

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ (РИНХ)

Факультет компьютерных технологий и информационной безопасности

Кафедра фундаментальной и прикладной математики

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой _____

д.ф.-м.н., доцент Стрюков М.Б.

«_____» _____ 20__ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

**«АНАЛИЗ РАЗВИТИЯ ТРАНСПОРТНОЙ ОТРАСЛИ СУБЪЕКТОВ
ЮЖНОГО ФЕДЕРАЛЬНОГО ОКРУГА
МАТЕМАТИЧЕСКИМИ МЕТОДАМИ С РАЗРАБОТКОЙ
ПРОГРАММНОГО КОМПЛЕКСА»**

Выполнил

студент группы ПМИ-341

И. А. Харитонов

Направление

*01.03.02. Прикладная математика и
информатика*

Направленность

*01.03.02. Математическое и информационное
обеспечение финансово-экономической деятельности*

Руководитель выпускной
квалификационной работы

к.ф.-м.н., доцент

Т. В. Богачев

Ростов-на-Дону, 2020

Оглавление

ВВЕДЕНИЕ.....	4
ГЛАВА 1. РЕГИОНАЛЬНЫЕ ТРАНСПОРТНЫЕ СИСТЕМЫ И ИХ ЭКОНОМИКО-МАТЕМАТИЧЕСКИЙ АНАЛИЗ: ЗАДАЧИ, МЕТОДЫ И СРЕДСТВА.....	6
Роль транспорта в экономике регионов Южного Федерального Округа.....	6
Рассматриваемые регионы ЮФО.....	6
Математическое моделирование.....	8
Среда разработки.....	10
Оптимизация с помощью программных средств.....	10
ГЛАВА 2. ОПИСАНИЕ МАТЕМАТИЧЕСКИХ МЕТОДОВ И ПРОГРАММНЫХ СРЕДСТВ, ИСПОЛЬЗУЕМЫХ ПРИ ПОСТРОЕНИИ МОДЕЛИ РЕГИОНАЛЬНЫХ ГРУЗОПЕРЕВОЗОК.	12
Основные сведения из теории нечетких множеств.....	12
Основные определения.....	12
Операции над нечеткими числами.....	14
Нечеткая линейная регрессия.....	14
Описание возможностей используемых программных средств.....	15
Импорт Данных.....	15
Подготовка данных для анализа.....	16
Создание функций Kosel и optumus.....	19
ГЛАВА 3. АНАЛИЗ РАЗВИТИЯ ТРАНСПОРТНОЙ ОТРАСЛИ СУБЪЕКТОВ ЮЖНОГО ФЕДЕРАЛЬНОГО ОКРУГА	

МАТЕМАТИЧЕСКИМИ МЕТОДАМИ С РАЗРАБОТКОЙ ПРОГРАММНОГО КОМПЛЕКСА	22
Волгоградская область.....	29
Подготовка данных.	29
Интерпретация результата	30
Ростовская область.....	31
Краснодарский край.....	33
Астраханская область	34
Республика Адыгея	35
Республика Калмыкия	37
ВЫВОДЫ.....	40
ЗАКЛЮЧЕНИЕ	41
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	42
ПРИЛОЖЕНИЯ.....	44
Приложение А. Статистические данные показателей развития автомобильных грузоперевозок в ЮФО.....	44

ВВЕДЕНИЕ

Автомобильные грузоперевозки всегда были важнейшим приоритетом и сильной стороной ЮФО. Округ расположен на юге Европейской части Российской Федерации, и включает (с 28 июля 2016 г.) в себя Республику Адыгея, Астраханскую область, Волгоградскую область, Республику Калмыкия, Краснодарский край, Республику Крым, Ростовскую область город Севастополь. Такое расположение способствует развитию транспортной отрасли, что в свою очередь способствует развитию грузовых перевозок (в частности, автомобильных), а общая площадь дорог, удовлетворяющих требование госта, составляет 15386.39, что также способствует развитию грузоперевозок автомобильным транспортом.

Объектом данного исследования является прогнозирование грузовых автомобильных перевозок в ЮФО.

Предмет исследования: математические методы оценки состояния грузовых автомобильных перевозок в ЮФО основанных на теории нечетких множеств.

Основная цель исследования: создание программного комплекса способного в полуавтоматическом формате строить модели линейной регрессии с нечеткими коэффициентами с последующей интерпретаций, проверкой на адекватность и сравнением полученных результатов с прогнозируемыми правительством Ростовской области.

Для достижения этой цели в работе были поставлены следующие задачи:

- 1) изучение текущей стратегии развития автомобильных грузоперевозок;
- 2) изучение теории нечетких множеств для применения их в нечеткой линейной регрессии;
- 3) анализ текущих инструментов для решения данной задачи;

4) реализация программного комплекса наиболее оптимальным (без ущерба эффективности) способом;

5) анализ полученных результатов с последующим сравнением их с программой развития предложенной правительством Ростовской области [1], [2], [3].

Для решения поставленных задач были использованы:

- метод оценки грузовых перевозок на основе теории нечетких множеств;
- программное решение, написанное на языке Python.

Информационной базой для исследования послужил официальный сайт Росстата и статистический сборник “Регионы России.

Используемое программное обеспечение:

- MS Excel;
- Python [4].

В заключении данной работы содержатся описанные результаты работы и основные выводы о проведенном анализе.

В приложении приведены таблицы значений исследуемых показателей по регионам и программный код.

ГЛАВА 1. РЕГИОНАЛЬНЫЕ ТРАНСПОРТНЫЕ СИСТЕМЫ И ИХ ЭКОНОМИКО-МАТЕМАТИЧЕСКИЙ АНАЛИЗ: ЗАДАЧИ, МЕТОДЫ И СРЕДСТВА.

Роль транспорта в экономике регионов Южного Федерального Округа.

Транспорт является системообразующей отраслью, важнейшей составной частью производственной и социальной инфраструктуры региона.

Тенденции развития экономики и социальной сферы ЮФО, ее стратегические интересы являются обоснованной базой определения перспектив, стратегических целей и динамики развития транспортного комплекса. Экономическая специализация промышленности и сельского хозяйства, территориальная дифференциация социальных условий жизни обуславливают пространственные особенности функционирования транспорта региона, формируют специфику межрегиональных транспортных связей [5].

Основу для разработки долгосрочной стратегии развития транспортного комплекса представляет анализ экономического и социального положения области и тенденций ее развития [6].

Рассматриваемые регионы ЮФО.

Для анализа выбраны регионы Южного Федерального округа, для которых доступны данные с 1996 по 2017 г. [7].

Ростовская область занимает 12 место по объему валового регионального продукта (ВРП) в Российской Федерации, ее площадь равна 100 967 км². Протяженность автомобильных дорог общего пользования регионального или межмуниципального значения, соответствующих нормативным требованиям к транспортно-эксплуатационным показателям, равна 4913,44 км (65,6%).

Республика Адыгея занимает 76 место по объему ВРП в Российской Федерации, ее площадь равна 7792 км². Протяженность автомобильных дорог

общего пользования регионального или межмуниципального значения, соответствующих нормативным требованиям к транспортно-эксплуатационным показателям, равна 713,96 км (52,0%).

Астраханская область занимает 55 место по объему ВРП в Российской Федерации, ее площадь равна 49024 км². Протяженность автомобильных дорог общего пользования регионального или межмуниципального значения, соответствующих нормативным требованиям к транспортно-эксплуатационным показателям, равна 662,89 км (30,2%).

Волгоградская область занимает 25 место по объему ВРП в Российской Федерации, ее площадь равна 112877 км². Протяженность автомобильных дорог общего пользования регионального или межмуниципального значения, соответствующих нормативным требованиям к транспортно-эксплуатационным показателям, равна 3273,93 км (33,0%).

Республика Калмыкия занимает 81 место по объему ВРП в Российской Федерации, ее площадь равна 74731 км². Протяженность автомобильных дорог общего пользования регионального или межмуниципального значения, соответствующих нормативным требованиям к транспортно-эксплуатационным показателям, равна 9,73 км (0,8%).

Краснодарский край занимает 5 место по объему ВРП в Российской Федерации, ее площадь равна 75485 км². Протяженность автомобильных дорог общего пользования регионального или межмуниципального значения, соответствующих нормативным требованиям к транспортно-эксплуатационным показателям, равна 5812,44 км (64,9%) [8].

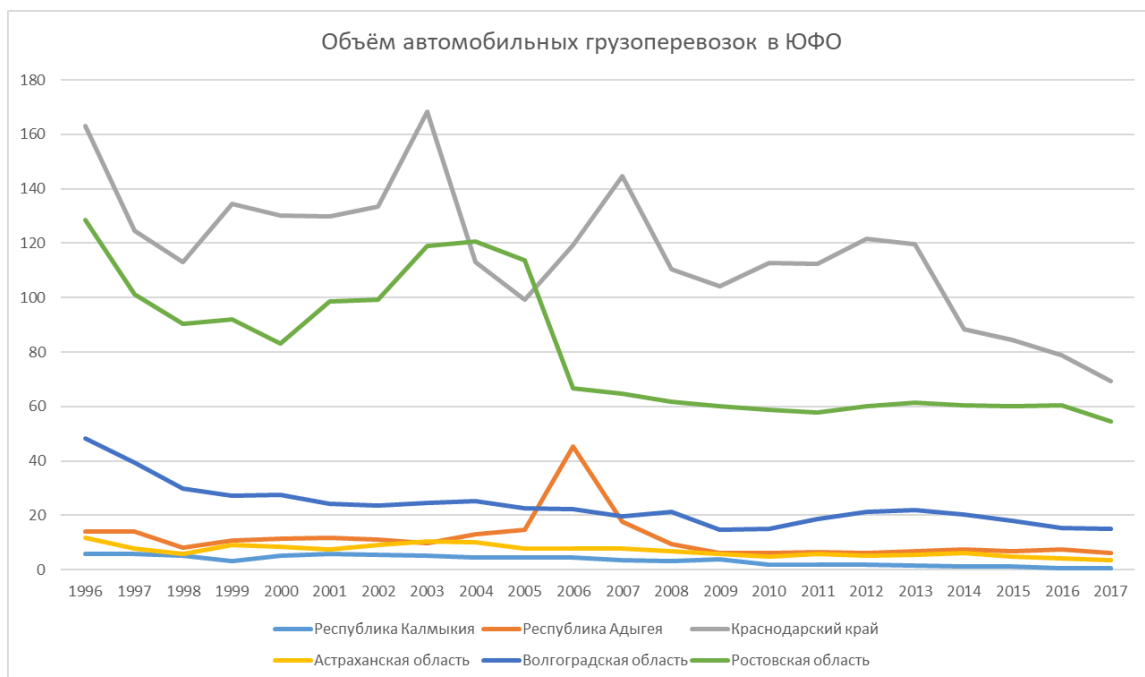


Рисунок 1. Объём автомобильных грузоперевозок ЮФО.

Исходя из представленного графика, можно сделать вывод, что основную роль в автомобильных грузоперевозках ЮФО играет Краснодарский край, с небольшим отставанием идёт Ростовская область, лучшим среди худших оказывается Волгоградская область, и в тройку аутсайдеров входят Республика Калмыкия, Республика Адыгея и Астраханская область.

По всей видимости, такое распределение зависит от исторически сложившихся факторов и непосредственно от площади субъектов, но это работает далеко не всегда, так как, к примеру, Волгоградская область занимает наибольшую площадь в ЮФО, но находится на последних местах по объёмам автомобильных грузовых перевозок.

Математическое моделирование.

В данной работе использовалась так называемая «Регрессионная модель». Регрессионная модель $f(\omega, x)$ — это параметрическое семейство функций, задающее отображение:

$$f : W \times X \rightarrow Y$$

где $w \in W$ — пространство параметров, $x \in X$ — пространство свободных переменных, Y — пространство зависимых переменных.

Так как регрессионный анализ предполагает поиск зависимости матожидания случайной величины от свободных переменных $E(y|x) = f(x)$, то в её состав входит аддитивная случайная величина ε :

$$y = f(w, x) + \varepsilon$$

Предположение о характере распределения случайной величины ν называется гипотезой порождения данных. Эта гипотеза играет центральную роль в выборе критерия оценки качества модели и, как следствие, в способе настройки параметров модели.

Регрессионная модель объединяет широкий класс универсальных функций, которые описывают некоторую закономерность. При этом для построения модели в основном используются измеряемые данные, а не знание свойств исследуемой закономерности. Такая модель часто не интерпретируема, но более точна. Это объясняется либо большим числом моделей-претендентов, которые используются для построения оптимальной модели, либо большой сложностью модели. Нахождение параметров регрессионной модели называется обучением модели.

Недостатки регрессионного анализа: модели, имеющие слишком малую сложность, могут оказаться неточными.

Примеры регрессионных моделей: линейные функции, алгебраические полиномы, ряды Чебышёва, нейронные сети без обратной связи, например, однослойный персептрон Розенблатта, радиальные базисные функции и прочее [9].

В задачах, связанных с реальной жизнью в целом и экономикой в частности, данные поступают в недостаточном объеме или в искаженном виде и, следовательно, содержат в себе неопределенность. При моделировании различных экономических процессов неопределенность может быть не только в

анализируемых данных, но часто имеет место неопределенность, связанная с неучтенными факторами. Поэтому целесообразно использовать методы теории нечетких множеств. Как говорил ее создатель Л.Заде: «Излишнее стремление к точности стало оказывать действие, сводящее на нет теорию управления и теорию систем, так как оно приводит к тому, что исследования в этой области сосредоточиваются на тех и только тех проблемах, которые поддаются точному решению. Многие классы важных проблем, в которых данные, цели и ограничения являются слишком сложными или плохо определенными для того, чтобы допустить точный математический анализ, оставались и остаются в стороне лишь по той причине, что они не поддаются математической трактовке» [10].

Модель нечеткой линейной регрессии соединяет в себе преимущества нечетко-логического подхода и классической линейной оптимизации.

Среда разработки

Метод нечеткого моделирования реализован в нескольких средах (R [11], MATLAB [12]), но был выбран язык Python, так как он ориентирован на повышение производительности разработчика и читаемости кода. Помимо реализации регрессии, представленное программное решение задачи также осуществляет оформление полученного результата в файле Excel.

Оптимизация с помощью программных средств

Благодаря развитию IT области появилась возможность использовать сложные математические методы без необходимой для этого базы знаний.

В данной работе используется библиотека CVXOPT [13] для решения задач выпуклой оптимизации, в частности, мы будем использовать функцию `solvers.lp()` для решения задачи линейного программирования, которая появляется непосредственно из выбранного нами способа прогнозирования. Так как в целях данной работы нет описания работы отдельных функций из отдельно

взятых библиотек, то ограничимся ссылкой на документацию, в которой авторы досконально расписали все используемые ими математические методы.

ГЛАВА 2. ОПИСАНИЕ МАТЕМАТИЧЕСКИХ МЕТОДОВ И ПРОГРАММНЫХ СРЕДСТВ, ИСПОЛЬЗУЕМЫХ ПРИ ПОСТРОЕНИИ МОДЕЛИ РЕГИОНАЛЬНЫХ ГРУЗОПЕРЕВОЗОК.

Основные сведения из теории нечетких множеств.

Основные определения.

Под нечетким множеством A понимается совокупность упорядоченных пар, составленных из элементов x универсального множества X и соответствующих степеней принадлежности $\mu_A(x)$, где $A = \{(x, \mu_A(x)) \mid x \in X\}$.

Примеры некоторых функций принадлежности:

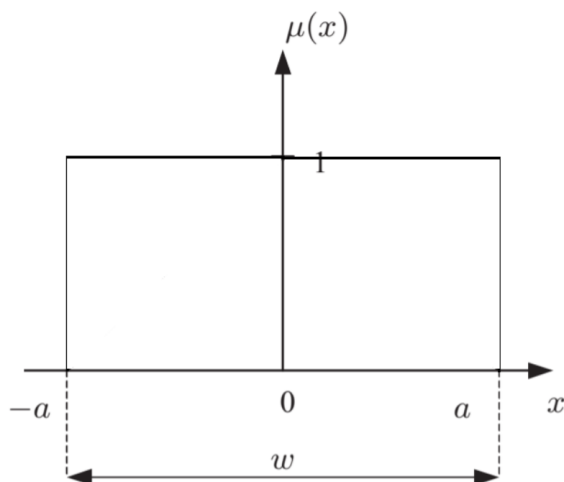


График 1. Функция принадлежности.

Непрерывная характеристическая функция принадлежности элементов множества действительных чисел к четкому множеству $-a \leq x \leq a$.

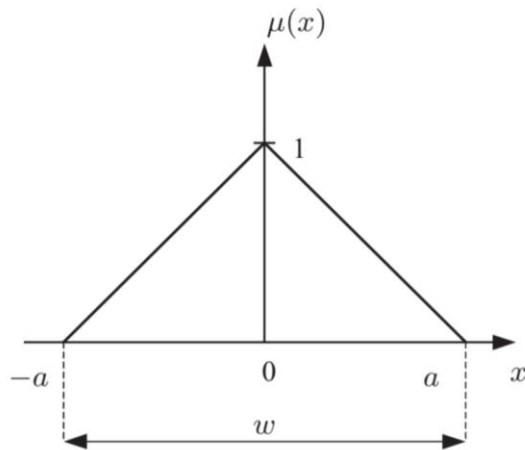


График 2. Функция принадлежности.

Непрерывная функция принадлежности элементов множества действительных чисел к нечёткому множеству чисел «примерно ноль».

$$\mu_A(x) = \frac{a - |x|}{a}, -a \leq x \leq a.$$

$$A = \left\{ \frac{0}{-a}; \dots; \frac{1}{0}; \dots; \frac{0.5}{\frac{a}{2}}; \dots; \frac{0}{a} \right\}.$$

Функция принадлежности $\mu_A(x)$ указывает, в какой степени элемент x принадлежит нечёткому множеству A . Функция $\mu_A(x)$ принимает значения в некотором линейно упорядоченном множестве M . Множество M называют множеством принадлежностей, часто в качестве M выбирается отрезок $[0,1]$. Если $M = \{0,1\}$, то нечёткое множество может рассматриваться как обычное чёткое множество.

Треугольным числом B называется тройка действительных чисел $a \leq b \leq c$ с функцией принадлежности:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & \text{if } x \in [a, b], \\ \frac{x-c}{b-c} & \text{if } x \in [b, c], \\ 0 & \text{else.} \end{cases}$$

Операции над нечеткими числами.

Суммирование двух нечетких чисел A_1 и A_2 обозначается $A_1 \oplus A_2 = B$.

При этом функция принадлежности $\mu_B(y) = \sup_{x_1, x_2} \min \mu_{A_1}(x_1), \mu_{A_2}(x_2)$, где $y = x_1 + x_2$.

В качестве примера сложим два нечетких числа:

$$A_1 = \left\langle \frac{0.7}{2}, \frac{1}{3}, \frac{0.6}{4} \right\rangle, A_2 = \left\langle \frac{0.8}{3}, \frac{1}{4}, \frac{0.5}{6} \right\rangle$$

$$A_1 \oplus A_2 = \frac{\min(0.7; 0.8)}{5} + \frac{\max\{\min(0.7; 1), \min(1; 0.8)\}}{6} + \frac{\max\{\min(1; 1), \min(0.6; 0.8)\}}{7} + \frac{\max\{\min(0.7; 0.5), \min(0.6; 1)\}}{8} + \frac{\min(1; 0.5)}{9} + \frac{\min(0.6; 0.5)}{10} = \frac{0.7}{5} + \frac{0.8}{6} + \frac{1}{7} + \frac{0.6}{8} + \frac{0.5}{9} + \frac{0.5}{10}$$

Нечеткая линейная регрессия.

Пусть задача нечеткой линейной регрессии [14] задана таким образом:

Для данных k результатов наблюдений зависимой переменной y_j от n факторов $x_i = (x_{i1}, x_{i2}, \dots, x_{ik})^T$, где $i = 1, \dots, n$, нужно найти нечеткие коэффициенты A_0, A_1, \dots, A_n такие, чтобы выполнялись условия $\mu_j(Y_j) \geq h$, где $\mu_j(Y_j)$ – функция принадлежности нечеткого множества, $Y_j = A_0 + A_1 x_{1j} + \dots + A_n x_{nj}$, величина h – заданный порог надежности, причем неопределенность, связанная с этими коэффициентами, была бы минимальной. Пусть параметрами модели являются числа $A_i = \langle a_i - r_i, a_i, a_i + r_i \rangle$, где $a_i \in R$ и $r_i \geq 0$. Для нахождения a_i и r_i получаем задачу линейного программирования:

$$f = k a_0 + \sum_{j=1}^k \sum_{i=1}^n r_i x_{ij} \rightarrow \min$$

$$\begin{cases} y_j \geq a_0 + \sum_{i=1}^n a_i x_{ij} - (1-h) \left(r_0 + \sum_{i=1}^n r_i x_{ij} \right), \\ y_j \leq a_0 + \sum_{i=1}^n a_i x_{ij} + (1-h) \left(r_0 + \sum_{i=1}^n r_i x_{ij} \right), \\ r_i \geq 0, j = 1 \dots k. \end{cases}$$

Описание возможностей используемых программных средств.

Для наглядности опишем все используемые, для создания программного решения, средства с помощью сравнения реализации модели нечёткой линейной регрессии на языке программирования Python с аналогичной реализацией в табличном процессоре Excel.

Импорт Данных.

EXCEL

Для того чтобы импортировать данные в Excel (не учитывая возможность парсинга), достаточно просто вставить их в рабочий лист.

Возьмём в качестве тестовых данных небольшую выборку, состоящую из двух показателей X и Y, где Y объясняемая переменная, а X объясняющая, объёмом в 10 значений каждая.

	A	B	C	D
1		X	Y	
2	1	40	100	
3	2	47	130	
4	3	50	145	
5	4	64	170	
6	5	38	90	
7	6	45	120	
8	7	30	36	
9	8	69	190	
10	9	73	210	
11	10	86	50	
12				

Рисунок 2. Импорт данных в Excel.

PYTHON

Импорт данных в Python доступен в различных вариантах (от импорта CSV файлов, до парсинга отдельных сайтов). В рамках данной работы остановимся на импорте данных из файлов Excel.

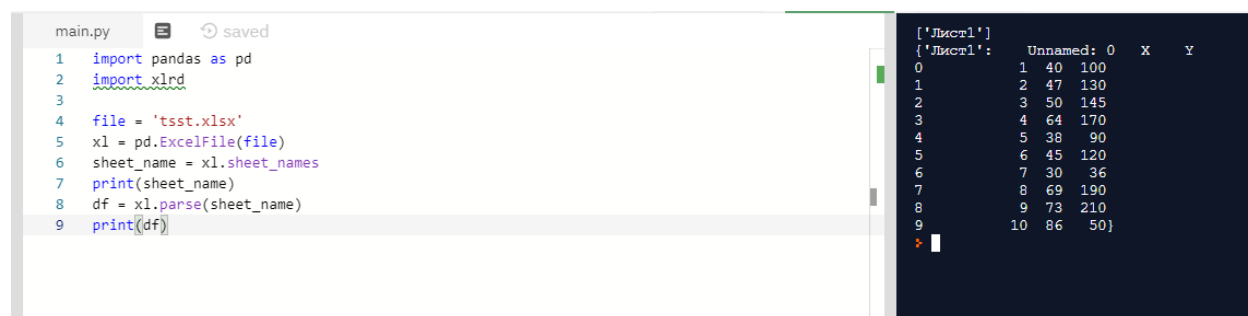
Перед началом работы с файлами импортируем библиотеки:

- `import pandas as pd` для работы с импортируемым файлом [15];
- `import xlrd` для непосредственно возможности импортировать данные.

Первоначально необходимо инициализировать переменную `file` и передать в неё строковое значение с путём до импортированного файла.

С помощью функции `ExcelFile` и библиотеки `Pandas` переведём импортированный файл в формат данных `Dataframe`.

С помощью функции `sheet_names` проверим, что мы можем получить название листа импортированного файла Excel и выведем в консоль непосредственно импортированные данные.



```
main.py saved
1 import pandas as pd
2 import xlrd
3
4 file = 'tsst.xlsx'
5 xl = pd.ExcelFile(file)
6 sheet_name = xl.sheet_names
7 print(sheet_name)
8 df = xl.parse(sheet_name)
9 print(df)
```

```
['Лист1']
{'Лист1': Unnamed: 0  X  Y
0      1  40  100
1      2  47  130
2      3  50  145
3      4  64  170
4      5  38  90
5      6  45  120
6      7  30  36
7      8  69  190
8      9  73  210
9     10  86  50}
```

Рисунок 3. Импорт данных в Python.

Подготовка данных для анализа.

EXCEL

В первую очередь необходимо определить ячейки для хранения данных нечетких и классических показателей нечеткой линейной регрессии. Так как в нашем примере всего один объясняющий показатель, то будут необходимы 4 коэффициента: два нечетких (p_0, p_1), два классических (a_0, a_1) и значение целевой функции p [16].

	A	B	C	D	E	F
1	a0	a1	p0	p1	p	
2						
3						
4						
5						
6		X	Y			
7	1	40	100			
8	2	47	130			
9	3	50	145			
10	4	64	170			
11	5	38	90			
12	6	45	120			
13	7	30	36			
14	8	69	190			
15	9	73	210			
16	10	86	50			
17						
18						

Рисунок 4. Резервирование переменных для коэффициентов.

Поскольку в модели нечеткой линейной регрессии необходимо использовать два почти идентичных набора уравнений для системы ограничений, то выбранный набор данных необходимо продублировать.

Рассчитаем сумму каждого объясняющего показателя по той же необходимости.

	A	B	C	D	E	F	G
1	h	a0	a1	p0	p1	p	
2							
3							
4							
5							
6		X	Y				
7	1	40	100	>=	=B\$2+B7*\$C\$2-(1-\$A\$2)*(\$D\$2+B7*\$E\$2)		
8	2	47	130	>=	=B\$2+B8*\$C\$2-(1-\$A\$2)*(\$D\$2+B8*\$E\$2)		
9	3	50	145	>=	=B\$2+B9*\$C\$2-(1-\$A\$2)*(\$D\$2+B9*\$E\$2)		
10	4	64	170	>=	=B\$2+B10*\$C\$2-(1-\$A\$2)*(\$D\$2+B10*\$E\$2)		
11	5	38	90	>=	=B\$2+B11*\$C\$2-(1-\$A\$2)*(\$D\$2+B11*\$E\$2)		
12	6	45	120	>=	=B\$2+B12*\$C\$2-(1-\$A\$2)*(\$D\$2+B12*\$E\$2)		
13	7	30	36	>=	=B\$2+B13*\$C\$2-(1-\$A\$2)*(\$D\$2+B13*\$E\$2)		
14	8	69	190	>=	=B\$2+B14*\$C\$2-(1-\$A\$2)*(\$D\$2+B14*\$E\$2)		
15	9	73	210	>=	=B\$2+B15*\$C\$2-(1-\$A\$2)*(\$D\$2+B15*\$E\$2)		
16	10	86	50	>=	=B\$2+B16*\$C\$2-(1-\$A\$2)*(\$D\$2+B16*\$E\$2)		
17		=СУММ(B7:B16)					
18							
19							
20		X	Y				
21	1	40	100	<=	=B\$2+B21*\$C\$2-(1-\$A\$2)*(\$D\$2+B21*\$E\$2)		
22	2	47	130	<=	=B\$2+B22*\$C\$2-(1-\$A\$2)*(\$D\$2+B22*\$E\$2)		
23	3	50	145	<=	=B\$2+B23*\$C\$2-(1-\$A\$2)*(\$D\$2+B23*\$E\$2)		
24	4	64	170	<=	=B\$2+B24*\$C\$2-(1-\$A\$2)*(\$D\$2+B24*\$E\$2)		
25	5	38	90	<=	=B\$2+B25*\$C\$2-(1-\$A\$2)*(\$D\$2+B25*\$E\$2)		
26	6	45	120	<=	=B\$2+B26*\$C\$2-(1-\$A\$2)*(\$D\$2+B26*\$E\$2)		
27	7	30	36	<=	=B\$2+B27*\$C\$2-(1-\$A\$2)*(\$D\$2+B27*\$E\$2)		
28	8	69	190	<=	=B\$2+B28*\$C\$2-(1-\$A\$2)*(\$D\$2+B28*\$E\$2)		
29	9	73	210	<=	=B\$2+B29*\$C\$2-(1-\$A\$2)*(\$D\$2+B29*\$E\$2)		
30	10	86	50	<=	=B\$2+B30*\$C\$2-(1-\$A\$2)*(\$D\$2+B30*\$E\$2)		
31		=СУММ(B21:B30)					
32							

Рисунок 5. Структура системы уравнений для модели нечеткой линейной регрессии.

Следующий шаг заключается в использовании «Поиска решений» (активация данного функционала происходит через Файл → Параметры → Надстройки → Поиск решения → Перейти → выбрать «Поиск решения» → ОК).

Отметим, что ячейка для целевой функции должна содержать в себе сумму произведений, указанную в определении модели нечеткой регрессии.

	A	B	C	D	E	F	G
1	h	a0	a1	p0	p1	p	
2	0,5	28,50001	1,368151	0	2,236302	1212,075	
3							
4							
5							
6		X	Y				
7	1	40	100	>=	38,5		
8	2	47	130	>=	40,25		
9	3	50	145	>=	41		
10	4	64	170	>=	44,49999		
11	5	38	90	>=	38		
12	6	45	120	>=	39,75		
13	7	30	36	>=	36		
14	8	69	190	>=	45,74999		
15	9	73	210	>=	46,74999		
16	10	86	50	>=	49,99999		
17		542					
18							
19							
20							
21		X	Y				
22	1	40	100	<=	127,9521		
23	2	47	130	<=	145,3562		
24	3	50	145	<=	152,8151		
25	4	64	170	<=	187,6233		
26	5	38	90	<=	122,9795		
27	6	45	120	<=	140,3836		
28	7	30	36	<=	103,089		
29	8	69	190	<=	200,0548		
30	9	73	210	<=	210		
31	10	86	50	<=	242,3219		
32							

Рисунок 6. Решение задачи линейного программирования для модели нечёткой линейной регрессии.

PYTHON

Создание функций `equation_system_generator` и `target_function_generator`.

Перед тем как непосредственно реализовывать функцию, оптимизирующую проделываемую работу, приведём общий вид ожидаемого результата.

Для удобства обозначим x как список переменных, а X – как массив [17] со всеми данными. В нашем первоначальном примере $x = [a_0, a_1, p_0, p_1]$ а $X = [[40,100],[47,130], \dots [86,50]]$ соответственно ожидаем, что при обращении к $x[0]$ получим свободный член a_0 , а при обращении к $X[2][1]$ получим 47 (второе значение первого столбца с данными).

Приведём общий вид генерируемых данных.

Система ограничений:

$$\begin{cases}
 z = N \times x[n] + x[n+1] \times X_{sum}[0] + x[n+2] \times X_{sum}[1] + \dots + x[2 \times n - 1] \times X_{sum}[n-2] \\
 x[0] + x[1] \times X[0][0] + x[2] \times X[0][1] \dots x[n-1] \times X[0][n-2] - (1-h) \times (x[n] + x[n+1]) \times X[0][0] + x[n+2] \times X[0][1] + \dots + x[2 \times n - 1] \times X[0][n-2] \leq X[0][n-1] \\
 \vdots \\
 x[0] + x[1] \times X[k-1][0] + x[2] \times X[k-1][1] \dots x[n-1] \times X[k-1][n-2] - (1-h) \times (x[n] + x[n+1]) \times X[k-1][0] + x[n+2] \times X[k-1][1] + \dots + x[2 \times n - 1] \times X[k-1][n-2] \leq X[k-1][n-1] \\
 x[0] + x[1] \times X[0][0] + x[2] \times X[0][1] \dots x[n-1] \times X[0][n-2] - (1-h) \times (x[n] + x[n+1]) \times X[0][0] + x[n+2] \times X[0][1] + \dots + x[2 \times n - 1] \times X[0][n-2] \geq X[0][n-1] \\
 \vdots \\
 x[0] + x[1] \times X[k-1][0] + x[2] \times X[k-1][1] \dots x[n-1] \times X[k-1][n-2] - (1-h) \times (x[n] + x[n+1]) \times X[k-1][0] + x[n+2] \times X[k-1][1] + \dots + x[2 \times n - 1] \times X[k-1][n-2] \geq X[k-1][n-1] \\
 x[n] \geq 0 \\
 \vdots \\
 x[2 \times n - 1] \geq 0
 \end{cases}$$

Функция `equation_system_generator` принимает два аргумента — количество строк и столбцов таблицы и после преобразований возвращает список с переменными.

```

1. def equation_system_generator(z, y):
2.     z = z * 2
3.     p = []
4.     for k in range(0, y * 2):
5.         if(k >= y):
6.             per = "(x[0]"
7.             for i in range(1, z):
8.                 if (i < (z/2)):
9.                     per += "+ float(X"+"[" + str(k - y) + "]" + "[" +
10.                        str(i - 1) + "])*x" + "[" + str(i) + "]"
11.                     if(i == (z / 2)):
12.                         if(k >= y):
13.                             per += " + (1 - h)*(" + "x[" + str(i) + "]"
14.                             else:
15.                                 per += " - (1 - h)*(" + "x[" + str(i) + "]"
16.
17.                     if(i > (z / 2) and i < z - 1):
18.                         per += " + float(X" + "[" + str(k - y) + "]" +
19.                            "[" + str((i - 1) - int(z / 2)) + "])*x[" + str(i) + "]"
20.                     if(i == (z - 1)):
21.                         if(k >= y):

```

```

21.         per += " + float(X" + "[" + str(k - y) + "]" +
    "[" + str((i - 1) - int(z / 2)) + "]" + "x[" + str(i) + "])" + " >= " +
    float(X" + "[" + str(k - y) + "]" + "[" + str(i - int(z / 2)) + "])"
22.         else:
23.             per += " + float(X" + "[" + str(k - y) + "]" +
    "[" + str((i - 1) - int(z / 2)) + "]" + "x[" + str(i) + "])" + " <= " +
    float(X" + "[" + str(k - y) + "]" + "[" + str(i - int(z / 2)) + "])"
24.
25.     else:
26.         per = "(x[0]"
27.         for i in range(1, z):
28.             if (i < (z / 2)):
29.                 per += "+ float(X" + "[" + str(k) + "]" + "[" +
    str(i - 1) + "])*x[" + str(i) + "]"
30.             if(i == (z / 2)):
31.                 if(k >= y):
32.                     per += " + (1 - h)*(" + "x[" + str(i) + "]"
33.                 else:
34.                     per += " - (1 - h)*(" + "x[" + str(i) + "]"
35.
36.
37.
38.             if(i > (z / 2) and i < z - 1):
39.                 per += " + float(X" + "[" + str(k) + "]" + "[" +
    str((i - 1) - int(z / 2)) + "]" + "x[" + str(i) + "]"
40.             if(i == (z - 1)):
41.                 if(k >= y):
42.                     per += " + float(X" + "[" + str(k) + "]" + "[" +
    + str((i - 1) - int(z / 2)) + "]" + "x[" + str(i) + "])" + " >= " +
    float(X" + "[" + str(k) + "]" + "[" + str(i - int(z / 2)) + "])"
43.                 else:
44.                     per += " + float(X" + "[" + str(k) + "]" + "[" +
    + str((i - 1) - int(z / 2)) + "]" + "x[" + str(i) + "])" + " <= " +
    float(X" + "[" + str(k) + "]" + "[" + str(i - int(z / 2)) + "])"
45.         p.append(eval(per))
46.
47.     for i in range(int(z / 2)):
48.         Z = "(x[" + str(i + int(z / 2)) + "]) >= 0"
49.         p.append(eval(Z))
50.     print("\n\n", p, "\n\n")
51.     return p

```

Функция `target_function_generator` принимает один аргумент — количество столбцов и после преобразований возвращает целевую функцию.

```

1. def target_function_generator(number_of_columns):
2.     target_function = "( N*x[" + str(number_of_columns) + "]"
3.     for i in range(1, int(number_of_columns)):
4.         target_function += "+x[" + str(i + int(number_of_columns)) +
    "]*float(X_sum[" + str(i - 1) + "])"
5.     target_function += ")"
6.     print(target_function)
7.     return eval(target_function)

```

ГЛАВА 3. АНАЛИЗ РАЗВИТИЯ ТРАНСПОРТНОЙ ОТРАСЛИ СУБЪЕКТОВ ЮЖНОГО ФЕДЕРАЛЬНОГО ОКРУГА МАТЕМАТИЧЕ- СКИМИ МЕТОДАМИ С РАЗРАБОТКОЙ ПРОГРАММНОГО КОМПЛЕКСА

Для анализа показателей развития информационно-коммуникационных технологий регионах Южного федерального округа по каждому субъекту были собраны статистические данные [18], [19].

- Республика Адыгея;
- Республика Калмыкия;
- Краснодарский край;
- Астраханская область;
- Волгоградская область;
- Ростовская область.

Был сформирован набор показателей, характеризующих развитие автомобильных грузоперевозок с 1996 по 2017 года (таблицы в приложении).

Теперь, когда имеются наборы показателей по каждому субъекту, можно воспользоваться программным комплексом для расчета модели, основанной на теории нечетких множеств. Для этого создадим файл Excel с 6 листами, где первые столбцы будут отведены для объясняющих показателей, а последний столбец для объясняемой переменной.

В процессе работы скрипта из каждого листа экспортируются данные, строится система уравнений для 10 различных уровней достоверности и решается каждая из этих систем, по найденным коэффициентам a и p вычисляются границы нечёткого интервала. Данные, по которым можно построить графики и значение коэффициентов импортируются в файл Excel. Для каждого региона будет создан отдельный файл с 20 листами, где на нечётных листах находятся данные для построения графиков, а на чётных листах значение коэффициентов для построения уравнения.

Ниже представлен код основной части программного комплекса

```
1. import pandas as pd
2. import xlrd
3. from itertools import chain
4. import xlwt
5. import openpyxl
6. import xlswriter
7. import numpy as np
8. from cvxopt.modeling import variable, op
9.
10.
11. def equation_system_generator(z, y):
12.     z = z * 2
13.     p = []
14.     for k in range(0, y * 2):
15.         if(k >= y):
16.             per = "(x[0]"
17.             for i in range(1, z):
18.                 if (i < (z/2)):
19.                     per += "+ float(X"+"[" + str(k - y) + "]" + "[" +
20. str(i - 1) + "])*x" + "[" + str(i) + "]"
21.                 if(i == (z / 2)):
22.                     if(k >= y):
23.                         per += " + (1 - h)*(" + "x[" + str(i) + "]"
24.                     else:
25.                         per += " - (1 - h)*(" + "x[" + str(i) + "]"
26.
27.                 if(i > (z / 2) and i < z - 1):
28.                     per += " + float(X" + "[" + str(k - y) + "]" +
29. "[" + str((i - 1) - int(z / 2)) + "])*x[" + str(i) + "]"
30.                 if(i == (z - 1)):
31.                     if(k >= y):
32.                         per += " + float(X" + "[" + str(k - y) + "]" +
33. "[" + str((i - 1) - int(z / 2)) + "])*x[" + str(i) + "])" + " >= " +
34. "float(X" + "[" + str(k - y) + "]" + "[" + str(i - int(z / 2)) + "])"
35.                     else:
36.                         per += " + float(X" + "[" + str(k - y) + "]" +
37. "[" + str((i - 1) - int(z / 2)) + "])*x[" + str(i) + "])" + " <= " +
38. "float(X" + "[" + str(k - y) + "]" + "[" + str(i - int(z / 2)) + "])"
39.
40.             else:
41.                 per = "(x[0]"
42.                 for i in range(1, z):
43.                     if (i < (z / 2)):
44.                         per += "+ float(X" + "[" + str(k) + "]" + "[" +
45. str(i - 1) + "])*x" + "[" + str(i) + "]"
46.                     if(i == (z / 2)):
47.                         if(k >= y):
48.                             per += " + (1 - h)*(" + "x[" + str(i) + "]"
49.                         else:
50.                             per += " - (1 - h)*(" + "x[" + str(i) + "]"
```

```

46.
47.
48.         if(i > (z / 2) and i < z - 1):
49.             per += " + float(X" + "[" + str(k) + "]" + "[" +
+ str((i - 1) - int(z / 2)) + "])*x[" + str(i) + "]"
50.             if(i == (z - 1)):
51.                 if(k >= y):
52.                     per += " + float(X" + "[" + str(k) + "]" + "[" +
+ str((i - 1) - int(z / 2)) + "])*x[" + str(i) + "]))" + " >= " +
+ "float(X" + "[" + str(k) + "]" + "[" + str(i - int(z / 2)) + "]))"
53.                 else:
54.                     per += " + float(X" + "[" + str(k) + "]" + "[" +
+ str((i - 1) - int(z / 2)) + "])*x[" + str(i) + "]))" + " <= " +
+ "float(X" + "[" + str(k) + "]" + "[" + str(i - int(z / 2)) + "]))"
55.             p.append(eval(per))
56.
57.     for i in range(int(z / 2)):
58.         Z = "(x[" + str(i + int(z / 2)) + "]) >= 0"
59.         p.append(eval(Z))
60.     print("\n\n", p, "\n\n")
61.     return p
62.
63.
64. def target_function_generator(number_of_columns):
65.     target_function = "( N*x[" + str(number_of_columns) + "]"
66.     for i in range(1, int(number_of_columns)):
67.         target_function += "+x[" + str(i + int(number_of_columns)) +
+ "]"*float(X_sum[" + str(i - 1) + "])"
68.     target_function += ")"
69.     print(target_function)
70.     return eval(target_function)
71.
72.
73. def date_list_generator():
74.     beginning_of_period = 1996
75.     end_of_period = 2017
76.     years = []
77.     for i in range((end_of_period + 1) - beginning_of_period):
78.         years.append(i + beginning_of_period)
79.     return(years)
80.
81.
82. def converter_of_multidimensional_lists_to_oneDimensional(temp_list):
83.     for ele in temp_list:
84.         if type(ele) == list:
85.             converter_of_multidimensional_lists_to_oneDimensional(ele)
86.         else:
87.             new_list.append(ele)
88.
89.
90. def coefficient_generator(name_of_indicators):
91.     number_of_indicators = len(name_of_indicators)
92.     fuzzy_odds = []
93.     classic_odds = []

```



```

94.     for i in range(number_of_indicators):
95.         fuzzy_odds.append('a' + str(i))
96.         classic_odds.append('r' + str(i))
97.     odds = ['h'] + fuzzy_odds + classic_odds + ['f']
98.     return(odds)
99.
100.
101.     def coefficient_name_generator(name_of_indicators):
102.         name_of_indicators = name_of_indicators[:-1]
103.         fuzzyness = "нечеткость"
104.         free_member = "свободный член"
105.         function_value = "значение функции"
106.         name_of_indicators = [fuzzyness]+[free_member]+name_of_indica-
107.             tors+[free_member]+name_of_indicators+[function_value]
108.         return(name_of_indicators)
109.
110.     file = 'tsst2.xlsx'
111.
112.     # на это число делится h в цикле что бы посмотреть на разлчные уровни
113.     нечёткости
114.     the_number_of_parts_into_which_the_confidence_level_is_split = 10
115.     m = 8 # Количество столбцов необходимых для конечного ответа
116.     # m=0 - хранит в себе значение уровня нечеткости
117.     # m=1 - хранит себе значение четких коэффициентов
118.     # m=2 - хранит в себе значение нечетких коэффициентов
119.     # m=3 - хранит в себе значение целевой функции
120.     # m=4 - хранит в себе значение центральной прямой нашей модели
121.     # m=5 - хранит в себе значение верхней прямой нашей модели
122.     # m=6 - хранит в себе значение нижней прямой нашей модели
123.     # m=7 - хранит в себе значение прямой содержащей реальные данные
124.     list_to_answer = [[0] * m for i in range(the_num-
125.         ber_of_parts_into_which_the_confidence_level_is_split)]
126.     list_for_the_final_answer = []
127.     work_excel_file = pd.ExcelFile(file)
128.     worksheet_names = work_excel_file.sheet_names
129.     for worksheet_name in worksheet_names:
130.         raw_dataframe = work_excel_file.parse(worksheet_name)
131.         X = raw_dataframe.to_numpy()
132.         dimension = X.shape
133.         the_number_of_rows_before_the_creation_of_the_control_sample =
134.             dimension[0]
135.         number_of_indicators = dimension[1]
136.         # уберем последние 4 значения из строк для того что бы они участво-
137.         вали в проверке адекватности модели
138.         the_number_of_rows_after_the_creation_of_the_control_sample =
139.             the_number_of_rows_before_the_creation_of_the_control_sample - 4
140.         # данная переменная участвует в работе генератора системы уравне-
141.         ний и я не уверен можно ли её убирать так что пусть будет
142.         N = the_number_of_rows_after_the_creation_of_the_control_sample
143.         #сумма столбцов показателей
144.         X_sum = []

```

```

141.         for i in range(int(number_of_indicators - 1)):
142.             z = 0
143.             for j in range(int(the_number_of_rows_after_the_crea-
            tion_of_the_control_sample)):
144.                 z += X[j][i]
145.                 X_sum.append(z)
146.
147.             # посчитаем модель с учётом различной нечеткости от 0 до 0.9 с
            шагом 0.1
148.             for h in range(the_number_of_parts_into_which_the_conf-
            dence_level_is_split):
149.                 # на данном шаге заведём новую переменную для того что бы сохра-
            нить переменную h которая будет нужна для того что бы
150.                 # заносить данные в нужные места в цикле ( что будет продемон-
            стрированно далее)
151.                 hh = h
152.                 # переменная h теперь будет пробегать значения от 0 до 0.9 с ша-
            гом 0.1 а переменная hh будет пробегать значения от 0 до 9
153.                 h = h / the_number_of_parts_into_which_the_conf-
            dence_level_is_split
154.                 # введем переменную для того что бы проверить не выходят ли ре-
            альные данные за рамки построенные моделью
155.                 test = 0
156.                 # заводим переменную x в два раза больше чем количество столбцов
            ведь в задаче линейного программирования
157.                 # для нечеткой регрессии ограничений необходимо в два раза больше
            чем переменных
158.                 x = variable(number_of_indicators * 2, 'x')
159.                 problem = op(target_function_generator(number_of_indicators),
            equation_system_generator(number_of_indicators,the_num-
            ber_of_parts_into_which_the_confidence_level_is_split))
160.                 problem.solve(solver = 'glpk')
161.                 problem.status
162.                 objective_function_value = (problem.objective.value()[0])
163.
164.                 classic_odds = []
165.                 fuzzy_odds = []
166.                 for i in range(number_of_indicators):
167.                     classic_odds.append(x.value[i])
168.                 for i in range(number_of_indicators,int(number_of_indica-
            tors) * 2):
169.                     fuzzy_odds.append(x.value[i])
170.
171.                 list_to_answer[hh][0] = h
172.                 list_to_answer[hh][1] = classic_odds
173.                 list_to_answer[hh][2] = fuzzy_odds
174.                 list_to_answer[hh][3] = objective_function_value
175.
176.
177.                 # возвращаем 4 строки обратно так как график нам нужен по
            всей выборке
178.                 the_number_of_rows_required_to_validate_the_model = the_num-
            ber_of_rows_before_the_creation_of_the_control_sample
179.

```

```

180.
181.     #####
182.     #####реальные данные####
183.     #####
184.     realY = []
185.
186.
187.     for j in range(the_number_of_rows_required_to_vali-
date_the_model):
188.         realY.append(X[j][number_of_indicators - 1])
189.
190.
191.     list_to_answer[hh][7] = realY
192.     #####
193.     ###центральные данные####
194.     #####
195.     dotY = []
196.
197.
198.     for j in range(the_number_of_rows_required_to_vali-
date_the_model):
199.         zz = classic_odds[0]
200.         for i in range(1, int(len(classic_odds))):
201.             zz += classic_odds[i] * X[j][i - 1]
202.         dotY.append(zz)
203.
204.
205.     list_to_answer[hh][4] = dotY
206.     #####
207.     ###вверхние данные####
208.     #####
209.     dotY_right = []
210.
211.
212.     for j in range (the_number_of_rows_required_to_vali-
date_the_model):
213.         z = classic_odds[0] + fuzzy_odds[0]
214.         for i in range(1, int(len(classic_odds))):
215.             z += (classic_odds[i] + fuzzy_odds[i]) * X[j][i - 1]
216.         dotY_right.append(z)
217.
218.
219.     list_to_answer[hh][5] = dotY_right
220.     #####
221.     ###нижние данные####
222.     #####
223.     dotY_left = []
224.
225.
226.     for j in range (the_number_of_rows_required_to_vali-
date_the_model):
227.         z = classic_odds[0] - fuzzy_odds[0]
228.         for i in range(1, int(len(classic_odds))):
229.             z += (classic_odds[i] - fuzzy_odds[i]) * X[j][i - 1]

```

```

230.             dotY_left.append(z)
231.
232.
233.             list_to_answer[hh][6] = dotY_left
234.
235.
236.             for i in range(the_number_of_rows_required_to_vali-
date_the_model):
237.                 if ((realY[i] >= dotY_left[i]) and (realY[i] <=
dotY_right[i])):
238.                     test = 1
239.                 else:
240.                     test = 0
241.                     break
242.                 if test == 1:
243.                     list_for_the_final_answer.append(list_to_answer[hh])
244.
245.
246.
247.
248.             name_export_file = worksheet_name + '.xlsx'
249.             writer = pd.ExcelWriter(name_export_file,engine = 'xlsxwriter')
250.             for i in range(the_number_of_parts_into_which_the_confir-
dence_level_is_split - 1):
251.
252.                 # так как мне захотелось сделать лист листов то в следствии
от него пришлось избавляться
253.                 old_list = list_for_the_final_answer[i + 1][0:4]
254.                 new_list = []
255.
256.
257.
258.                 converter_of_multidimensional_lists_to_oneDimen-
sional(old_list)
259.                 list_with_data = pd.DataFrame({'года': date_list_generator(),
260.                                                 'верхняя часть интервала':
list_for_the_final_answer[i + 1][5],
261.                                                 'центральная часть интервала':
list_for_the_final_answer[i + 1][4],
262.                                                 'нижняя часть интервала':
list_for_the_final_answer[i + 1][6],
263.                                                 'Реальные дан-
ные':list_for_the_final_answer[i + 1][7]})
264.
265.                 list_with_names = pd.DataFrame({'Названия показателей': coef-
ficient_name_generator(list(raw_dataframe)),
266.                                                 'Условные обозначения': coefficient_gen-
erator(list(raw_dataframe)),
267.                                                 'коэффициенты': new_list})
268.
269.                 list_with_data.to_excel(writer, sheet_name = "данные при h" +
str(i + 1))
270.                 list_with_names.to_excel(writer, sheet_name = "коэффициенты
при h" + str(i + 1))

```

```
271.     writer.save ()
272.     writer.close ()
```

Волгоградская область

Для первой из рассматриваемых областей полностью продемонстрируем алгоритм работы с программным комплексом, а в остальных случаях оставим только конечный результат.

Подготовка данных.

Сохраним таблицу с данными в файле Excel с названием tsst2.xlsx. Назовём лист с данными Волгоградская область (остальные области будут добавляться в этот же файл, а листы, на которых они будут находиться будут называться соответственно). Вместо X1...X3 кратко переименуем показатели.

	A	B	C	D	E
1	<i>Плотность дорог</i>	<i>ВРП</i>	<i>Индексы</i>	<i>У</i>	
2	72	10699	122,7	48,3	
3	74	11523,5	116,5	39,4	
4	74	11844,4	109,4	30	
5	75	17739,6	121,6	27,2	
6	76	27680,4	143,2	27,4	
7	77	30513,2	153,8	24,2	
8	78	38617,7	81,1	23,7	
9	78	47927,2	149,1	24,6	
10	78	57931,8	89,9	25,4	
11	79	77811,8	93,5	22,7	
12	109	95952,1	157,4	22,3	
13	108	126901,8	128,7	19,8	
14	108	160024,2	107,3	21,3	
15	108	145453,6	102,3	14,7	
16	109	166028,8	141,6	15	
17	119	195463,7	120,4	18,6	
18	134	220755,1	98,3	21,2	
19	140	235814,1	96,9	21,9	
20	140	279101,4	114,5	20,3	
21	141	290186	117,9	17,9	
22	142	292565,7	97,4	15,4	
23	143	305100	104,1	14,9	
24					

Рисунок 7. Формат данных пригодных для импорта в программный комплекс.

В 75 и 76 строках, необходимо вручную указать начало и конец рассматриваемого периода.

В итоге, для построения модели в программном комплексе, необходимо сделать всего одно изменение в исходном коде и при необходимости обнулить отрицательные значения (в данном примере такая необходимость возникает при каждом возникновении, но если данный программный комплекс использовать с другими данными где объясняемая переменная может принимать отрицательные значения тогда интервал следует использовать без изменений) в полученных файлах.

Интерпретация результата

Решение задачи линейного программирования для Волгоградской области представлено в таблице 1.

Таблица 1. Значения нечетких коэффициентов модели для Волгоградской области

h	a_0	a_1	a_2	a_3	r_0	r_1	r_2	r_3	f
0,4	7,803	0,2992	-0,00017	0,0806	0	0	0	0,148508026	302,4811

Уравнение нечеткой регрессии при $h = 0,4$ имеет вид

$$Y = 7,80331 - 0,299235X_1 - 0,00017X_2 + \langle -0,06376101; 0,080601; 0,23356301 \rangle X_3$$

Обратим внимание, что нечетким является только коэффициент при неизвестной X_3 . Учитывая, что $Y = \langle Y_d, Y_m, Y_u \rangle$, получаем график нечеткой линейной регрессии (Рисунок 8).

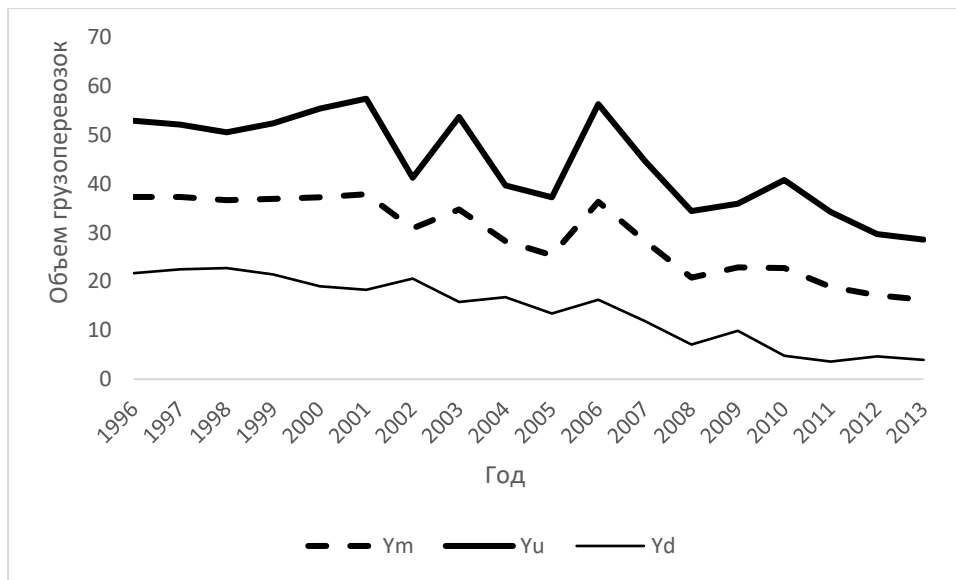


Рисунок 8. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в Волгоградской области организациями всех видов деятельности (Y , млн.т.), $h = 0,4$.

Оценки результатов с помощью контролирующей выборки приведен в таблице 2.

Таблица 2. Оценка результатов исследования Волгоградской области с помощью контролирующей выборки для модели с показателем нечеткости 0,4

Год	Ограничение слева	Ограничение справа	Y контролирующее	Принадлежность Y интервалу
2014	24,68675605	-4,463248007	20,3	+
2015	23,75419341	-6,261400288	17,9	+
2016	19,37540499	-5,421192352	15,4	+
2017	18,87534772	-7,626970385	14,9	+

Ростовская область

Решение задачи линейного программирования для Ростовской области представлено в таблице 3.

Таблица 3. Значения нечетких коэффициентов модели для Ростовской области

h	a_0	a_1	a_2	a_3	r_0	r_1	r_2	r_3	f

0,4	142,47	-0,27852	-0,00013	-0,0223	18,7793049	0,144052487	0	0	623,63108
-----	--------	----------	----------	---------	------------	-------------	---	---	-----------

Уравнение нечеткой регрессии при $h = 0,4$ имеет вид

$$Y = \langle 123,691; 142,47; 161,249 \rangle - \langle 0,42252; 0,27852; 0,13452 \rangle X_1 - 0,00013 X_2 - 0,0223 X_3$$

График нечеткой линейной регрессии имеют вид (Рисунок 9).

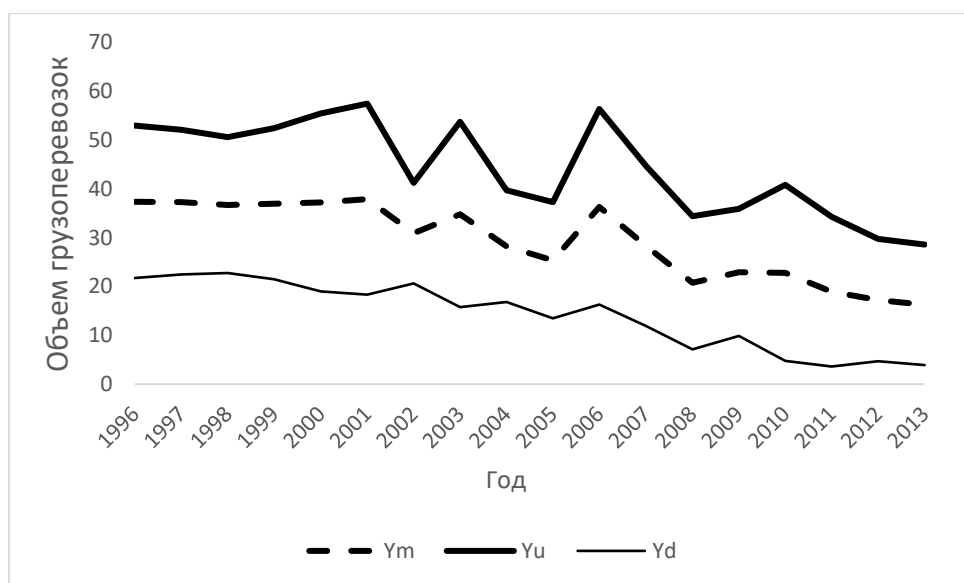


Рисунок 9. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в Ростовской области организациями всех видов деятельности (Y , млн.т.), $h = 0,4$.

Оценки результатов с помощью контролирующей выборки приведен в таблице 4.

Таблица 4. Оценка результатов исследования Ростовской области с помощью контролирующей выборки для модели с показателем нечеткости 0,4

Год	Ограничение слева	Ограничение справа	Y контролирующее	Принадлежность Y интервалу
2014	91,76603379	20,69986928	60,3	+
2015	85,66224535	-27,09176269	60	+
2016	83,0084226	30,03369041	60,4	+

2017	80,42320536	-32,90701263	54,6	+
------	-------------	--------------	------	---

Краснодарский край

Решение задачи линейного программирования для Краснодарского края (представлено в таблице 5).

Таблица 5. Значения нечетких коэффициентов модели для Краснодарского края

h	a_0	a_1	a_2	a_3	r_0	r_1	r_2	r_3	f
0,4	26,61	0,0971	-0,000199	0,8921	0	0	0	0,405293983	819,301787

Уравнение нечеткой регрессии при $h = 0,4$ имеет вид

$$Y = 26,6072 + 0,09712 X_1 - 0,00019 X_2 + \langle 0,4871; 0,8921; 1,2971 \rangle X_3 .$$

График нечеткой линейной регрессии имеют вид (Рисунок 10).

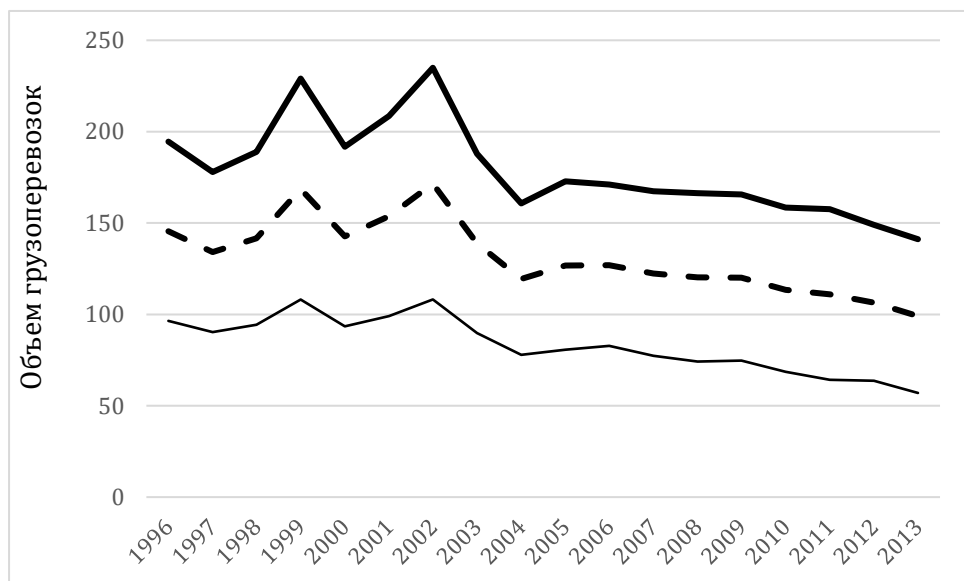


Рисунок 10. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в Краснодарского края организациями всех видов деятельности (Y , млн.т.), $h = 0,4$. Оценки результатов с помощью контролирующей выборки приведен в таблице 6.

Таблица 6. Оценка результатов исследования Краснодарского края с помощью контролирующей выборки для модели с показателем нечеткости 0,4

Год	Ограничение слева	Ограничение справа	Y контролирующее	Принадлежность Y интервалу
2014	172,8605366	67,15986584	88,3	+
2015	152,3822849	57,05714003	84,6	+
2016	119,4483523	43,73943627	78,8	+
2017	133,52843	45,25540042	69,4	+

Астраханская область

Решение задачи линейного программирования для Астраханской области представлено в таблице 7.

Таблица 7. Значения нечетких коэффициентов модели для Астраханской области

h	a_0	a_1	a_2	a_3	r_0	r_1	r_2	r_3	f
0,4	24,509	0,27684	0.0000105312	0,001000945	2,438	0	0	0,014259	70,1504

Уравнение нечеткой регрессии при $h = 0,4$ имеет вид

$$Y = \langle 22,07; 24,508; 26,946 \rangle - 0,2768 X_1 + \langle -0,01325; 0,001; 0,01526 \rangle X_3$$

График нечеткой линейной регрессии имеют вид (Рисунок 11).

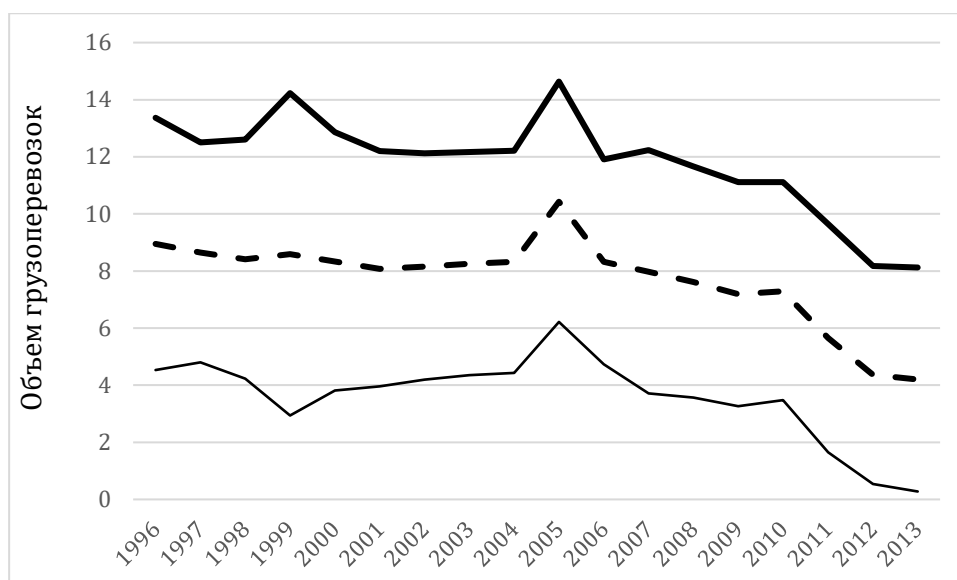


Рисунок 11. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в Астраханской области организациями всех видов деятельности (Y , млн.т.), $h = 0,4$.

Оценки результатов с помощью контролирующей выборки приведен в таблице 8.

Таблица 8. Оценка результатов исследования Астраханской области с помощью контролирующей выборки для модели с показателем нечеткости 0,4

Год	Ограничение слева	Ограничение справа	Y контролирующее	Принадлежность Y интервалу
2014	8,081147545	-0,824590181	6,3	+
2015	7,591780754	-0,421316533	4,9	+
2016	7,514754546	-0,030633373	4,3	+
2017	8,428326387	0,768863012	3,5	+

Республика Адыгея

Решение задачи линейного программирования для республики Адыгея представлено в таблице 9.

Таблица 9. Значения нечетких коэффициентов модели для республики
Адыгея

h	a_0	a_1	a_2	a_3	r_0	r_1	r_2	r_3	f
0,4	24,5083	0,002566	-0.0000207209	0,0229	31,5905	0	0	0	537,0387

Уравнение нечеткой регрессии при $h = 0,4$ имеет вид

$$Y = \langle -7,082; 24,508; 56,098 \rangle + 0,00256 X_1 + 0,0229 X_3 .$$

График нечеткой линейной регрессии имеют вид (рисунок 12).

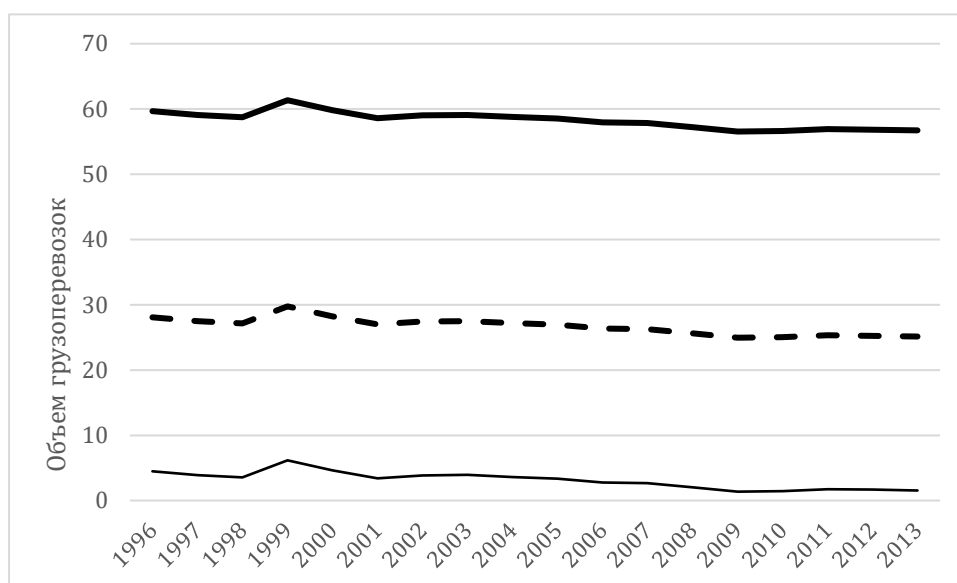


Рисунок 12. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в республике Адыгея организациями всех видов деятельности (Y , млн.т.), $h = 0,4$.

Обратим внимание на то, что коэффициент при переменной X_3 . Равен нулю, а нечетким является только свободный член.

Так как нижняя граница построенного графика совпадает с осью абсцисс (в силу невозможности принятия данным показателем отрицательных значений). Оценки результатов с помощью контролирующей выборки не имеют значения так как нижняя граница построенного графика совпадает с осью абсцисс.

Построим аналогичную модель для значения порога надежности $h = 0,2$ (рисунок 13).

Уравнение нечеткой регрессии при $h = 0,2$ имеет вид

$$Y = \langle 0,816; 24,508; 48,2 \rangle + 0,00256 X_1 + 0,0229 X_3 .$$

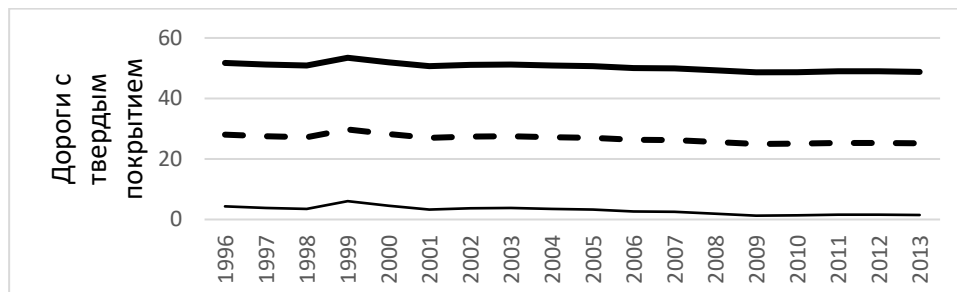


Рисунок 13. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в республике Адыгея организациями всех видов деятельности (Y , млн.т.), $h = 0,2$.

Как видно, в этом случае модель является более адекватной, оценки результатов с помощью контролирующей выборки приведены в таблице 10

Таблица 10. Оценка результатов исследования республике Адыгея с помощью контролирующей выборки для модели с показателем нечеткости 0,2

Год	Ограничение слева	Ограничение справа	Y контролирующее	Принадлежность Y интервалу
2014	24,68675605	1,993166	20,3	+
2015	23,75419341	0,790479	17,9	+
2016	19,37540499	0,472263	15,4	+
2017	18,87534772	0,127489	14,9	+

Республика Калмыкия

Решение задачи линейного программирования для республики Калмыкия представлено в таблице 11.

Таблица 11. Значения нечетких коэффициентов модели для республики Калмыкия

h	a_0	a_1	a_2	a_3	r_0	r_1	r_2	r_3	f
0,4	14,56523201	-0,1430164	-0.0000160283	-0,042756	0	0	0.00000 0450151	0,0168	32,199

Уравнение нечеткой регрессии при $h = 0,4$ имеет вид

$$Y = 14,565 - 0,14301 X_1 + \langle -0,05955; -0,04275; -0,02595 \rangle X_3.$$

График нечеткой линейной регрессии имеют вид (Рисунок 14).

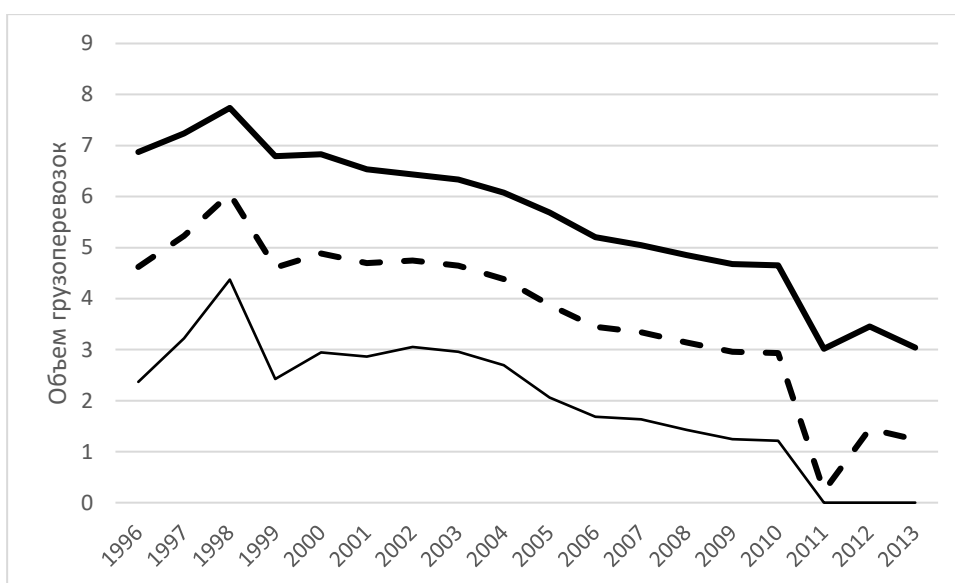


Рисунок 14. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в республике Калмыкия организациями всех видов деятельности (Y , млн.т.), $h = 0,4$.

Оценки результатов с помощью контролирующей выборки приведен в таблице 12.

Таблица 12. Оценка результатов исследования республики Калмыкия с помощью контролирующей выборки для модели с показателем нечеткости 0,4

Год	Ограничение слева	Ограничение справа	Y контролирующее	Принадлежность Y интервалу
2014	2,442074624	0	1,3	+
2015	2,353488854	0	1,2	+

2016	1,967329339	0	0,6	+
2017	1,215316336	0	0,5	+

Обратим внимание на то, что, как и в предыдущем случае, коэффициент при переменной X_3 равен нулю, а нечетким является только свободный член.

Так как нижняя граница построенного графика совпадает с осью абсцисс (в силу невозможности принятия данным показателем отрицательных значений), несмотря на то что в полученный интервал входят контролируемые показатели, данная модель не применима для республики Калмыкия. Как и в предыдущем случае, построим аналогичную модель для $h = 0,2$.

Уравнение нечеткой регрессии при $h = 0,2$ имеет вид

$$Y = 14,565 - 0,14301 X_1 + \langle -0,05535; -0,04275; -0,03015 \rangle X_3 .$$

График нечеткой линейной регрессии имеют вид (Рисунок 15).

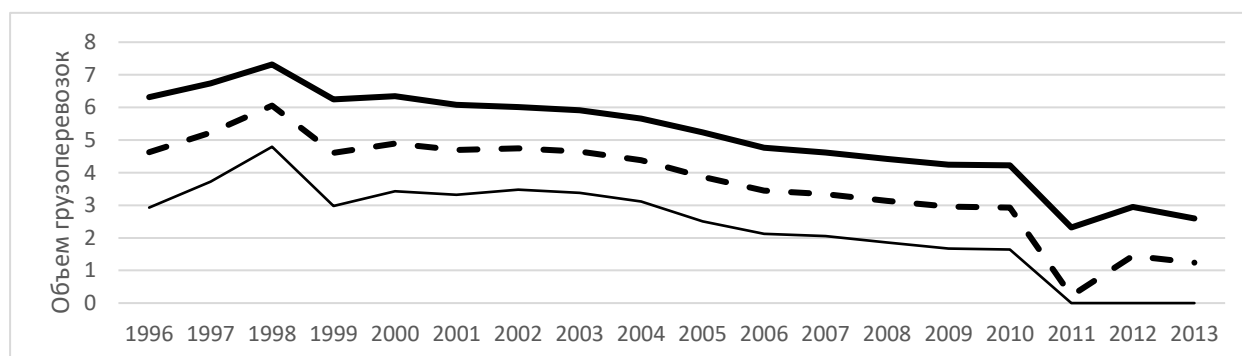


Рисунок 15. График нечеткой линейной регрессии для показателя перевозки грузов автомобильным транспортом в республике Калмыкия организациями всех видов деятельности (Y , млн.т.), $h = 0,2$.

Очевидно, данная модель не является адекватной и для случая $h = 0,2$ для выбранных показателей данного региона так как нижняя граница построенного графика совпадает с осью абсцисс. Необходимо проанализировать факторы, входящие в модель.

ВЫВОДЫ

Проверка построенных для Волгоградской, Ростовской и Астраханской областей, а также для Краснодарского края моделей подтвердила их адекватность, в том числе и с помощью контрольных выборок за последующие 4 года. Модель для республики Адыгея модель может применяться при малых значениях порога надежности, Построенная по данным показателей республики Калмыкия модель не является адекватной, необходим более глубокий анализ факторов, влияющих на объем грузоперевозок в этом регионе.

Стоит отметить, что если основываться на построенную модель, то поставленная правительством Ростовской области цель, заключающаяся в увеличении объема грузовых перевозок автомобильным транспортом в 1.6-1.8 раза, кажется труднодостижимой без дополнительных усилий по стимулированию этой отрасли [20].

Была создана программа на языке Python, позволяющая анализировать показатели различных отраслей с визуальным представлением данных в MS Excel.

ЗАКЛЮЧЕНИЕ

В работе проведен анализ объема грузовых автомобильных перевозок в шести областях ЮФО. Построенные с помощью нечеткой линейной регрессии модели позволили исследовать зависимость объема перевозки грузов автомобильным транспортом в этих регионах от выбранных факторов в пяти регионах из шести.

Представляется, что данный метод целесообразно использовать для анализа и прогнозирования значений различных экономических показателей. Созданная программа на языке Python позволяет анализировать показатели различных отраслей не только региональной экономики с визуальным представлением данных в MS Excel.

Также можно отметить, что Python обладает широкими возможностями для реализации самых сложных алгоритмов и моделей при анализе данных и экономико-математическом моделировании.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Стратегия развития транспортного комплекса Ростовской области до 2030 года. [Электронный ресурс]. – Режим доступа <http://www.mindortrans.donland.ru/Default.aspx?pageid=107384>
2. Стратегия социально-экономического развития Ростовской области на период до 2030 года. URL: <http://www.economy.gov.ru/mines/main>.
3. Стратегия развития транспортного комплекса Ростовской области до 2030 года. URL: http://www.don-land.ru/Data/Sites/1/media/antikorrupt_expert/2011/pro-ect_2011_1808_strateg.doc/copy.yandex.net.
4. Документация по языку программирования Python. [Электронный ресурс]. – Режим доступа <https://docs.python.org/>
5. Альбеков А.У. Терминально-логистическая инфраструктура транспортного комплекса Ростовской области // Национальные интересы: приоритеты и безопасность. 2011. №36. URL: <https://cyberleninka.ru/article/n/terminalno-logisticheskaya-infrastruktura-transportnogo-kompleksa-rostovskoy-oblasti> (дата обращения: 25.05.2020).
6. Прогноз социально-экономического развития Российской Федерации на 2011 год и плановый период 2012 и 2013 годов (разработан Минэкономразвития России). URL: <http://www.economy.gov.ru/mines/main>.
7. Сценарий развития транспорта России до 2030 года: отчет по НИР ФГУП «Научный центр по комплексным транспортным проблемам Минтранса России». М., 2008.
8. Информационно-аналитические материалы субъектов Южного федерального округа, Минтранса России, СКЖД-филиала ОАО «РЖД», НЦ КТП Минтранса России, ОАО «Гипрогор», Strategy Partners.
9. Прикладной регрессионный анализ [Электронный ресурс]. – Режим доступа http://www.machinelearning.ru/wiki/index.php?title=Регрессионная_модель

10. S.M. Chen. Forecasting enrollments based on fuzzy time series // Fuzzy Sets Systems, 1996, vol. 81, no. 3, pp. 311-319.
11. FuzzyR: Fuzzy Logic Toolkit for R. [Электронный ресурс]. – Режим доступа <https://cran.r-project.org/web/packages/FuzzyR/index.html>
12. Fuzzy Inference System Modeling. [Электронный ресурс]. – Режим доступа <https://www.mathworks.com/products/fuzzy-logic.html>
13. Документация по библиотеке cvxopt. [Электронный ресурс]. – Режим доступа <http://cvxopt.org/documentation/>
14. Н. Tanaka, S. Uejima, K. Asai, Linear regression analysis with fuzzy model, IEEE Transactions on Systems, Man and Cybernetics.– 12 (6).– Pp. 903-907. –1982.
15. Документация по библиотеке pandas. [Электронный ресурс]. – Режим доступа <https://pandas.pydata.org/pandas-docs/stable/>
16. Волкова Е. С., Гисин В. Б. Нечеткая линейная регрессия в модели роста технологических знаний. Вестник Финансового университета. – № 5 (89). – С.97–104.–2015.
17. Документация по библиотеке numpy. [Электронный ресурс]. – Режим доступа <https://docs.scipy.org/doc/>
18. Регионы России. Социально-экономические показатели. Статистические сборники. – Москва. [Электронный ресурс]. – Режим доступа <https://www.gks.ru/folder/210/document/13204>
19. Концепция долгосрочного социально-экономического развития Российской Федерации на период до 2020 года: распоряжение Правительства РФ от 17.11.2008 № 1662-р.
20. Транспорт в России 2007. Стат. сб. М.: Росстат, 2007. 8. Федеральная целевая программа «Развитие транспортной системы России (2010 — 2015гг.)», подпрограмма «Автомобильные дороги» // Минтранс России. М., 2008.

ПРИЛОЖЕНИЯ

Приложение А. Статистические данные показателей развития автомобильных грузоперевозок в ЮФО.

Значения выбранных показателей 1996-2017 гг., Волгоградская область

год	X1	X2	X3	Y
1996	72	10699	122,7	48,3
1997	74	11523,5	116,5	39,4
1998	74	11844,4	109,4	30
1999	75	17739,6	121,6	27,2
2000	76	27680,4	143,2	27,4
2001	77	30513,2	153,8	24,2
2002	78	38617,7	81,1	23,7
2003	78	47927,2	149,1	24,6
2004	78	57931,8	89,9	25,4
2005	79	77811,8	93,5	22,7
2006	109	95952,1	157,4	22,3
2007	108	126901,8	128,7	19,8
2008	108	160024,2	107,3	21,3
2009	108	145453,6	102,3	14,7
2010	109	166028,8	141,6	15
2011	119	195463,7	120,4	18,6
2012	134	220755,1	98,3	21,2
2013	140	235814,1	96,9	21,9
2014	140	279101,4	114,5	20,3
2015	141	290186	117,9	17,9
2016	142	292565,7	97,4	15,4
2017	143	305100	104,1	14,9

Значения выбранных показателей 1996-2017 гг., Ростовская область

год	X1	X2	X3	Y
1996	103	7088,7	131,1	128,4
1997	105	8062,9	102,9	101,2
1998	104	9163,7	76,9	90,3
1999	109	15672,2	112	92
2000	114	20003,8	170,3	83,1
2001	118	26817,8	129,1	98,7
2002	120	31941,7	115,7	99,4
2003	123	39225,1	137	119
2004	123	50843,4	113,7	120,7
2005	48	60906,4	111,5	113,6
2006	141	78328,3	105,3	66,8
2007	143	104603,1	103,8	64,7

2008	142	134137,4	113	61,9
2009	140	129626	110,7	60,1
2010	139	154127,9	109,3	58,9
2011	139	179470,3	108,8	57,7
2012	202	198129,2	104,4	60,1
2013	258	215923,2	104	61,4
2014	260	237465,8	106,8	60,3
2015	261	280522	113,1	60
2016	262	300186,2	106,7	60,4
2017	263	318800	103,6	54,6

Значения выбранных показателей 1996-2017 гг., Краснодарский край

год	X1	X2	X3	Y
1996	132	9036,5	120,9	163
1997	132	8985	108,1	124,6
1998	134	10817,1	116,7	113,2
1999	135	20919,8	149,1	134,6
2000	137	26713,9	121,1	130,2
2001	140	34912,4	135,1	129,7
2002	142	42477,6	156,5	133,5
2003	143	48613,5	121,1	168,3
2004	143	61455,6	102,1	113
2005	136	73146,3	113,8	99,1
2006	219	91368,7	109	119,2
2007	225	125700,4	110,9	144,6
2008	239	155103,6	113,6	110,5
2009	272	165555,1	112,2	104,2
2010	280	196914,3	110,9	112,9
2011	296	236750,6	115,2	112,3
2012	419	274995,7	105,2	121,6
2013	428	309837,7	103,8	119,7
2014	438	328771,2	130,4	88,3
2015	447	352601,2	117,6	84,6
2016	454	363731,3	93,4	78,8
2017	463	398400	108,9	69,4

Значения выбранных показателей 1996-2017 гг., Астраханская область

год	X1	X2	X3	Y
1996	57	7918,6	138,6	11,6
1997	58	9149,3	99,2	7,9
1998	59	10914,7	122,8	5,9
1999	59	17584,6	224,6	9,1
2000	60	27815,3	146,3	8,6
2001	61	32037,2	118	7,5
2002	61	40786,8	107	9

2003	61	50508,2	102,9	10,6
2004	61	56725,9	101,9	10
2005	54	70393,9	124,2	7,9
2006	62	85172,3	80,4	7,7
2007	64	99999,4	127,7	7,7
2008	67	146391	112,7	6,9
2009	68	133018,5	104,2	5,8
2010	68	143388,8	96,8	5
2011	75	170504,7	109,3	5,8
2012	81	206677,1	96,4	5,3
2013	84	269821,7	104,1	5,4
2014	87	290822,2	141,3	6,3
2015	88	315996,9	110	4,9
2016	88	332447,4	93,6	4,3
2017	88	413400	97,6	3,5

Значения выбранных показателей 1996-2017 гг., республика Адыгея

год	X1	X2	X3	Y
1996	195	5369,9	137,8	14,1
1997	199	6257	113	14,2
1998	199	7522,3	100	8,2
1999	202	10909,6	216,2	10,8
2000	202	12315,1	150	11,4
2001	203	14858,6	100	11,6
2002	207	17707,5	120	11,2
2003	207	22078,8	127,8	9,9
2004	208	28084,8	120	13
2005	204	38389,5	118,4	14,7
2006	203	47373,4	100,3	45,3
2007	203	66366,2	113,1	17,7
2008	204	82378,2	100	9,5
2009	206	94436,5	81,8	6
2010	207	107297,5	96,6	6,3
2011	556	128749,3	90,1	6,4
2012	562	147262,9	102,8	6,3
2013	564	159096,3	108,1	6,8
2014	564	168880,1	136,5	7,5
2015	563	187211,1	100,7	6,8
2016	564	201918,1	100	7,4
2017	570	219300	100	6,3

Значения выбранных показателей 1996-2017 гг., Республика Калмыкия

год	X1	X2	X3	Y
1996	57	7918,6	138,6	11,6
1997	58	9149,3	99,2	7,9

1998	59	10914,7	122,8	5,9
1999	59	17584,6	224,6	9,1
2000	60	27815,3	146,3	8,6
2001	61	32037,2	118	7,5
2002	61	40786,8	107	9
2003	61	50508,2	102,9	10,6
2004	61	56725,9	101,9	10
2005	54	70393,9	124,2	7,9
2006	62	85172,3	80,4	7,7
2007	64	99999,4	127,7	7,7
2008	67	146391	112,7	6,9
2009	68	133018,5	104,2	5,8
2010	68	143388,8	96,8	5
2011	75	170504,7	109,3	5,8
2012	81	206677,1	96,4	5,3
2013	84	269821,7	104,1	5,4
2014	87	290822,2	141,3	6,3
2015	88	315996,9	110	4,9
2016	88	332447,4	93,6	4,3
2017	88	413400	97,6	3,5