

Федеральное государственное бюджетное образовательное учреждение высшего  
профессионального образования  
«Уфимский государственный авиационный технический университет»

Отчет по лабораторной работе №5  
по дисциплине: «Статистическое моделирование»  
на тему: «Временные ряды».

Выполнил:  
Пахтусов Н. Г., ПРО-306  
Проверила:  
Рассадникова Е. Ю.

Уфа, 2015 г.

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>3</b>
<b>2</b>	<b>Работа с временными рядами в R</b>	<b>3</b>
<b>3</b>	<b>Данные</b>	<b>3</b>
<b>4</b>	<b>Начальный анализ</b>	<b>3</b>
4.1	Автокорреляция и периодичность . . . . .	3
4.2	Тренд . . . . .	4
4.3	Стационарность . . . . .	5
<b>5</b>	<b>Построение ARIMA модели</b>	<b>5</b>
5.1	Нахождения оптимальной модели . . . . .	6
5.2	Обработка полученных данных и прогноз . . . . .	8
<b>6</b>	<b>Вывод</b>	<b>10</b>

## 1 Постановка задачи

В данной работе нам необходимо для некоторых временных рядов построить модель ARIMA и провести её улучшение (если это необходимо).

## 2 Работа с временными рядами в R

В базовых возможностях R входят средства для представления и анализа временных рядов. Основным типом временных данных является «ts», который представляет собой временной ряд, состоящий из значений, разделенных одинаковыми интервалами времени. Временные ряды могут быть образованы и неравномерно отстоящими друг от друга значениями. В этом случае следует воспользоваться специальными типами данных — zoo и its, которые становятся доступными после загрузки пакетов с теми же именами.

Методы для анализа временных рядов и их моделирования включают ARIMA-модели, реализованные в функциях `arima()`, `AR()` и `VAR()`, структурные модели в `StructTS()`, функции автокорреляции и частной автокорреляции в `acf()` и `pacf()`, классическую декомпозицию временного ряда в `decompose()`, STL-декомпозицию в `stl()`, скользящее среднее и авторегрессивный фильтр в `filter()`.

## 3 Данные

В качестве исходных данных будем использовать данные о смертности в США за каждый месяц, начиная с января 1973 года и заканчивая декабрём 1978. Данные взяты с сайта <https://datamarket.com/>. Загрузим исходные данные:

```
> fram <- read.csv("/home/nick/Projects/R/Lab5/data.csv")
```

Запишем данные в переменную `data` данные из столбца с названием `data`, укажем, что начинаем с 1 месяца 1973 года, и что всего у нас 12 месяцев:

```
> data <- ts(fram$data,
+   start = c(1973, 1),
+   frequency = 12
+ )
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1973	9007	8106	8928	9137	10017	10826	11317	10744	9713	9938	9161	8927
1974	7750	6981	8038	8422	8714	9512	10120	9823	8743	9129	8710	8680
1975	8162	7306	8124	7870	9387	9556	10093	9620	8285	8433	8160	8034
1976	7717	7461	7776	7925	8634	8945	10078	9179	8037	8488	7874	8647
1977	7792	6957	7726	8106	8890	9299	10625	9302	8314	8850	8265	8796
1978	7836	6892	7791	8129	9115	9434	10484	9827	9110	9070	8633	9240

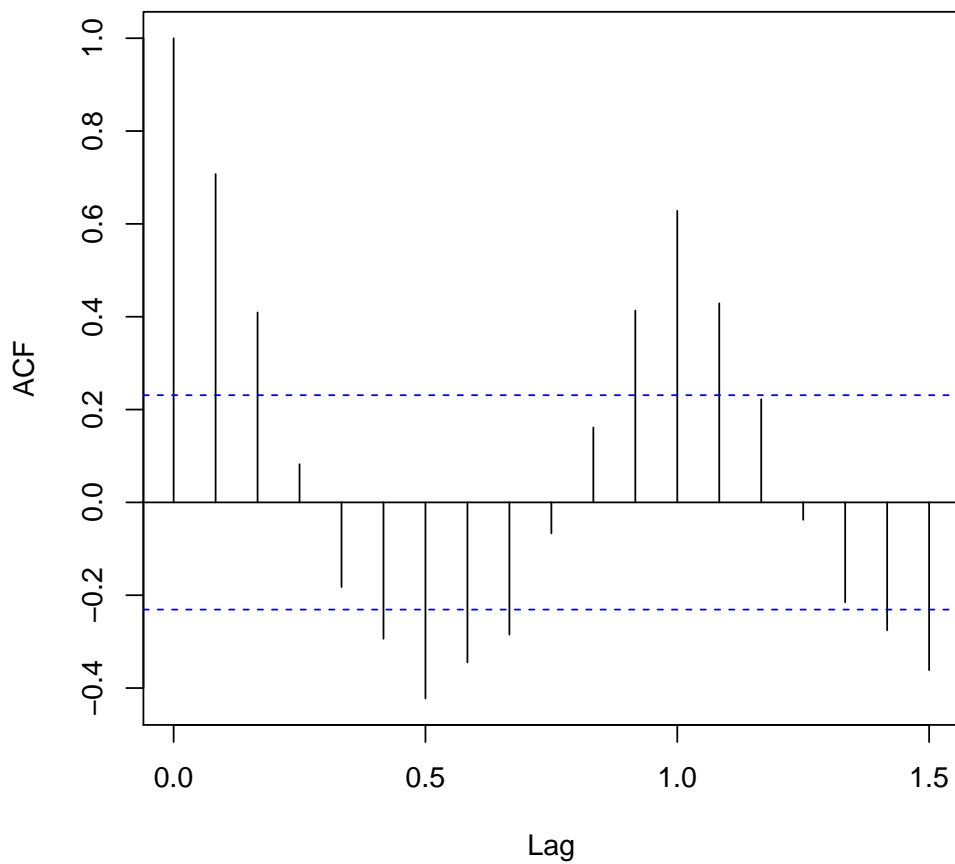
## 4 Начальный анализ

Теперь можно приступить к анализу.

### 4.1 Автокорреляция и периодичность

Применим функцию `acf` («auto-correlation function», ACF). Она выводит коэффициенты автокорреляции и рисует график автокорреляции. Чем больше палочек выходит за полосы, тем более значима периодичность.

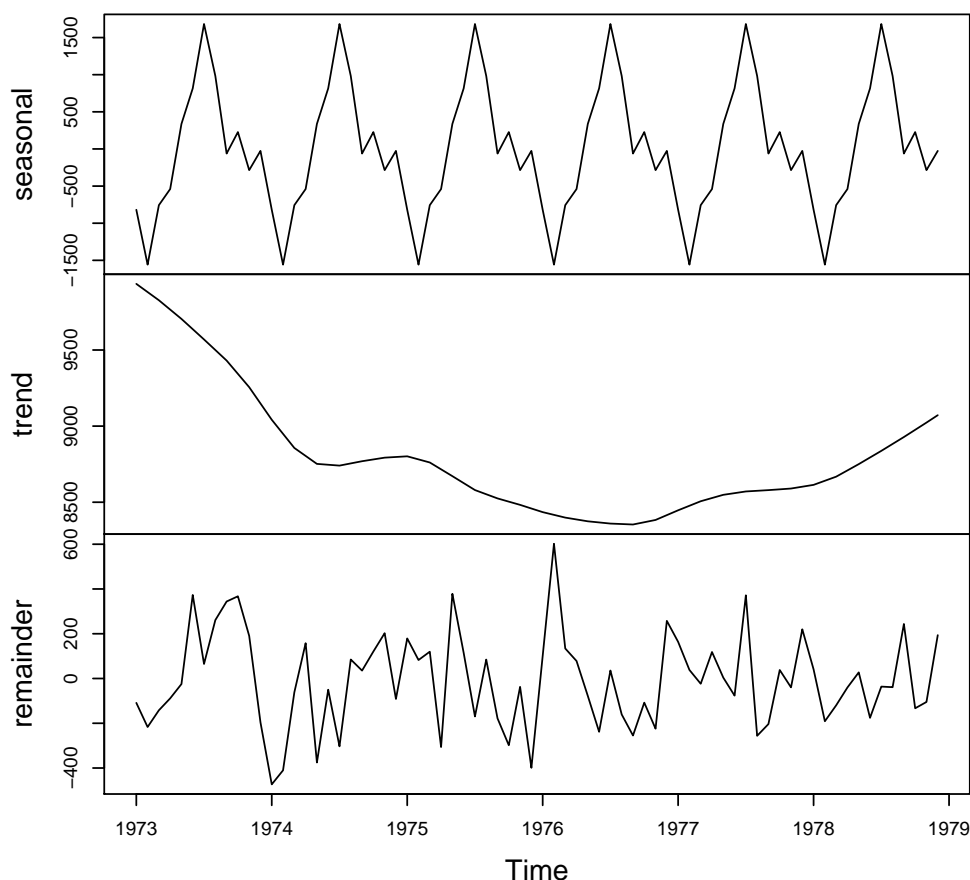
```
> acf(data, main="")
```



## 4.2 Тренд

То, что волнообразный график пиков как бы затухает, говорит о том, что в наших данных возможен тренд. Проверим это с помощью функции `stl` (STL – «Seasonal Decomposition of Time Series by Loess»), которая вычленяет из временного ряда три компоненты: сезонную (в данном случае, годовую), тренд и случайную, при помощи сглаживания данных методом LOESS.

```
> plot(stl(data, s.window="periodic")$time.series, main="")
```



Из графика видно, что тенденция к уменьшению смертности практически не прослеживается.

### 4.3 Стационарность

Проверим ряд на стационарность, используя тест Дики – Фуллера (DF-test). В этом нам поможет функция из пакета «tseries» `adf.test`. На вход ей подаётся временной ряд, гипотеза, которую нужно проверить и периодичность (принято брать годовую, а у нас месячные данные, значит,  $k = 12$ ).

```
> adf.test(data, alternative="stationary", k=12)
```

Augmented Dickey-Fuller Test

data: data

Dickey-Fuller = -1.6502, Lag order = 12, p-value = 0.7178

alternative hypothesis: stationary

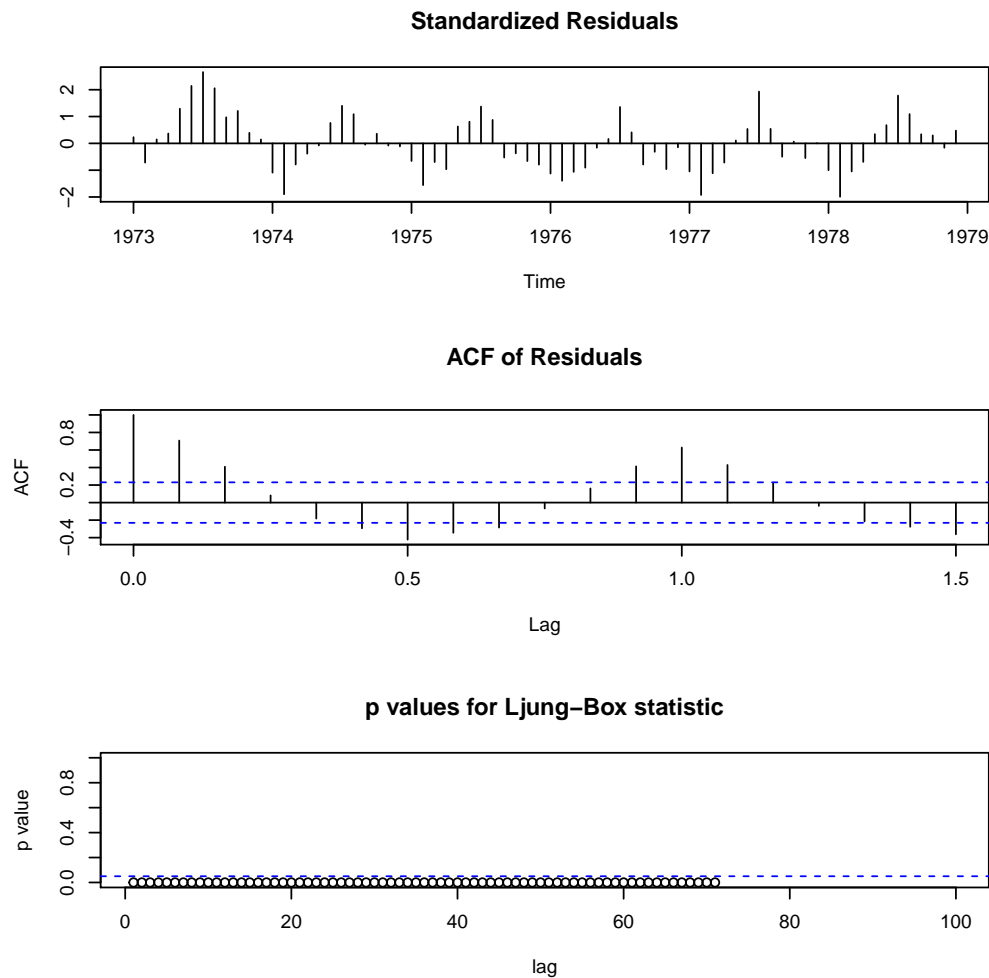
Значение  $p$  оказалось равным 0.7178. Таким образом, наш временной ряд стационарен с высокой вероятностью и имеет степень интегрированности  $I(0)$ .

## 5 Построение ARIMA модели

Построим модель временного ряда общего числа смертей распространенным методом ARIMA («Autoregressive Integrated Moving Average», авторегрессия интегрированного скользящего среднего). Аргумент `order` отвечает за несезонную часть модели ARIMA. Он состоит из трёх значений –  $(p, d, q)$ , где  $p$  является порядком авторегрессионной модели (AR),  $d$  отвечает за степень интегрирования, а  $q$  – за порядок модели скользящего среднего (MA). Таким образом, необходимо выбрать оптимальное значение параметра `order`. Для этого мы будем перебирать различные значения каждого из трех его компонентов. В этом нам поможет функция AIC. AIC (Akaike's Information Criterion) – информационный критерий Акаике. Это критерий, который позволяет сравнивать модели между собой. Чем меньше AIC, тем лучше модель.

Для начала, проверим, соответствуют ли остатки модели «белому шуму» на модели с параметром  $order = (0,0,0)$  с помощью функции `tsdiag`:

```
> tsdiag(arima(data, order = c(0,0,0)), 100)
```



Из графиков выше можно заметить, что остатки зависимы (это видно по ACF). p-value ниже уровня значимости, значит, неизвестно, являются ли остатки белым шумом.

## 5.1 Нахождения оптимальной модели

Найдём оптимальную модель:

1. Коэффициент МА:

```
> mini = 100500
> minInd = 0
> first <- 0
> for (m in 10:16) #должно быть 1:length(data)
+ {
+   mm <- arima(data, order=c(0,0,m))
+   cur = AIC(mm)
+   if(cur < mini){
+     minInd = m
+     mini = cur
+     first <- minInd
+   }
+ }
```

МА-коэффициент получился равным 15:

```
> first
```

```
[1] 15
```

2. Коэффициент AR:

```
> second <- 0
> mini <- 100500
> errCount <- 0
> for (m in 9:13)
+ {
+   a <- try(arima(data, order=c(m,0,first)))
+   if(class(a) == "try-error"){
+     errCount <- errCount + 1
+     next
+   }
+   cur = AIC(a)
+   if(cur < mini){
+     minInd = m
+     mini = cur
+     second <- minInd
+   }
+ }
```

AR-коэффициент получился равным 12:

```
> second
```

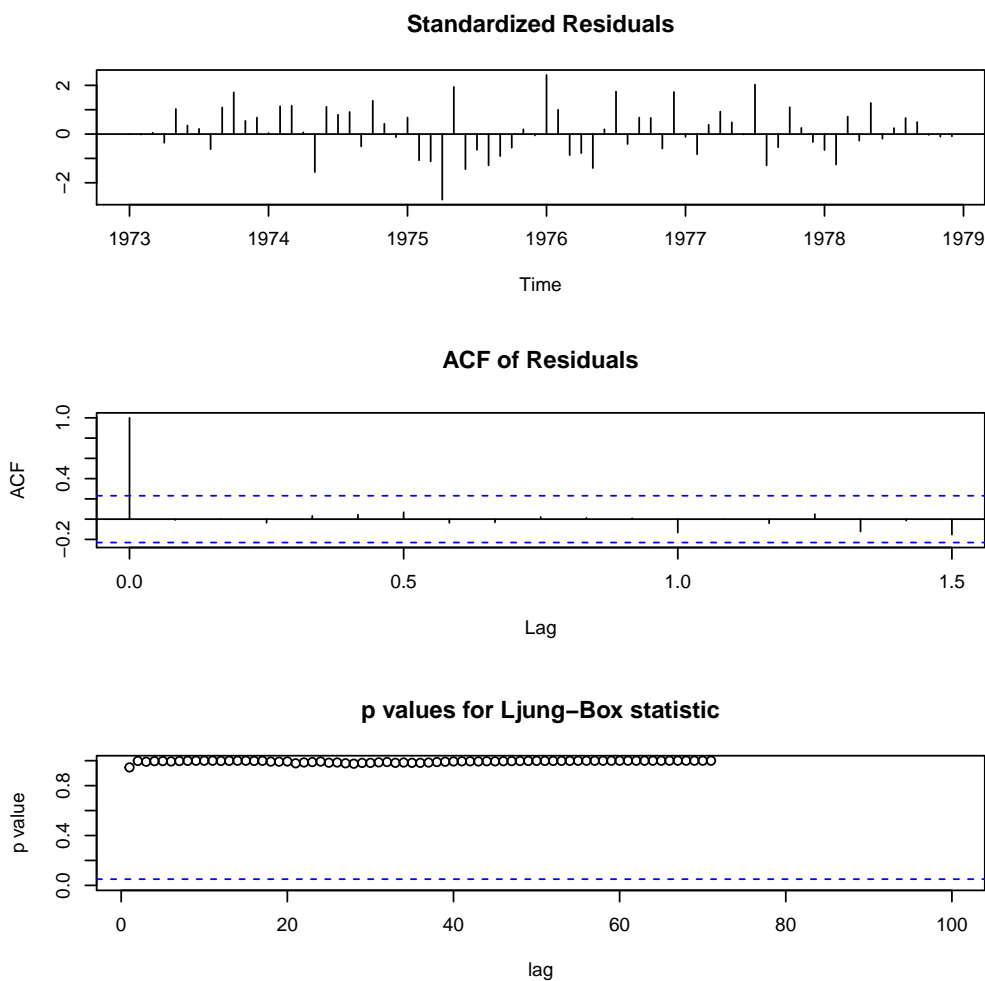
```
[1] 11
```

3. Найдём оптимальную модель:

```
> mini <- 100500
> errCount <- 0
> third <- arima(data, order=c(second,0,first))
> for (m in 3:5)
+ {
+   a <- try(arima(data, order=c(12,m,15)))
+   if(class(a) == "try-error"){
+     errCount <- errCount + 1
+     next
+   }
+   cur = AIC(a)
+   if(cur < mini){
+     minInd = m
+     mini = cur
+     third <- a
+   }
+ }
```

Снова проверим, соответствуют ли остатки модели «белому шуму»:

```
> tsdiag(third, 100)
```



Из графика видно, что остатки независимы (это видно по ACF). p-value гораздо больше уровня значимости, значит, остатки, с большой вероятностью, являются «белым шумом». Значит, модель получилась хорошей.

## 5.2 Обработка полученных данных и прогноз

С помощью predict обработаем полученные данные. Полученные цифры прогноза:

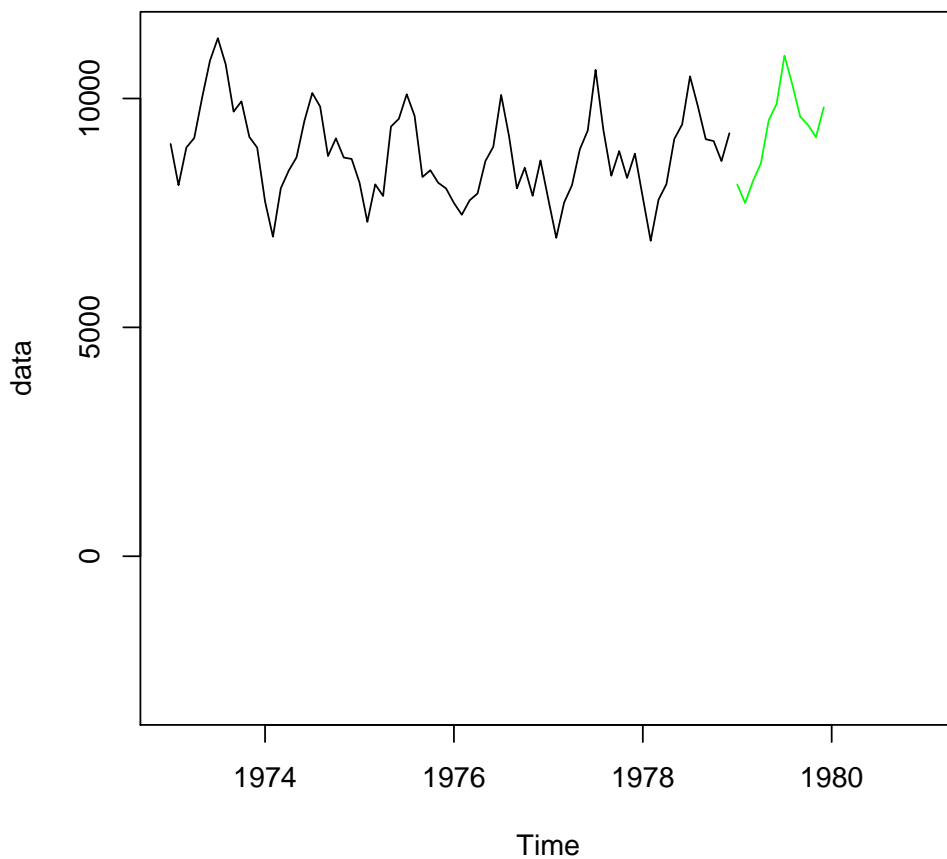
```
> predict(third, n.ahead = 12, se.fit = TRUE)$pred
```

	Jan	Feb	Mar	Apr	May	Jun	Jul
1979	8121.523	7718.303	8197.107	8580.422	9526.644	9882.218	10937.174
	Aug	Sep	Oct	Nov	Dec		
1979	10303.080	9602.219	9419.893	9157.032	9809.976		

На основании прогноза построим график с 1973 по 1979 годы:

```
> plot(data, xlim=c(1973,1981), ylim=c(min(data)-10000,max(data)))
> lines(predict(third, n.ahead=12, se.fit = TRUE)$pred, col="green")
```





Полученные цифры возможной ошибки:

```
> predict(third, n.ahead = 12, se.fit = TRUE)$se
```

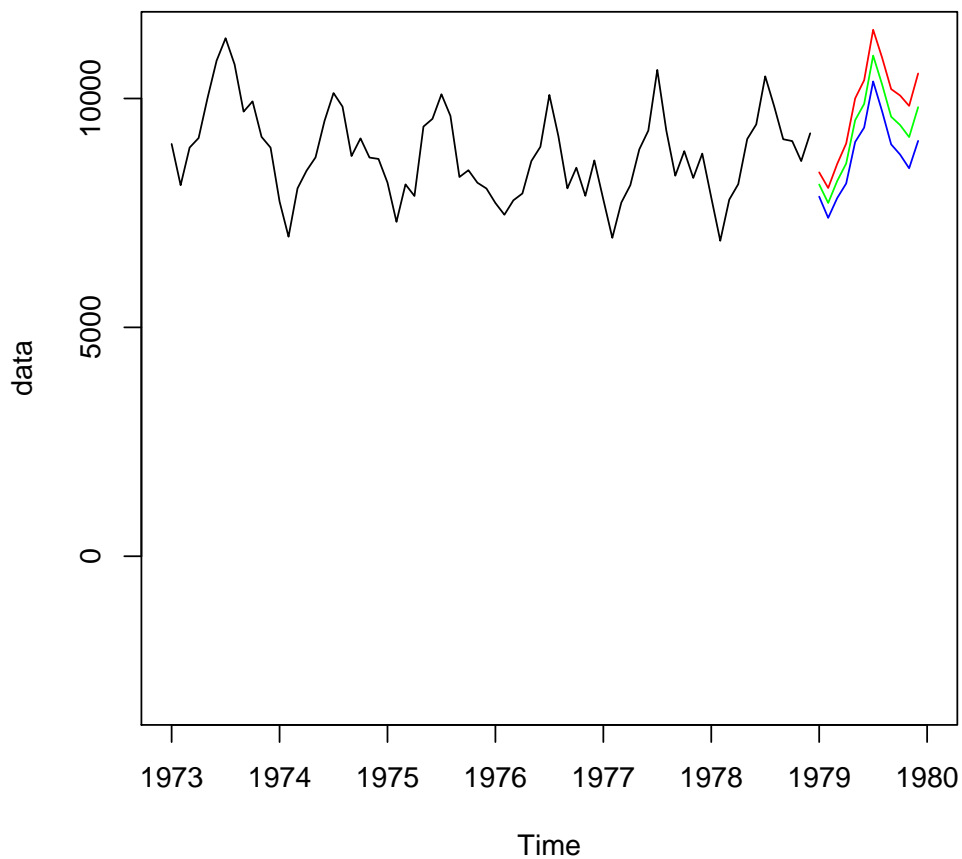
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
1979	265.2952	326.6829	372.4956	436.9623	475.6924	517.4967	564.4794	585.3229

	Sep	Oct	Nov	Dec
1979	603.5406	645.0931	681.8046	736.2778

Добавим на график верхние и нижние границы прогноза:

```
> plot(data, xlim=c(1973,1980), ylim=c(min(data)-10000,max(data)))
> lines(predict(third, n.ahead=12, se.fit = TRUE)$pred, col="green")
> lines(predict(third,
+             n.ahead=12,
+             se.fit = TRUE)$se +
+             predict(third, n.ahead=12, se.fit = TRUE)$pred,
+             col="red")
> lines(-predict(third, n.ahead=12, se.fit = TRUE)$se +
+       predict(third, n.ahead=12, se.fit = TRUE)$pred,
+       col="blue")
```



## 6 Вывод

В ходе данной работы мы научились строить ARMA-модели и находить самые оптимальные из них, также, на основе полученной модели, мы научились строить прогнозы при помощи языка программирования R и инструмента R-studio из проекта R-project.