

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и
стохастического анализа

**ИССЛЕДОВАНИЕ КАЧЕСТВА МОДЕЛЕЙ ВРЕМЕННЫХ
РЯДОВ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 412 группы
направления 01.03.02 — Прикладная математика и информатика
механико-математического факультета
Плотникова Георгия Андреевича

Научный руководитель

к. ф.-м. н.

Н. Ю. Агафонова

Заведующий кафедрой

д. ф.-м. н.

С. П. Сидоров

Саратов 2017

ВВЕДЕНИЕ

Актуальность работы. Для широкого круга задач, связанных с экономической деятельностью эффективно применение эконометрических моделей, построение временных рядов. В большинстве случаев основной целью является построение прогноза.

Данные, собранные на протяжении некоторого временного периода, обычно показывают тренд, или основное направление движения показателей, характерное для разного периода времени.

Кроме того, с большой частотой, полученные наблюдения имеют зависимость от своих предыдущих значений, т.е. автокоррелируют. Следовательно, полученные данные не могут быть рассмотрены в качестве случайной выборки.

Построение таких моделей должно происходить крайне аккуратно, внимательно и с особой точностью.

Кроме того, текущие значения процесса могут зависеть от своих предыдущих показателей. Такой процесс называется авторегрессией. Точность любого прогноза зависит от выбранной модели, количества и качества данных и других факторов.

Тема была обусловлена актуальностью задачи построение качественного прогноза, т.е. выбора адекватной модели временного ряда - “Исследование качества модели авторегрессии”.

Целью работы является исследование качества основных моделей, используемых в анализе временных рядов, проведение тестов для установления стационарности временного ряда, написание программного кода, реализующего данный функционал.

Объект исследования - модели авторегрессии.

Предмет исследования - качественные характеристики различных моделей авторегрессии. Обоснование выбора той или иной модели. Тестирование временного ряда на стационарность

Для достижения поставленных целей в работе необходимо решить следующие **задачи**:

- определить основные характеристики;

- рассмотреть модели авторегрессии;
- проанализировать процесс идентификации модели;

Теоретико-методологической основой исследования являются труды, описывающие анализ временных рядов, моделей авторегрессии и тестирования этих моделей (Бокс. Дж., Магнус. Я.Р., Андерсон Т.)

Практическая значимость – разработка программного продукта, позволяющего выполнить моделирование, тестирование и может быть использован для решения как учебных, так и задач, основанных на реальных данных.

1 Основное содержание работы

Выпускная квалификационная работа состоит из введения, основных определений, двух теоретических и одной практической главы, заключения, списка использованных источников и приложений.

Введение содержит основные положения: актуальность темы исследования; цель, объект, предмет, задачи исследования; практическую значимость исследования. Данная работа состоит из 4 разделов:

В первом разделе работы **“Основные понятия и определения”** приведены основные определения и понятия, используемые при анализе временных рядов, авторегрессии и в эконометрическом анализе в целом, такие как понятия детерминированности, стационарности, понятие стохастического в широком и узком смысле процесса. Рассмотрена модель парной регрессии и метод наименьших квадратов. Приведен пример возможной реализации стохастического процесса.

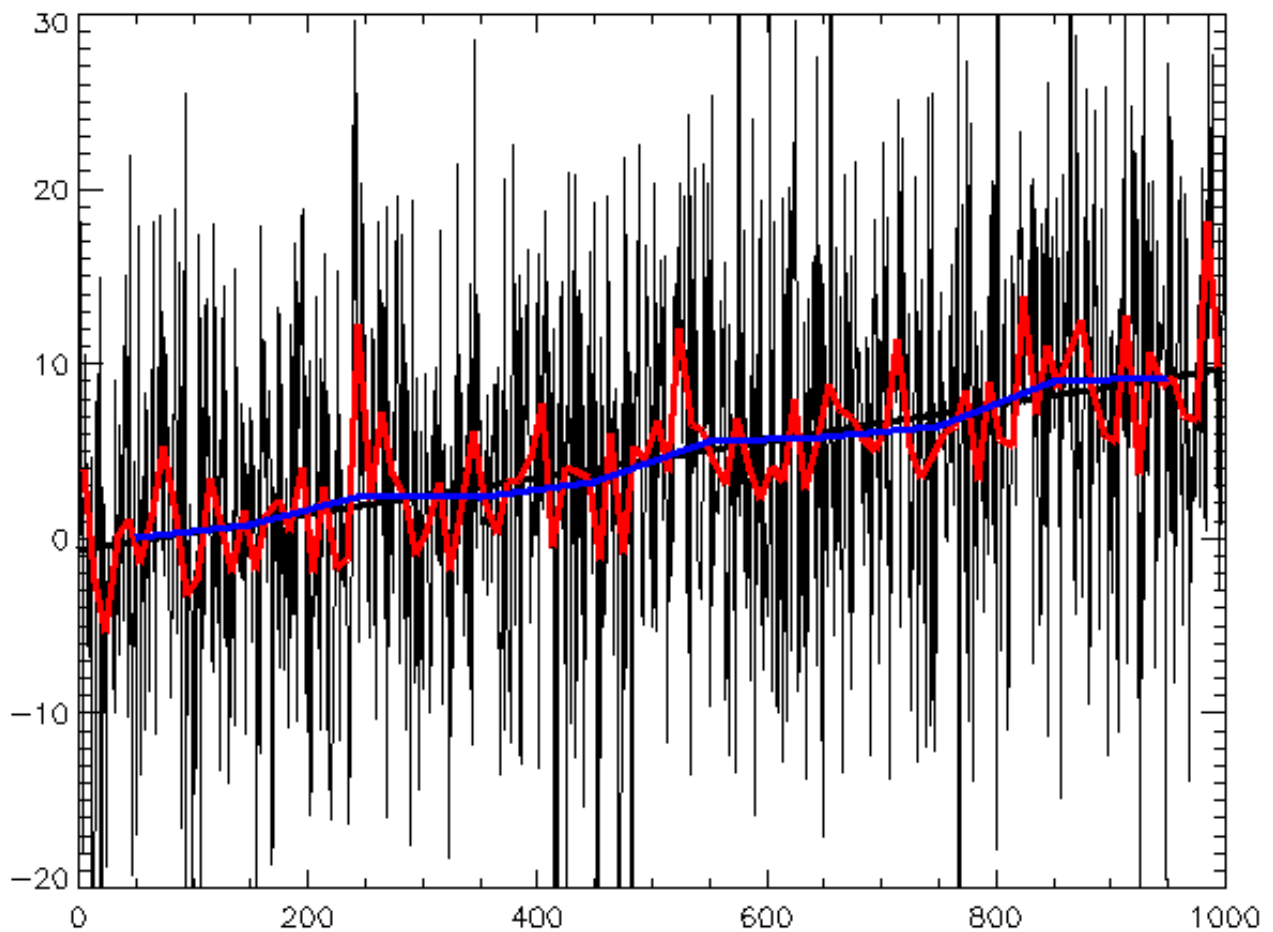


Рисунок 1.1: Пример временного ряда [5]

Во втором разделе работы “**Стационарные стохастические модели**” раскрывается смысл понятия стационарных стохастических моделей, рассматриваются основные модели временных рядов - модель авторегрессии $AR(k)$, модель скользящего среднего $MA(m)$, модель авторегрессии скользящего среднего $ARMA(k, m)$. Исследуются их характеристики, оценки параметров, условия их автокорреляции.

Подробным образом исследуется переход от модели $ARMA$ к моделям AR и MA , возможные ошибки и несостыковки, варианты их избежания и корректировки. Обозначены зависимости коэффициентов автокорреляции моделей AR и MA . Так же сделаны некоторые замечания относительно стационарности процесса.

Кроме того изучается процесс идентификации модели $ARMA$, его составные этапы и алгоритм. Проводится подбор наилучшей модели на основе схожести изучаемого процесса и теоретической модели по оценке поведения их автокорреляционных функций, возможные реализации такого процесса. Рассмотрены возможные свойства ряда ошибки такого процесса, схожесть с “белым шумом”. Приведены требования для значений дисперсии процесса и соответствующие формулы для подсчета выборочных коэффициентов.

В третьем разделе **Тестирование временных рядов на стационарность** рассматриваются основные критерии тестирования временных рядов.

Тесты разделяются на параметрические и непараметрические. Параметрические тесты проводятся при строгих предположениях о законе распределения и параметрах временного ряда. К этим тестам относятся тест Стьюдента, который рассчитывается на основе t -критерия. Временной ряд разделяется на части и на основе выведенных в этом разделе формул подсчитывается его значение. Так же здесь рассмотрен критерий Фишера и критерий Кокрена, разработанный на основе критерия Фишера, с большим количеством разбиений временного ряда на отрезки. Более мощным параметрическим критерием является критерий Бартлетта, но так же он является и более чувствительным к отклонениям от нормального распределения. Плюсом параметрических тестов является высокая точность. К недостаткам относится чувствительность к характеристикам выбранных параметров и они являются крайне ограниченными в применении из-за своих предположений отно-

сительности нормальности закона распределения временного ряда и сложности вычислений.

Другим видом тестов являются непараметрические тесты, которые более удобны в применении из-за отсутствия требований на закон распределения временного ряда и большей простотой вычислений. Вместо критерия Стьюдента используется непараметрический критерий Манна-Уитни или u^* -критерий. Он слабее критерия Стьюдента, в случае нормального распределения, но предпочтительнее при распределении, отличном от нормального. Используется для проверки идентичности распределения двух совокупностей. Далее рассматривается тест Сиджела-Тьюки, который является своеобразной заменой параметрического критерия Фишера, который основан на сопоставлении рангов элементов двух совокупностей. Для проведения этого теста необходимо центрировать рассматриваемый временной ряд, далее его разделяют на две части, в первой части содержатся элементы первой централизованной совокупности, а во второй части второй централизованной совокупности, при этом элементы этих совокупностей помнят о своих совокупностях. Далее по специальному правилу ранжирования элементы соединяют в одну таблицу. Последним непараметрическим критерием является критерий Вальда-Воловитца, который основан на анализе закономерностей серий.

В четвертом и последнем разделе **Построение программного продукта** приведен процесс разработки и использования программного продукта для тестирования временных рядов.

Рассмотрены основные требования к функционалу и интерфейсу приложения.

Для разработки был выбран язык Java из-за простоты и понятности написания кода, огромного количества стандартных библиотек, предназначенных для различных задач и целей. Одной из такой библиотек для статистиков - Apache Common Math [8] - было решено воспользоваться при разработке данного приложения. В ней содержится готовый и оптимизированный функционал для разработки статистических приложений и тестов. Так, в данной библиотеке предоставлены классы для Критерия Стьюдента и Манна-Уитни, критерия Колмогорова для проверки нормальности распределения.

Любое приложение должно быть интуитивно понятно для пользователя. Для разрабатываемого приложения был выбран консольный интерфейс, принимающий ряд интуитивно понятных команд.

```
Добро пожаловать в приложение для проверки временного ряда на стационарность.
=====
Пожалуйста, выберите путь к файлу с данными, в формате 'C:/Data/.../data.txt'. Либо выберите файл 'по умолчанию'
E:\диплом\Diploma
по умолчанию
=====
Значения вашего временного ряда
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0

ТЕСТЫ
Выберите тест(номер), по которому вы хотите проверить ваш временной ряд. Если вы не хотите тестировать ваш ряд далее, введите 'нет'
1. Критерий Стьюдента.
2. Критерий Фишера.
3. Критерий Кокрена.
4. Тест Манна-Уитни.
↓
Тестируем по критерию Стьюдента
Математическое ожидание: 5.2631578947368425
Дисперсия: 7.982456140350877
Значение критерия Стьюдента: 8.042839624832093 при альфа = 0.05
Число степеней свободы: 18
Критическое значение следует найти по таблице.
Тестируем дальше?
```

Рисунок 1.2: Интерфейс программного продукта

В приложении приведен программный код приложения и используемых классов.

ЗАКЛЮЧЕНИЕ

В ходе данной работы были рассмотрены различные модели временных рядов

1. Модель авторегрессии $AR(k)$,
2. Модель скользящего среднего $MM(m)$,
3. Модель авторегрессии скользящего среднего $ARMA(k, m)$.

критерии и тесты, устанавливающие стационарность временного ряда, с помощью которых определяется качество проверяемой модели

1. Параметрические критерии Стьюдента, Фишера, Кокрена и Бартлетта
2. Непараметрические критерии Манна-Уитни, Сиджела-Тьюки и Вальда-Вольфовитца.

На основе этих критериев написано консольное приложение, используемое для тестирования временных рядов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бокс. Дж. Анализ временных рядов. Прогноз и управление. Выпуск 1 [Текст] / Дж.Бокс, Г.Дженкинс. Москва: Мир, 1974. 408с.
2. Магнус Я.Р. Эконометрика. Начальный курс [Текст] / Я.Р. Магнус, П.К. Катышев, А.А. Пересецкий. Москва: Дело, 2004. 570с.
3. Андерсон Т. Статистический анализ временных рядов [Текст]/ Т.Андерсон. – Москва: Мир, 1976. 744с.
4. Курош. А Курс высшей алгебры. [Текст] / А.Г. Курош. Москва: Наука, 1968. 431с.
5. Временные ряды [Электронный ресурс]: свободная энциклоп. / электрон. энциклоп Режим доступа: https://ru.wikipedia.org/wiki/Временной_ряд
6. Волков И. Случайные процессы [Текст] / И.К.Волков, С.М.Зуев, Г.М. Цветкова. Москва: МГТУ им. Н.Э. Баумана, 1999, 448с.
7. Левин Б. Теоретические основы статистической радиотехники [Текст] / Б.Р. Левин. Москва: Радио и связь, 1989. 656с.
8. Commons Math: The Apache Commons Mathematics Library [Электронный ресурс]: электрон. репозиторий Режим доступа: <https://commons.apache.org/proper/commons-math/>

2 Листинг программы

```
import Test.Critical;
import org.apache.commons.math3.distribution.NormalDistribution;
import org.apache.commons.math3.stat.StatUtils;
import org.apache.commons.math3.stat.correlation.PearsonsCorrelation;
import org.apache.commons.math3.stat.inference.MannWhitneyUTest;
import org.apache.commons.math3.stat.inference.TestUtils;

import java.io.*;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;

/*
 * Copyright 2017 Georgii Plotnikov

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/
public class ConsoleApp {
    public static void main(String[] args) {
        System.out.println("Добро пожаловать в приложение для проверки временного ряда на стационарность.");
        System.out.println("=====");
        System.out.println();
        System.out.println("Пожалуйста, выберите путь к файлу с данными, в формате 'C:/Data/.../data.txt'. Либо выберите файл");
        String currentPath = new File("").getAbsolutePath();
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        File data = new File(currentPath + "\\src\\main\\resources\\gas.txt"); //файл по умолчанию
        String filename;
        try {
            if (!((filename = bufferedReader.readLine()).equalsIgnoreCase("по умолчанию"))) {
                data = new File(filename);
                bufferedReader.close();
            }
            FileReader reader = new FileReader(data);
            BufferedReader br = new BufferedReader(reader);

            ArrayList<Double> timeSeria = new ArrayList<>();
            String result;
            while ((result = br.readLine()) != null) {
                Double temp = Double.parseDouble(result);
                timeSeria.add(temp);
            }
            br.close();
            reader.close();
            System.out.println("=====");
            System.out.println("Значения вашего временного ряда");
            for (Double d : timeSeria) {
```

```

        System.out.println(d);
    }
    System.out.println();
    System.out.println("ТЕСТЫ");
    System.out.println("Выберите тест(номер), по которому вы хотите проверить ваш временной ряд. Если вы не хотите т
    System.out.println("1. Критерий Стьюдента.");
    System.out.println("2. Критерий Фишера.");
    System.out.println("3. Критерий Кокрена.");
    System.out.println("4. Тест Манна-Уитни.");
    BufferedReader testReader = new BufferedReader(new InputStreamReader(System.in));
    String test;
    double[] seriaArray = new double[timeSeria.size()];
    for (int i = 0; i < timeSeria.size(); ++i) {
        seriaArray[i] = timeSeria.get(i);
    }
    double[] timeArray = new double[seriaArray.length];
    for(int i=0; i < timeArray.length; ++i){
        timeArray[i] = i + 1;
    }
    while (((test = testReader.readLine().trim()) != "") && (!test.equalsIgnoreCase("Her"))) {
        switch (test) {
            case "1":
                studentTest(seriaArray, timeArray);
                break;
            case "2":
                System.out.println("Тестируем по критерию Фишера");
                System.out.println("Математическое ожидание: " + StatUtils.mean(seriaArray));
                System.out.println("Дисперсия: " + StatUtils.variance(seriaArray));
                System.out.println("Значение лямбда по критерию Колмогорова: " + TestUtils.kolmogorovSmirnovTest(new
                System.out.println("Значение критерия Фишера: " + (StatUtils.variance(seriaArray)) / StatUtils.vari
                System.out.println("Число степеней свободы: " + (seriaArray.length - 1));
                System.out.println("Число степеней свободы: " + (seriaArray.length - 1));
                System.out.println("Критическое значение следует найти по таблице.");
                break;
            case "3":
                System.out.println("Тестируем по критерию Кокрена");
                System.out.println("Математическое ожидание: " + StatUtils.mean(seriaArray));
                System.out.println("Дисперсия: " + StatUtils.variance(seriaArray));
                System.out.println("Значение критерия Кокрена: " + (StatUtils.variance(seriaArray)) / StatUtils.vari
                break;
            case "4":
                System.out.println("Проводим тест Манна-Уитни");
                System.out.println("Математическое ожидание: " + StatUtils.mean(seriaArray));
                System.out.println("Дисперсия: " + StatUtils.variance(seriaArray));
                //Здесь возвращается нормализованное значение статистики U критерия.
                System.out.println("Значение критерия Манна-Уитни: " + new MannWhitneyUTest().mannWhitneyUTest(seria

                break;
            default:
                System.out.println("Неизвестная команда");
        }
        System.out.println("Тестируем дальше?");
    }
    testReader.close();
    System.out.println("Ваши тесты окончены.");
} catch (FileNotFoundException ex) {
    System.out.println("Файл не найден");
} catch (IOException ex) {

```

```

        System.out.println("Ошибка ввода");
    }
}

private static void studentTest(double[] seriaArray, double[] timeArray) {
    HashMap<Integer, Double> critical = Critical.Student();
    System.out.println("Тестируем по критерию Стьюдента");
    System.out.println("Математическое ожидание: " + StatUtils.mean(seriaArray));
    System.out.println("Дисперсия: " + StatUtils.variance(seriaArray));
    double student = TestUtils.t(0.05, seriaArray);
    System.out.println("Значение критерия Стьюдента: " + student + " при альфа = " + 0.05);
    System.out.println("Число степеней свободы: " + (seriaArray.length - 1));
    double crit = checkCritical(seriaArray.length - 1);
    System.out.println("Критическое значение = " + checkCritical(seriaArray.length - 1));
    if (!(student < crit)) {
        System.out.println("Полученное значение критерия Стьюдента больше критического," +
            " следовательно гипотезу о стационарности отвергаем.");
        System.out.println("Пробуем получить стационарный ряд, вычисляя разности.");
        int count = 0;
        while (count <= 2) {
            double newArray[] = new double[seriaArray.length - 1 - count];
            for (int i = count + 1; i <= newArray.length; ++i) {
                double z = seriaArray[i] - seriaArray[i - 1];
                newArray[i - 1 - count] = z;
            }
            crit = checkCritical(newArray.length - 1);
            double curr = TestUtils.t(0.05, newArray);
            System.out.println("Полученное значение: " + curr + "; Критическое значение: " + crit);
            if (crit > curr) {
                System.out.println("По критерию Стьюдента преобразованный ряд стационарный.");
                try {
                    count = 0;
                    while(count <= 4) {
                        System.out.println("Коэффициент автокорреляции " + (count + 1) + "-го порядка: " + PearsenCorrel);
                        count++;
                    }
                    double[] coeff = AutoRegression.calculateARCoefficients(newArray, count, false);
                    for(int i = 0; i < coeff.length; ++i){
                        System.out.println("Коэффициент авторегрессии AR(" + (i + 1) + ") = " + coeff[i]);
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
                break;
            } else {
                count++;
                System.out.println("По критерию Стьюдента преобразованный ряд не является стационарным.");
            }
        }
    } else {
        System.out.println("По критерию Стьюдента данный ряд является стационарным.");
    }
}

private static double PearsenCorrelationCoeff(double[] seriaArray, int order){
    double result;
    double newArray[] = Arrays.copyOfRange(seriaArray, order + 1, seriaArray.length);
}

```

```

double sum = 0;

double[] tempArray = Arrays.copyOfRange(seriaArray, 0, seriaArray.length - (1 + order));

for(int i = 0; i < newArray.length; ++i){
    sum += tempArray[i] * newArray[i];
}

double seriaMean = StatUtils.mean(tempArray);
double seriaVariance = StatUtils.variance(tempArray);

double newArrMean = StatUtils.mean(newArray);
double newArrVariance = StatUtils.variance(newArray);

sum = sum / newArray.length;

result = (sum - (seriaMean * newArrMean))/((Math.sqrt(seriaVariance)*Math.sqrt(newArrVariance)));

return result;
}

private static double checkCritical(int freedomCount){
    if(freedomCount > 80 && freedomCount < 90){
        return (Double) Critical.Student().get(80);
    } else if(freedomCount > 90 && freedomCount < 100){
        return (Double) Critical.Student().get(90);
    }
    else return (Double) Critical.Student().get(freedomCount);
}
}

```