяса и мясопродуктов на душу населения") { Bukva = 4; }

if (comboBox1.Text == "Структура потребительских расходов домашних хозяйств") { Bukva = 5; }

if (comboBox1.Text == "Заболеваемость на 1000 человек населения по основным классам болезней") { Bukva = 6; }

if (comboBox1.Text == "Число зарегистрированных преступлений на 100 000 человек населения") { Bukva = 7; }

if (comboBox1.Text == "Число зарегистрированных убийств и покушений на убийство") { Bukva = 8; }

if (comboBox1.Text == "Валовой региональный продукт") { Bukva = 9; }

if (comboBox1.Text == "Индексы промышленного производства") { Bukva = 10; }

if (comboBox1.Text == "Индексы производства продукции сельского хозяйства") { Bukva = 11; }

#endregion

#region Заполнениетаблицы

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < ChisloStolbcov; j++)

{

dataGridView1.Rows[i].Cells[j].Value = table[Bukva, i, j];

Application.DoEvents();

}

}

#endregion

#region Заполнениетаблицынормирования

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < ChisloStolbcov; j++)

{

dataGridView2.Rows[i].Cells[j].Value = normirovenie[Bukva, i, j];

Application.DoEvents();

}

}

#endregion

#region Описаниелинийграфика

var ser1 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser1",

Color = System.Drawing.Color.Green,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser2 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser2",

Color = System.Drawing.Color.Red,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser3 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser3",

Color = System.Drawing.Color.Yellow,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser4 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser4",

Color = System.Drawing.Color.Blue,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser5 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser5",

Color = System.Drawing.Color.Orange,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser6 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser6",

Color = System.Drawing.Color.Gray,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser7 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser7",

Color = System.Drawing.Color.Black,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser8 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser8",

Color = System.Drawing.Color.Purple,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

#endregion

#region Построениеграфика

System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea8 = new System.Windows.Forms.DataVisualization.Charting.ChartArea();

int God = 1990;

this.chart2.ChartAreas.Clear();

this.chart2.ChartAreas.Add(chartArea8);

chart2.Series.Clear();

this.chart2.Series.Add(ser1);

this.chart2.Series.Add(ser2);

this.chart2.Series.Add(ser3);

this.chart2.Series.Add(ser4);

this.chart2.Series.Add(ser5);

this.chart2.Series.Add(ser6);

this.chart2.Series.Add(ser7);

this.chart2.Series.Add(ser8);

for (int i = 0; i < ChisloStolbcov; i++)

{

ser1.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[0].Cells[i].Value));

ser2.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[1].Cells[i].Value));

ser3.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[2].Cells[i].Value));

ser4.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[3].Cells[i].Value));

ser5.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[4].Cells[i].Value));

ser6.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[5].Cells[i].Value));

ser7.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[6].Cells[i].Value));

ser8.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[7].Cells[i].Value));

God = God + 1;

}

#endregion

}

private void button1\_Click(object sender, EventArgs e)

{

openFileDialog1.Filter = "Excel файлы|\*.xlsx";

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

ObjExcel = new Microsoft.Office.Interop.Excel.Application();

ObjWorkBook = ObjExcel.Workbooks.Open(openFileDialog1.FileName, 0, false, 5, "", "", false, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "", true, false, 0, true, false, false);

}

else { goto uhodim; }

timer1.Enabled = true;

progressBar1.Visible = true;

int God = 1990;

ObjWorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)ObjWorkBook.Sheets[1];

for (int i = 1; i < 255; i++)

{ Microsoft.Office.Interop.Excel.Range range = ObjWorkSheet.get\_Range("A" + i.ToString(), "A" + i.ToString());

if (range.Text.ToString() == "") { break; }

ChisloStolbcov += 1;

}

ChisloStolbcov -= 1;

#region Созданиетаблицыдлязначений

dataGridView1.RowCount = 8;

dataGridView1.ColumnCount = ChisloStolbcov;

for (int i = 0; i < ChisloStolbcov; i++) { dataGridView1.Columns[i].Width = 45; }

dataGridView1.Rows[0].HeaderCell.Value = "Москва";

dataGridView1.Rows[1].HeaderCell.Value = "Тыва";

dataGridView1.Rows[2].HeaderCell.Value = "Дагестан";

dataGridView1.Rows[3].HeaderCell.Value = "Ингушетия";

dataGridView1.Rows[4].HeaderCell.Value = "Кабардино-Балкария";

dataGridView1.Rows[5].HeaderCell.Value = "Карачаево-Черкессия";

dataGridView1.Rows[6].HeaderCell.Value = "СевернаяОсетия";

dataGridView1.Rows[7].HeaderCell.Value = "Ставрополье";

dataGridView1.RowHeadersWidth = 150;

for (int i = 0; i < ChisloStolbcov; i++)

{

dataGridView1.Columns[i].HeaderCell.Value = God.ToString();

God += 1;

}

#endregion

progressBar1.Value = 15;

#region Созданиетаблицыдлянормирования

dataGridView2.RowCount = 8;

dataGridView2.ColumnCount = ChisloStolbcov;

for (int i = 0; i < ChisloStolbcov; i++) { dataGridView2.Columns[i].Width = 45; }

dataGridView2.TopLeftHeaderCell.Value = "Нормирование";

dataGridView2.Rows[0].HeaderCell.Value = "Москва";

dataGridView2.Rows[1].HeaderCell.Value = "Тыва";

dataGridView2.Rows[2].HeaderCell.Value = "Дагестан";

dataGridView2.Rows[3].HeaderCell.Value = "Ингушетия";

dataGridView2.Rows[4].HeaderCell.Value = "Кабардино-Балкария";

dataGridView2.Rows[5].HeaderCell.Value = "Карачаево-Черкессия";

dataGridView2.Rows[6].HeaderCell.Value = "СевернаяОсетия";

dataGridView2.Rows[7].HeaderCell.Value = "Ставрополье";

dataGridView2.RowHeadersWidth = 150;

God = 1990;

for (int i = 0; i < ChisloStolbcov; i++)

{

dataGridView2.Columns[i].HeaderCell.Value = God.ToString();

God += 1;

}

#endregion

progressBar1.Value = 30;

#region Созданиетаблицыдляранжирования

dataGridView3.RowCount = 8;

dataGridView3.ColumnCount = ChisloStolbcov;

for (int i = 0; i < ChisloStolbcov; i++) { dataGridView3.Columns[i].Width = 45; }

dataGridView3.TopLeftHeaderCell.Value = "Ранжирование";

dataGridView3.Rows[0].HeaderCell.Value = "Москва";

dataGridView3.Rows[1].HeaderCell.Value = "Тыва";

dataGridView3.Rows[2].HeaderCell.Value = "Дагестан";

dataGridView3.Rows[3].HeaderCell.Value = "Ингушетия";

dataGridView3.Rows[4].HeaderCell.Value = "Кабардино-Балкария";

dataGridView3.Rows[5].HeaderCell.Value = "Карачаево-Черкессия";

dataGridView3.Rows[6].HeaderCell.Value = "СевернаяОсетия";

dataGridView3.Rows[7].HeaderCell.Value = "Ставрополье";

dataGridView3.RowHeadersWidth = 150;

God = 1990;

for (int i = 0; i < ChisloStolbcov; i++)

{

dataGridView3.Columns[i].HeaderCell.Value = God.ToString();

God += 1;

}

#endregion

#region Создание таблицы для нормирования ранжирования

dataGridView5.RowCount = 8;

dataGridView5.ColumnCount = ChisloStolbcov;

for (int i = 0; i < ChisloStolbcov; i++) { dataGridView5.Columns[i].Width = 45; }

dataGridView5.TopLeftHeaderCell.Value = "НормированиеР";

dataGridView5.Rows[0].HeaderCell.Value = "Москва";

dataGridView5.Rows[1].HeaderCell.Value = "Тыва";

dataGridView5.Rows[2].HeaderCell.Value = "Дагестан";

dataGridView5.Rows[3].HeaderCell.Value = "Ингушетия";

dataGridView5.Rows[4].HeaderCell.Value = "Кабардино-Балкария";

dataGridView5.Rows[5].HeaderCell.Value = "Карачаево-Черкессия";

dataGridView5.Rows[6].HeaderCell.Value = "СевернаяОсетия";

dataGridView5.Rows[7].HeaderCell.Value = "Ставрополье";

dataGridView5.RowHeadersWidth = 150;

God = 1990;

for (int i = 0; i < ChisloStolbcov; i++)

{

dataGridView5.Columns[i].HeaderCell.Value = God.ToString();

God += 1;

}

#endregion

#region Создание таблицы для мест ранжирования

dataGridView6.RowCount = 8;

dataGridView6.ColumnCount = ChisloStolbcov;

for (int i = 0; i < ChisloStolbcov; i++) { dataGridView6.Columns[i].Width = 45; }

dataGridView6.TopLeftHeaderCell.Value = "Места";

dataGridView6.Rows[0].HeaderCell.Value = "Москва";

dataGridView6.Rows[1].HeaderCell.Value = "Тыва";

dataGridView6.Rows[2].HeaderCell.Value = "Дагестан";

dataGridView6.Rows[3].HeaderCell.Value = "Ингушетия";

dataGridView6.Rows[4].HeaderCell.Value = "Кабардино-Балкария";

dataGridView6.Rows[5].HeaderCell.Value = "Карачаево-Черкессия";

dataGridView6.Rows[6].HeaderCell.Value = "СевернаяОсетия";

dataGridView6.Rows[7].HeaderCell.Value = "Ставрополье";

dataGridView6.RowHeadersWidth = 150;

God = 1990;

for (int i = 0; i < ChisloStolbcov; i++)

{

dataGridView6.Columns[i].HeaderCell.Value = God.ToString();

God += 1;

}

#endregion

progressBar1.Value = 45;

#region Созданиетаблицыдлясравнениядвухрегионов

dataGridView4.RowCount = 1;

dataGridView4.ColumnCount = ChisloStolbcov;

for (int i = 0; i < ChisloStolbcov; i++) { dataGridView4.Columns[i].Width = 45; }

God = 1990;

for (int i = 0; i < ChisloStolbcov; i++)

{

dataGridView4.Columns[i].HeaderCell.Value = God.ToString();

God += 1;

}

#endregion

progressBar1.Value = 60;

#region 3-хмернаяматрицадляданных

table = new double[12, 8, ChisloStolbcov];

normirovenie = new double[12, 8, ChisloStolbcov];

string[]Bukvi = {"B","C","D","E","F","G","H","I","J","K","L","M"};

for (int i = 0; i < 12; i++)

{

for (int j = 1; j < 9; j++)

{

ObjWorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)ObjWorkBook.Sheets[j];

for (int z = 2; z < 255; z++)

{

Application.DoEvents();

Microsoft.Office.Interop.Excel.Range range = ObjWorkSheet.get\_Range(Bukvi[i] + z.ToString(), Bukvi[i] + z.ToString());

if (range.Text.ToString() == "") { break; }

table[i, j - 1, z - 2] = Convert.ToDouble(range.Text);

}

}

}

#endregion

progressBar1.Value = 80;

#region Нормирование

doublemax = 0, min = 0;

double[] arr = new double[8];

bool flag = true;

for (int l = 0; l < 12; l++)

{

max = 0;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

{

for (int i = 0; i < 8; i++)

{

if (table[l, i, j] == 0) { goto t3; }

if (max < table[l, i, j]) { max = table[l, i, j]; }

if (l == 2) { if (min > table[l, i, j]) { min = table[l, i, j]; } }

t3: ;

}

}

for (int j = 0; j < dataGridView1.ColumnCount; j++)

{

string m1 = Convert.ToString(table[l,0,j]);

if (m1 == "0")

{

for (int i = 0; i < 8; i++)

{

Application.DoEvents();

normirovenie[l, i, j] = 0;

}

goto t1;

}

for (int i = 0; i < 8; i++)

{

if (l == 2)

{

if (flag == true)

{

max = (max - min) + 20;

flag = false;

}

for (int z = 0; z < 8; z++)

{

arr[z] = table[l, z, j] - (min - 20);

}

}

if (l == 2) { normirovenie[l,i,j] = Math.Round((arr[i]) / (max), 3); goto t2; }

if ((l == 0) || (l == 3)|| (l == 5) || (l == 6) || (l == 7) || (l == 8)) { if (table[l, i, j] == 0) { normirovenie[l, i, j] = 0; goto t2; } normirovenie[l, i, j] = Math.Round(1 - (table[l, i, j] / max), 4); goto t2; }

if (table[l, i, j] == 0) { normirovenie[l, i, j] = 0; goto t2; }

normirovenie[l, i, j] = Math.Round(table[l, i, j] / max, 4);

Application.DoEvents();

t2: ;

}

t1: ;

}

}

#endregion

progressBar1.Value = 100;

timer1.Enabled = false;

label1.Visible = true;

comboBox1.Enabled = true;

comboBox2.Enabled = true;

comboBox3.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

MessageBox.Show("Загрузкауспешнозавершена", "", MessageBoxButtons.OK);

label1.Visible = false;

uhodim: ;

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

if (ObjExcel != null) { ObjExcel.Quit(); }

}

private void button2\_Click(object sender, EventArgs e)

{

double summ = 0;

#region Ранжирование

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < ChisloStolbcov; j++)

{

if ((checkBox1.Checked == true) && (normirovenie[0, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[0, i, j], 2); }

if ((checkBox2.Checked == true) && (normirovenie[1, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[1, i, j], 2); }

if ((checkBox3.Checked == true) && (normirovenie[2, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[2, i, j], 2); }

if ((checkBox4.Checked == true) && (normirovenie[3, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[3, i, j], 2); }

if ((checkBox5.Checked == true) && (normirovenie[4, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[4, i, j], 2); }

if ((checkBox6.Checked == true) && (normirovenie[5, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[5, i, j], 2); }

if ((checkBox7.Checked == true) && (normirovenie[6, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[6, i, j], 2); }

if ((checkBox8.Checked == true) && (normirovenie[7, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[7, i, j], 2); }

if ((checkBox9.Checked == true) && (normirovenie[8, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[8, i, j], 2); }

if ((checkBox10.Checked == true) && (normirovenie[9, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[9, i, j], 2); }

if ((checkBox11.Checked == true) && (normirovenie[10, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[10, i, j], 2); }

if ((checkBox12.Checked == true) && (normirovenie[11, i, j] != 0)) { summ = summ + Math.Pow(0 - normirovenie[11, i, j], 2); }

summ = Math.Round(Math.Sqrt(summ), 4);

dataGridView3.Rows[i].Cells[j].Value = Convert.ToString(summ);

summ = 0;

}

}

#endregion

#region НормированиеР

double max = 0, min = 0;

for (int j = 0; j < dataGridView3.ColumnCount; j++)

{

for (int z = 0; z < 8; z++)

{

if (Convert.ToDouble(dataGridView3.Rows[z].Cells[j].Value) == 0) { goto t3; }

if (max < Convert.ToDouble(dataGridView3.Rows[z].Cells[j].Value)) { max = Convert.ToDouble(dataGridView3.Rows[z].Cells[j].Value); }

t3: ;

}

}

for (int j = 0; j < dataGridView3.ColumnCount; j++)

{

for (int i = 0; i < 8; i++)

{

if (Convert.ToDouble(dataGridView3.Rows[i].Cells[j].Value) == 0) { dataGridView5.Rows[i].Cells[j].Value = 0; goto t2; }

dataGridView5.Rows[i].Cells[j].Value = Math.Round((Convert.ToDouble(dataGridView3.Rows[i].Cells[j].Value) - min) / (max - min), 4);

dataGridView6.Rows[i].Cells[j].Value = dataGridView5.Rows[i].Cells[j].Value;

Application.DoEvents();

t2: ;

}

}

#endregion

#region Места

double[] arr = new double[8];

max = 0; int schot = 1, index = 0; bool flag = false;

for (int j = 0; j < dataGridView6.ColumnCount; j++)

{

for (int i = 0; i < 8; i++)

{

if (dataGridView6.Rows[i].Cells[j].Value == null) { dataGridView6.Rows[i].Cells[j].Value = "0"; }

arr[i] = Convert.ToDouble(dataGridView6.Rows[i].Cells[j].Value);

}

for (int z = 0; z < 8; z++)

{

for (int i = 0; i < 8; i++)

{

if ((arr[i] > max) && (arr[i] != 0)) { max = arr[i]; index = i; flag = true; }

}

if (flag == true)

{

dataGridView6.Rows[index].Cells[j].Value = schot;

max = 0;

schot = schot + 1;

arr[index] = 0;

}

flag = false;

}

schot = 1;

}

#endregion

#region Построениеграфика

System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea7 = new System.Windows.Forms.DataVisualization.Charting.ChartArea();

int God = 1990;

this.chart1.ChartAreas.Clear();

this.chart1.ChartAreas.Add(chartArea7);

chart1.Series.Clear();

#region Описаниелинийграфика

var series1 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series1",

Color = System.Drawing.Color.Green,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series2 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series2",

Color = System.Drawing.Color.Red,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series3 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series3",

Color = System.Drawing.Color.Yellow,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series4 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series4",

Color = System.Drawing.Color.Blue,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series5 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series5",

Color = System.Drawing.Color.Orange,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series6 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series6",

Color = System.Drawing.Color.Gray,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series7 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series7",

Color = System.Drawing.Color.Black,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series8 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series8",

Color = System.Drawing.Color.Purple,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

#endregion

this.chart1.Series.Add(series1);

this.chart1.Series.Add(series2);

this.chart1.Series.Add(series3);

this.chart1.Series.Add(series4);

this.chart1.Series.Add(series5);

this.chart1.Series.Add(series6);

this.chart1.Series.Add(series7);

this.chart1.Series.Add(series8);

for (int i = 0; i < ChisloStolbcov; i++)

{

series1.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[0].Cells[i].Value));

series2.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[1].Cells[i].Value));

series3.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[2].Cells[i].Value));

series4.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[3].Cells[i].Value));

series5.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[4].Cells[i].Value));

series6.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[5].Cells[i].Value));

series7.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[6].Cells[i].Value));

series8.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[7].Cells[i].Value));

God = God + 1;

}

chart1.Invalidate();

#endregion

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (tt == true) { label1.Visible = true; tt = false; }

else { label1.Visible = false; tt = true; }

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

double summ = 0;

int m1 = 99, m2 = 99;

if(comboBox2.Text == "Москва") { m1 = 0; }

if(comboBox2.Text == "Тыва") { m1 = 1; }

if(comboBox2.Text == "Дагестан") { m1 = 2; }

if(comboBox2.Text == "Ингушетия") { m1 = 3; }

if(comboBox2.Text == "Кабардино-Балкария") { m1 = 4; }

if(comboBox2.Text == "Карачаево-Черкессия") { m1 = 5; }

if(comboBox2.Text == "Северная Осетия") { m1 = 6; }

if (comboBox2.Text == "Ставрополье") { m1 = 7; }

if (comboBox3.Text == "Москва") { m2 = 0; }

if (comboBox3.Text == "Тыва") { m2 = 1; }

if (comboBox3.Text == "Дагестан") { m2 = 2; }

if (comboBox3.Text == "Ингушетия") { m2 = 3; }

if (comboBox3.Text == "Кабардино-Балкария") { m2 = 4; }

if (comboBox3.Text == "Карачаево-Черкессия") { m2 = 5; }

if (comboBox3.Text == "Северная Осетия") { m2 = 6; }

if (comboBox3.Text == "Ставрополье") { m2 = 7; }

#region Избежание ошибок

if (m1 == 99)

{

MessageBox.Show("Неправильно заполненно левое поле", "Ошибка", MessageBoxButtons.OK);

goto t1;

}

if (m2 == 99)

{

MessageBox.Show("Неправильно заполненно правое поле", "Ошибка", MessageBoxButtons.OK);

goto t1;

}

if (m1 == m2)

{

MessageBox.Show("Был выбран один и тот же регион", "Ошибка", MessageBoxButtons.OK);

goto t1;

}

#endregion

for (int i = 0; i < ChisloStolbcov; i++)

{

if ((checkBox1.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[0, m1, i] - normirovenie[0, m2, i], 2); }

if ((checkBox2.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[1, m1, i] - normirovenie[1, m2, i], 2); }

if ((checkBox3.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[2, m1, i] - normirovenie[2, m2, i], 2); }

if ((checkBox4.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[3, m1, i] - normirovenie[3, m2, i], 2); }

if ((checkBox5.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[4, m1, i] - normirovenie[4, m2, i], 2); }

if ((checkBox6.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[5, m1, i] - normirovenie[5, m2, i], 2); }

if ((checkBox7.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[6, m1, i] - normirovenie[6, m2, i], 2); }

if ((checkBox8.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[7, m1, i] - normirovenie[7, m2, i], 2); }

if ((checkBox9.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[8, m1, i] - normirovenie[8, m2, i], 2); }

if ((checkBox10.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[9, m1, i] - normirovenie[9, m2, i], 2); }

if ((checkBox11.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[10, m1, i] - normirovenie[10, m2, i], 2); }

if ((checkBox12.Checked == true) && (normirovenie[0, m1, i] != 0) && (normirovenie[0, m2, i] != 0)) { summ = summ + Math.Pow(normirovenie[11, m1, i] - normirovenie[11, m2, i], 2); }

summ = Math.Round(Math.Sqrt(summ),4);

dataGridView4.Rows[0].Cells[i].Value = Convert.ToString(summ);

summ = 0;

}

t1: ;

}

private void button4\_Click(object sender, EventArgs e)

{

checkBox1.Checked = true;

checkBox2.Checked = true;

checkBox3.Checked = true;

checkBox4.Checked = true;

checkBox5.Checked = true;

checkBox6.Checked = true;

checkBox7.Checked = true;

checkBox8.Checked = true;

checkBox9.Checked = true;

checkBox10.Checked = true;

checkBox11.Checked = true;

checkBox12.Checked = true;

}

private void chart2\_DoubleClick(object sender, EventArgs e)

{

Form2 form2 = new Form2();

form2.chart2.Visible = true;

#region Описаниелинийграфика

var ser1 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser1",

Color = System.Drawing.Color.Green,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser2 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser2",

Color = System.Drawing.Color.Red,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser3 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser3",

Color = System.Drawing.Color.Yellow,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser4 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser4",

Color = System.Drawing.Color.Blue,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser5 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser5",

Color = System.Drawing.Color.Orange,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser6 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser6",

Color = System.Drawing.Color.Gray,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser7 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser7",

Color = System.Drawing.Color.Black,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var ser8 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Ser8",

Color = System.Drawing.Color.Purple,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

#endregion

#region Построениеграфика

System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea8 = new System.Windows.Forms.DataVisualization.Charting.ChartArea();

int God = 1990;

form2.chart2.ChartAreas.Clear();

form2.chart2.ChartAreas.Add(chartArea8);

form2.chart2.Series.Clear();

form2.chart2.Series.Add(ser1);

form2.chart2.Series.Add(ser2);

form2.chart2.Series.Add(ser3);

form2.chart2.Series.Add(ser4);

form2.chart2.Series.Add(ser5);

form2.chart2.Series.Add(ser6);

form2.chart2.Series.Add(ser7);

form2.chart2.Series.Add(ser8);

for (int i = 0; i < ChisloStolbcov; i++)

{

ser1.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[0].Cells[i].Value));

ser2.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[1].Cells[i].Value));

ser3.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[2].Cells[i].Value));

ser4.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[3].Cells[i].Value));

ser5.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[4].Cells[i].Value));

ser6.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[5].Cells[i].Value));

ser7.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[6].Cells[i].Value));

ser8.Points.AddXY(God, Convert.ToDouble(dataGridView2.Rows[7].Cells[i].Value));

God = God + 1;

}

#endregion

form2.ShowDialog();

}

private void chart1\_DoubleClick(object sender, EventArgs e)

{

Form2 form2 = new Form2();

form2.chart1.Visible = true;

#region Построениеграфика

System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea7 = new System.Windows.Forms.DataVisualization.Charting.ChartArea();

int God = 1990;

form2.chart1.ChartAreas.Clear();

form2.chart1.ChartAreas.Add(chartArea7);

form2.chart1.Series.Clear();

#region Описаниелинийграфика

var series1 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series1",

Color = System.Drawing.Color.Green,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series2 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series2",

Color = System.Drawing.Color.Red,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series3 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series3",

Color = System.Drawing.Color.Yellow,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series4 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series4",

Color = System.Drawing.Color.Blue,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series5 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series5",

Color = System.Drawing.Color.Orange,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series6 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series6",

Color = System.Drawing.Color.Gray,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series7 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series7",

Color = System.Drawing.Color.Black,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

var series8 = new System.Windows.Forms.DataVisualization.Charting.Series

{

Name = "Series8",

Color = System.Drawing.Color.Purple,

IsVisibleInLegend = false,

IsXValueIndexed = true,

ChartType = SeriesChartType.Line

};

#endregion

form2.chart1.Series.Add(series1);

form2.chart1.Series.Add(series2);

form2.chart1.Series.Add(series3);

form2.chart1.Series.Add(series4);

form2.chart1.Series.Add(series5);

form2.chart1.Series.Add(series6);

form2.chart1.Series.Add(series7);

form2.chart1.Series.Add(series8);

for (int i = 0; i < ChisloStolbcov; i++)

{

series1.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[0].Cells[i].Value));

series2.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[1].Cells[i].Value));

series3.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[2].Cells[i].Value));

series4.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[3].Cells[i].Value));

series5.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[4].Cells[i].Value));

series6.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[5].Cells[i].Value));

series7.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[6].Cells[i].Value));

series8.Points.AddXY(God, Convert.ToDouble(dataGridView3.Rows[7].Cells[i].Value));

God = God + 1;

}

#endregion

form2.ShowDialog();

}

}

}

1. http://www.gks.ru/wps/wcm/connect/rosstat\_main/rosstat/ru/statistics/publications/catalog/doc\_1140086420641 [↑](#footnote-ref-2)