

Task 1

Oleg Chaban

1 этап

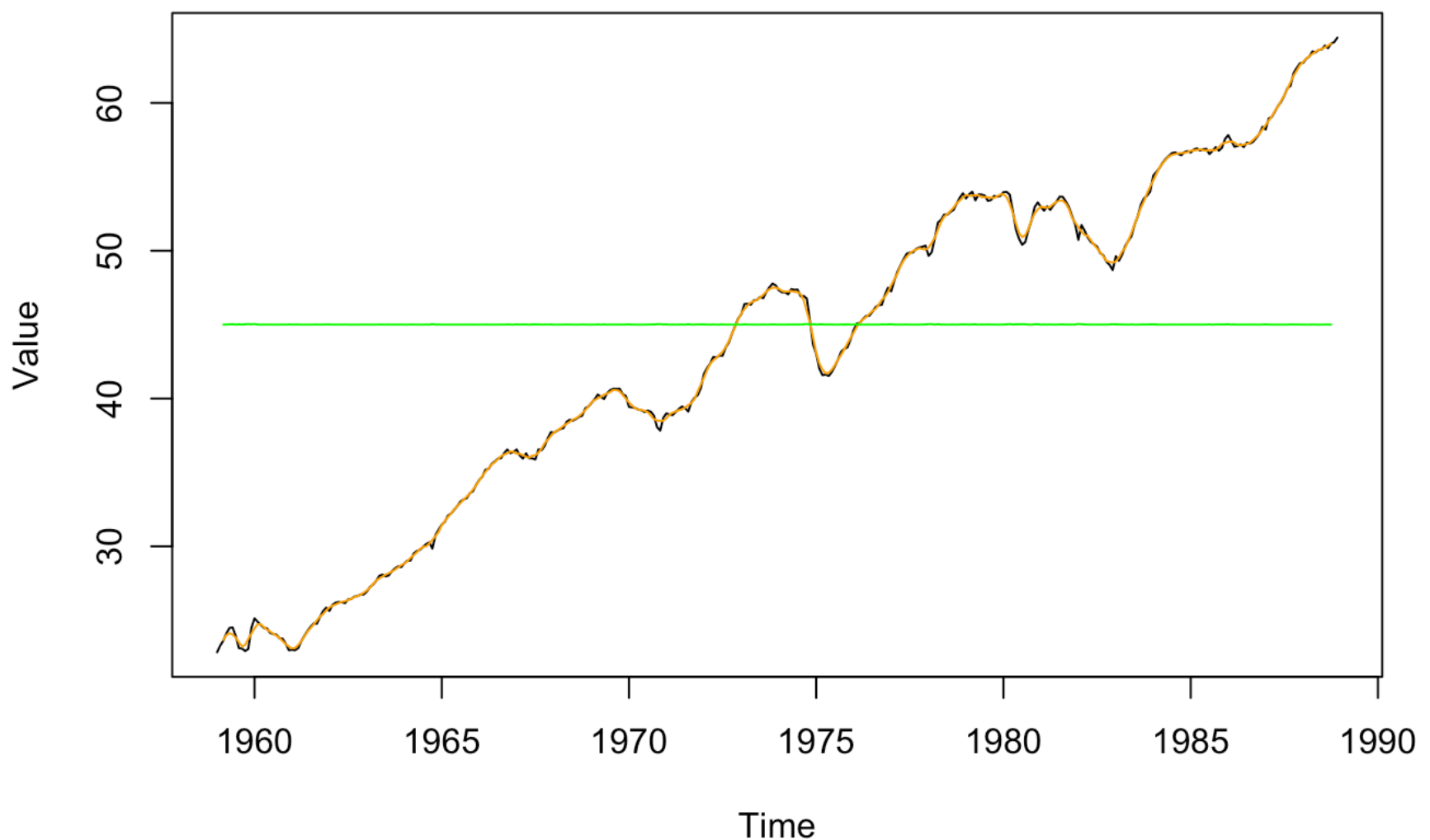
Считываем данных из training.csv.

```
dates_rows<-read.csv("training.csv", header = TRUE, row.names = 1)
dates_rows.ts<-ts(dates_rows, start=c(1959,1), frequency=12)
```

Построим скользящие статистики - скользящее среднее и стандартное отклонение.

```
dates_rows_rl<-rollmean(dates_rows.ts, 5)
dates_rows_d<-((dates_rows_rl-dates_rows.ts)^2/359)^(1/2)
```

Строим график, на котором отрисованны сам ряд и его скользящие статистики. Скользящее среднее окрашено оранжевым цветом, стандартное отклонение - зеленым.



Изучив данные графики, можно сказать, что что ряд является нестационарным.

Проведем тест Дики-Фуллера.

```
adf.test(dates_rows.ts)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data:  dates_rows.ts  
## Dickey-Fuller = -3.2505, Lag order = 7, p-value = 0.07962  
## alternative hypothesis: stationary
```

Так как корень один, то можно считать, что ряд нестационарен. В результате, после проведения двух тестов, можно сказать, что исходный ряд является нестационарным.

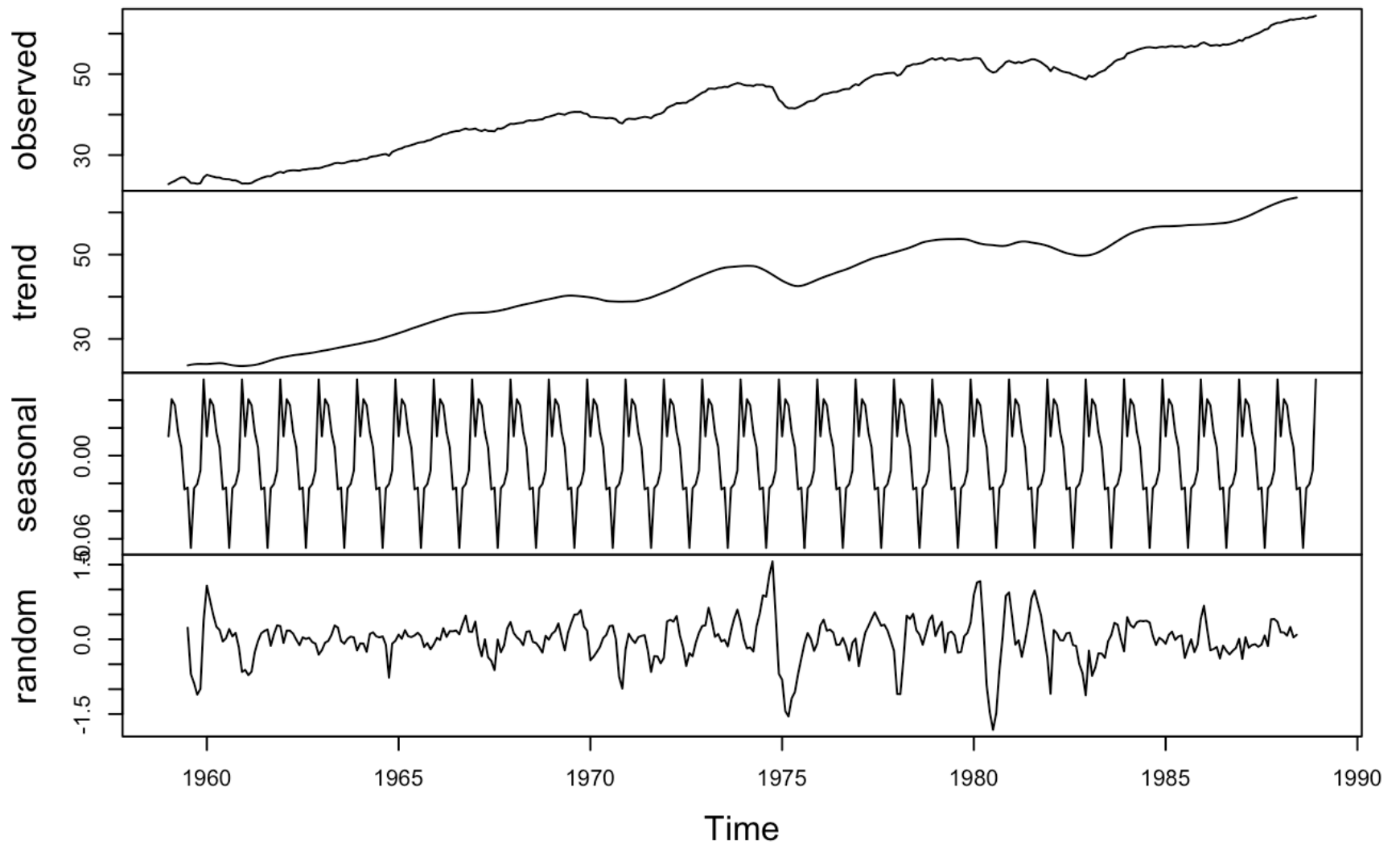
2 этап

Раскладываем временной ряд на тренд, сезональность, остаток в соответствии с аддитивной, мультипликативной моделями.

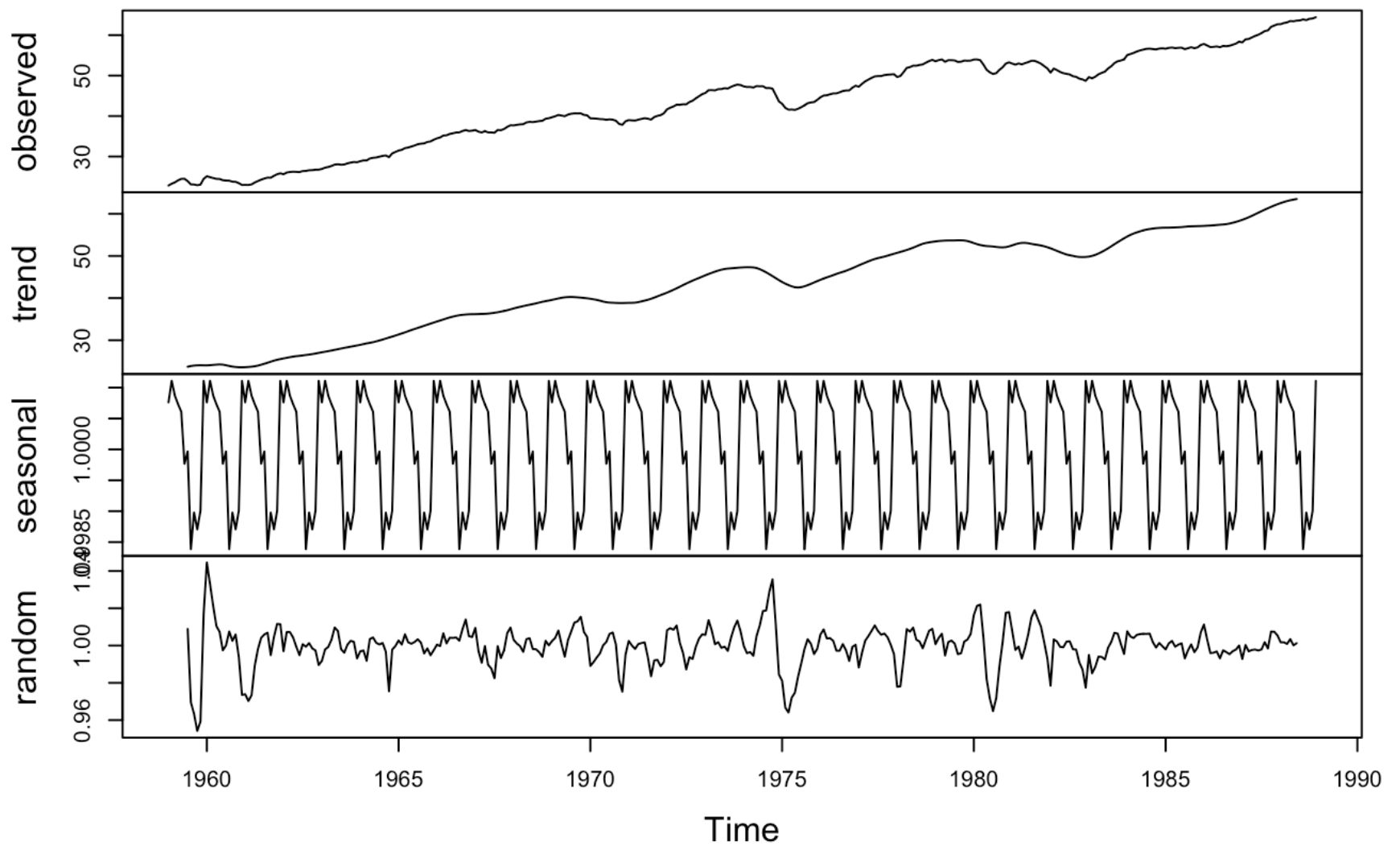
```
dates_rows_add<-decompose(dates_rows.ts,type = "additive")  
dates_rows_mul<-decompose(dates_rows.ts,type = "multiplicative")
```

Построим графики.

Decomposition of additive time series



Decomposition of multiplicative time series



В обеих моделях видно, что тренд не стационарен, но сезональность и остаток - стационарны в широком смысле, т.к. статистические характеристики не изменяются с течением времени. Чтобы удостовериться в этом, проведем тест Дики-Фуллера.

```
adf.test(na.remove(dates_rows_add$seasonal), k = 0)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: na.remove(dates_rows_add$seasonal)  
## Dickey-Fuller = -10.169, Lag order = 0, p-value = 0.01  
## alternative hypothesis: stationary
```

```
adf.test(na.remove(dates_rows_add$trend), k = 0)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: na.remove(dates_rows_add$trend)  
## Dickey-Fuller = -0.6874, Lag order = 0, p-value = 0.9711  
## alternative hypothesis: stationary
```

```
adf.test(na.remove(dates_rows_add$random), k = 0)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: na.remove(dates_rows_add$random)  
## Dickey-Fuller = -7.3949, Lag order = 0, p-value = 0.01  
## alternative hypothesis: stationary
```

```
adf.test(na.remove(dates_rows_mul$seasonal), k = 0)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: na.remove(dates_rows_mul$seasonal)  
## Dickey-Fuller = -9.192, Lag order = 0, p-value = 0.01  
## alternative hypothesis: stationary
```

```
adf.test(na.remove(dates_rows_mul$trend), k = 0)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: na.remove(dates_rows_mul$trend)  
## Dickey-Fuller = -0.6874, Lag order = 0, p-value = 0.9711  
## alternative hypothesis: stationary
```

```
adf.test(na.remove(dates_rows_mul$random), k = 0)
```

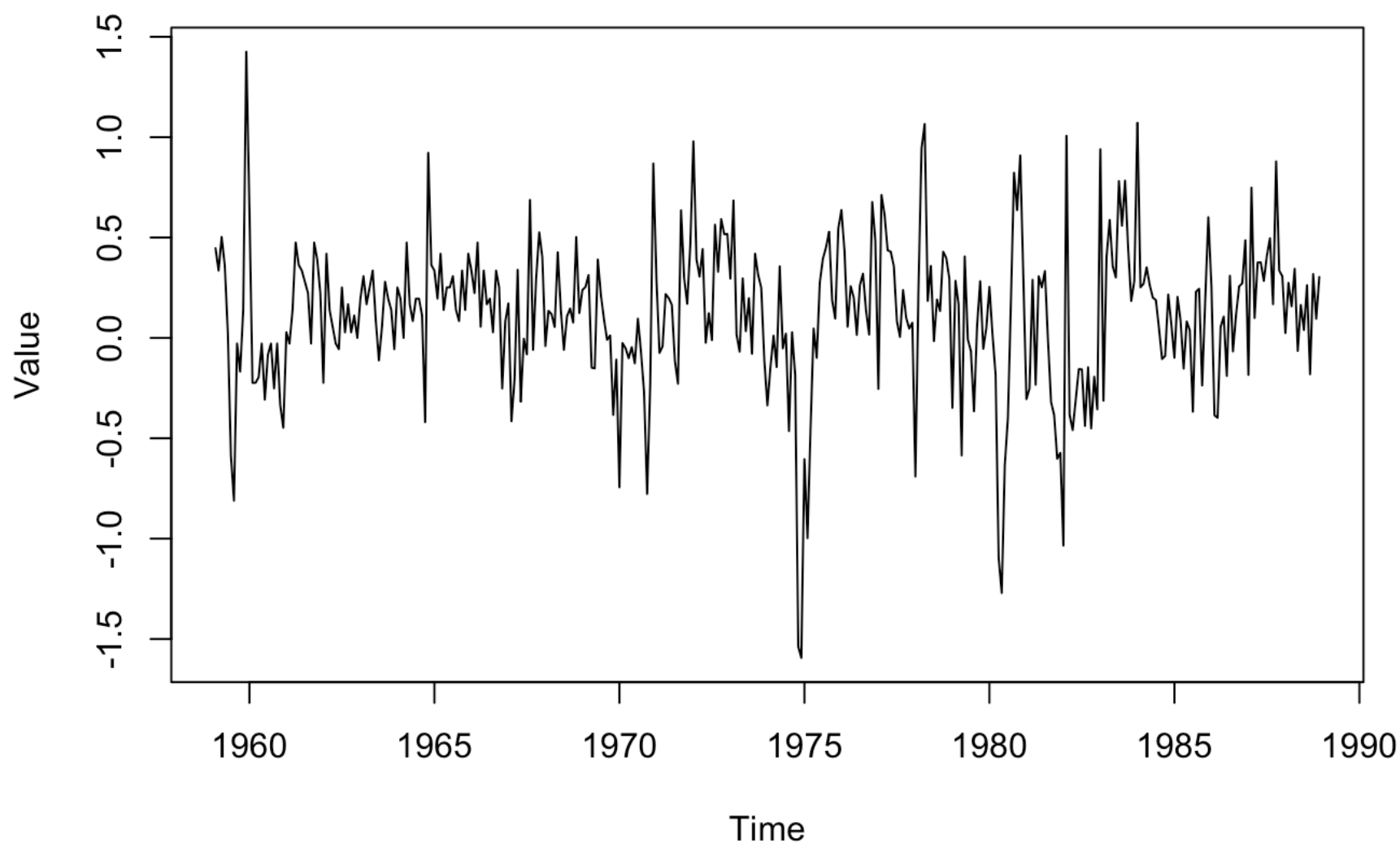
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: na.remove(dates_rows_mul$random)  
## Dickey-Fuller = -7.5842, Lag order = 0, p-value = 0.01  
## alternative hypothesis: stationary
```

Наше предположение верно. В обеих моделях значение параметра p-value у тренда большое, следовательно, ряд не является стационарным. А для остатка и сезональности он мал, что подтверждает их стационарность.

3 этап

Проанализируем последовательно разности временного ряда, для того, чтобы определить порядок интегрированности этого ряда.

```
diff_n<-diff(dates_rows.ts, differences=1)  
plot.ts(diff_n)
```



```
adf.test(diff_n, k = 0)
```

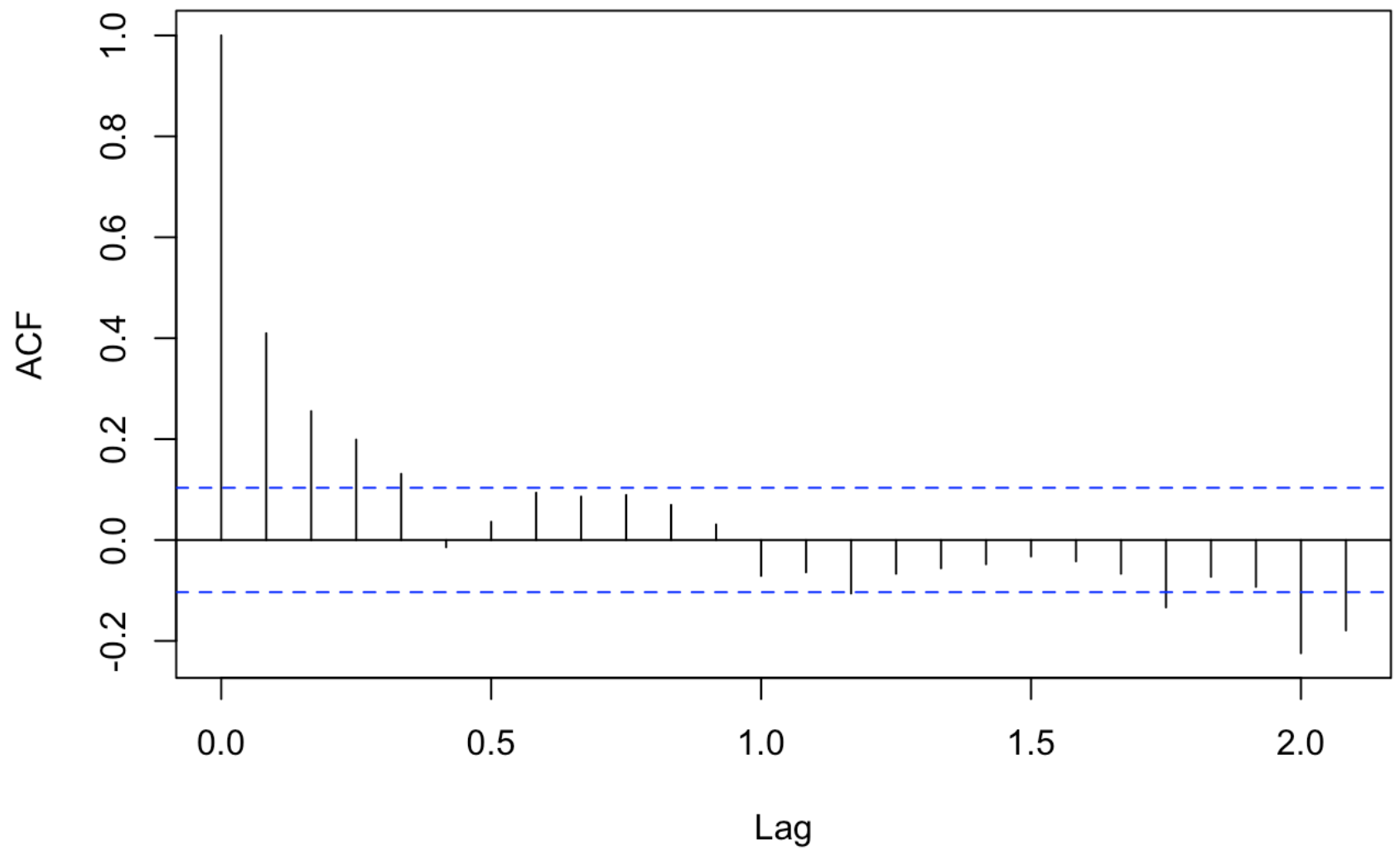
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: diff_n  
## Dickey-Fuller = -12.2, Lag order = 0, p-value = 0.01  
## alternative hypothesis: stationary
```

Из результатов теста Дики-Фуллера можно сделать вывод, что порядок интегрированности равен 1, т.к. первая разность исходного ряда - стационарна.

Построим функции автокоррелеации и частичной автокоррелеации

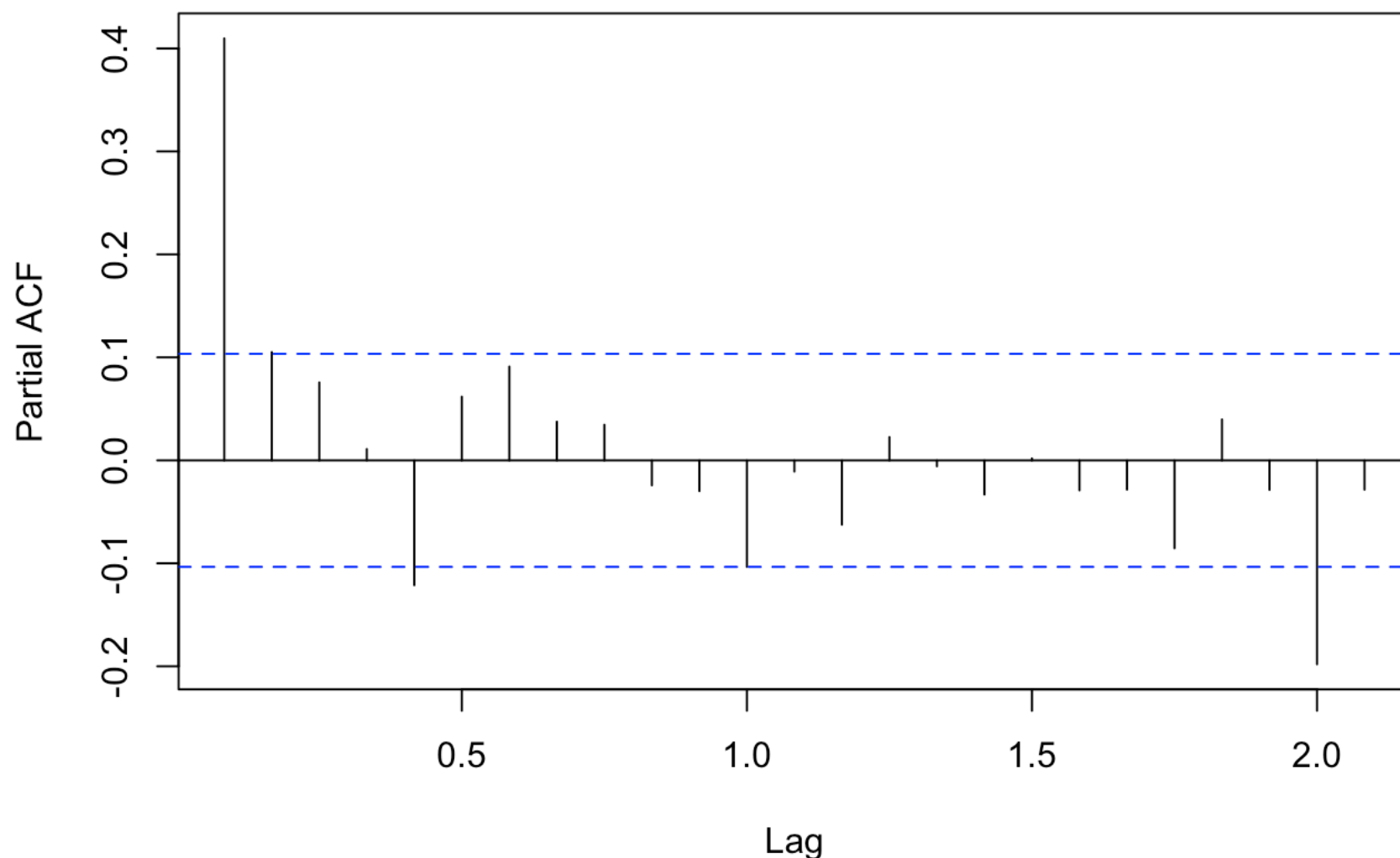
```
acf(diff_n, main="Автокоррелеация")
```

Автокоррелеация



```
pacf(diff_n,main="Частичная автокоррелеация")
```

Частичная автокорреляция



С помощью `pacf` и `acf` находим параметры p и q для построения моделей ARIMA. Параметр d равен порядку интегрированного ряда. Построим несколько моделей ARIMA с параметрами, выбранными таким образом, что: $0 \leq p \leq 2$, $d=1$, $0 \leq q \leq 4$.

```
model01<-Arima(dates_rows.ts,order=c(2,1,0))
model02<-Arima(dates_rows.ts,order=c(0,1,4))
model03<-Arima(dates_rows.ts,order=c(2,1,4))
model04<-Arima(dates_rows.ts,order=c(1,1,1))
```

Считываем данные из `testing.csv`.

```
test<-read.csv("testing.csv",header=TRUE, row.names = 1)
test.ts<-ts(test, start=c(1989,1), frequency=12)
```

Вычисляем для каждой модели прогноз с помощью `forecast.Arima`. Так же проверяем, что для каждого прогноза значение `r2_score` достаточно близко к нулю. Визуализируем `forecast.Arima` для всех моделей.

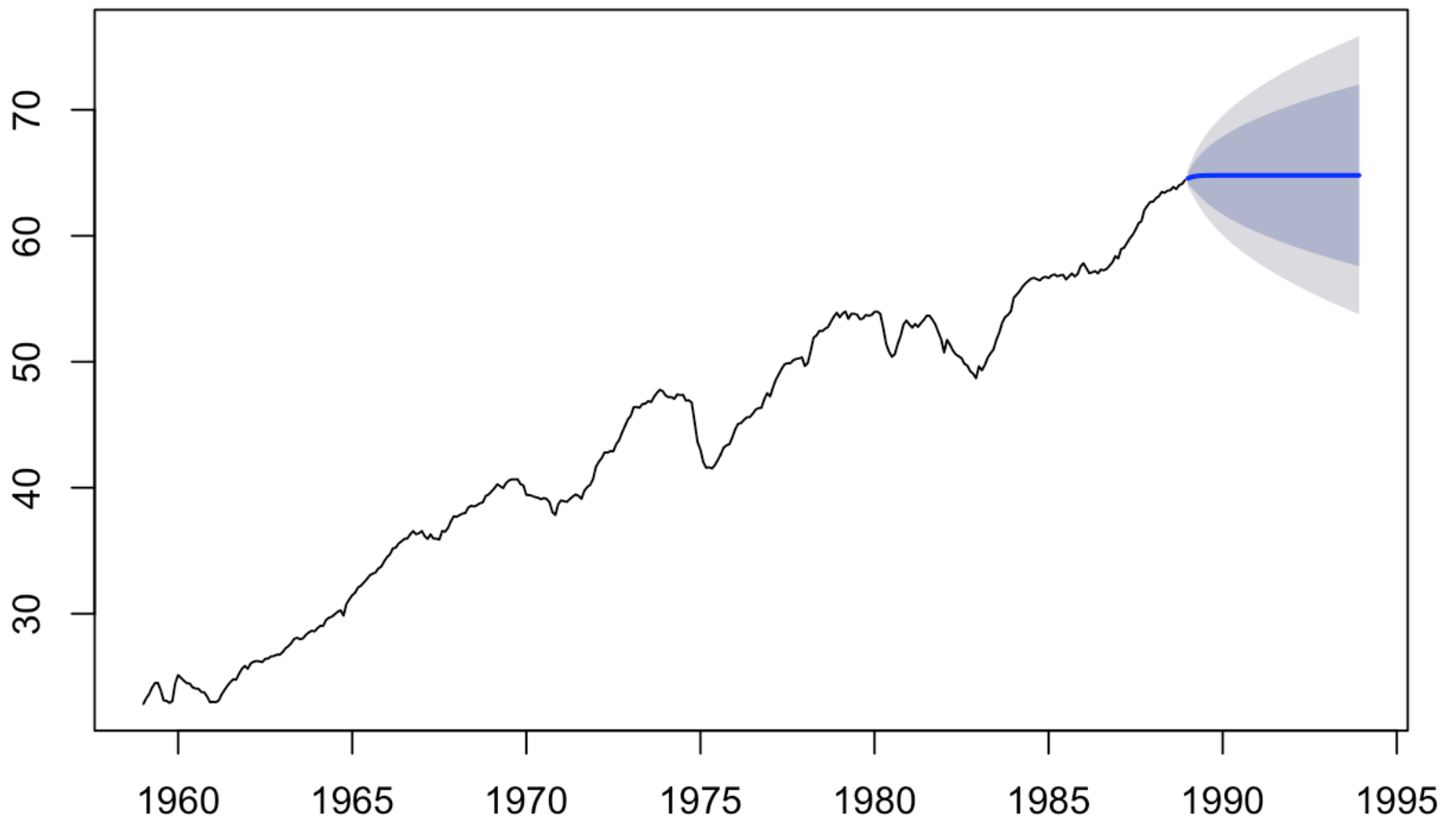
```
frcst_1<-forecast.Arima(model01,h=60)
R2_Score(frcst_1$mean,test.ts)
```

```
## [1] -0.08292902
```



```
plot.forecast(frcst_1)
```

Forecasts from ARIMA(2,1,0)

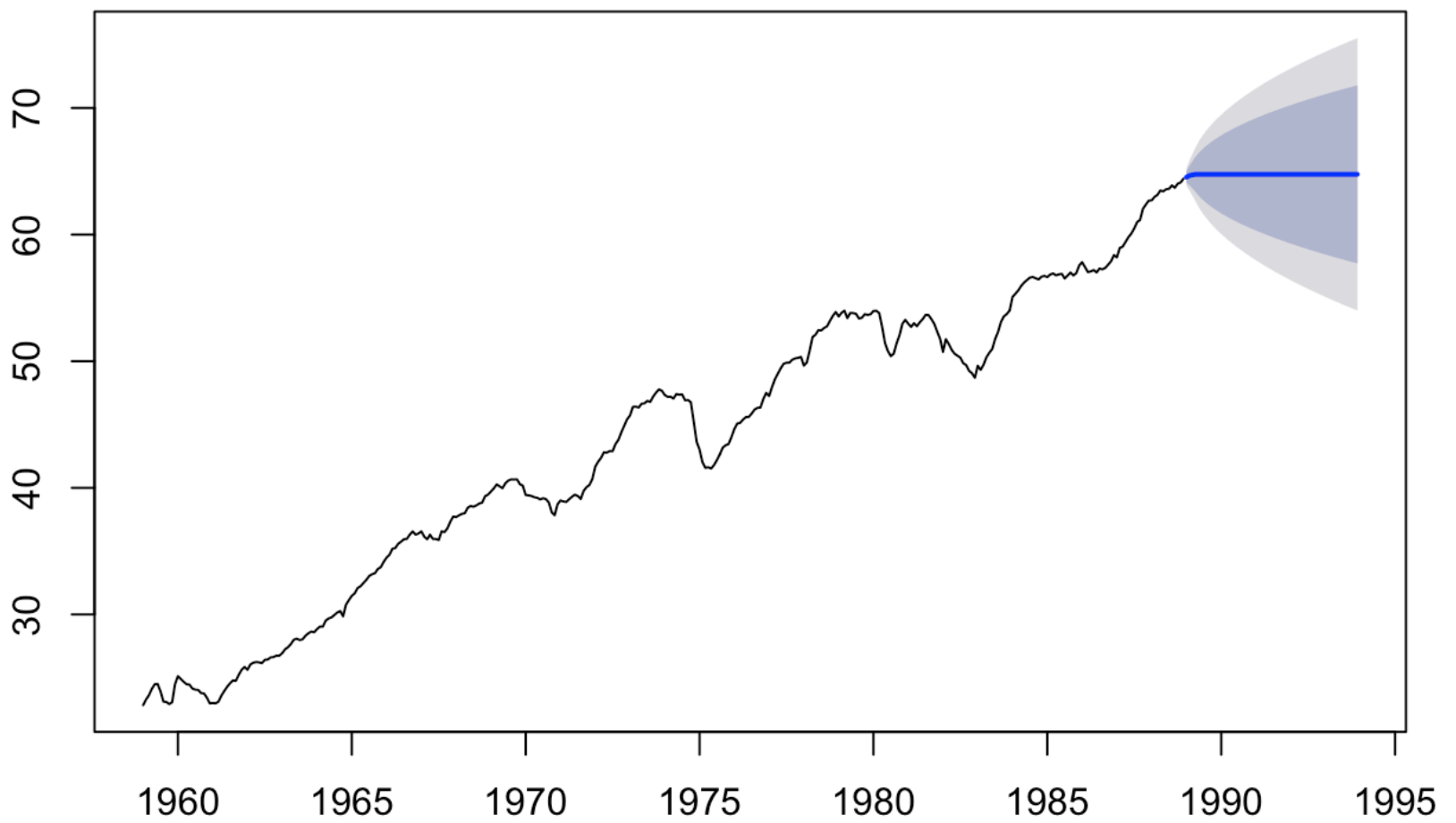


```
frcst_2<-forecast.Arima(model02,h=60)  
R2_Score(frcst_2$mean,test.ts)
```

```
## [1] -0.09951206
```

```
plot.forecast(frcst_2)
```

Forecasts from ARIMA(0,1,4)

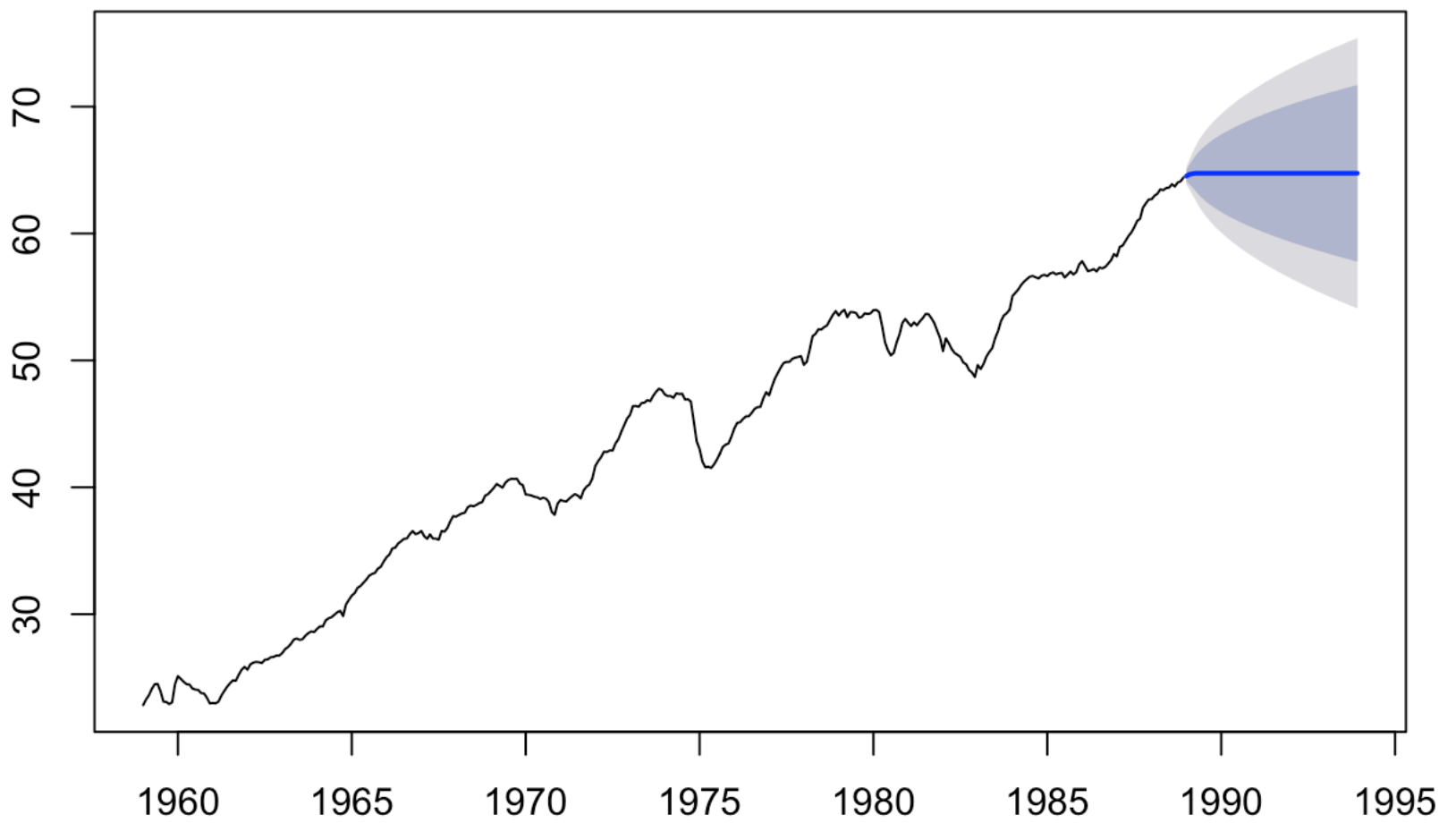


```
frcst_3<-forecast.Arima(model03,h=60)  
R2_Score(frcst_3$mean,test.ts)
```

```
## [1] -0.104477
```

```
plot.forecast(frcst_3)
```

Forecasts from ARIMA(2,1,4)

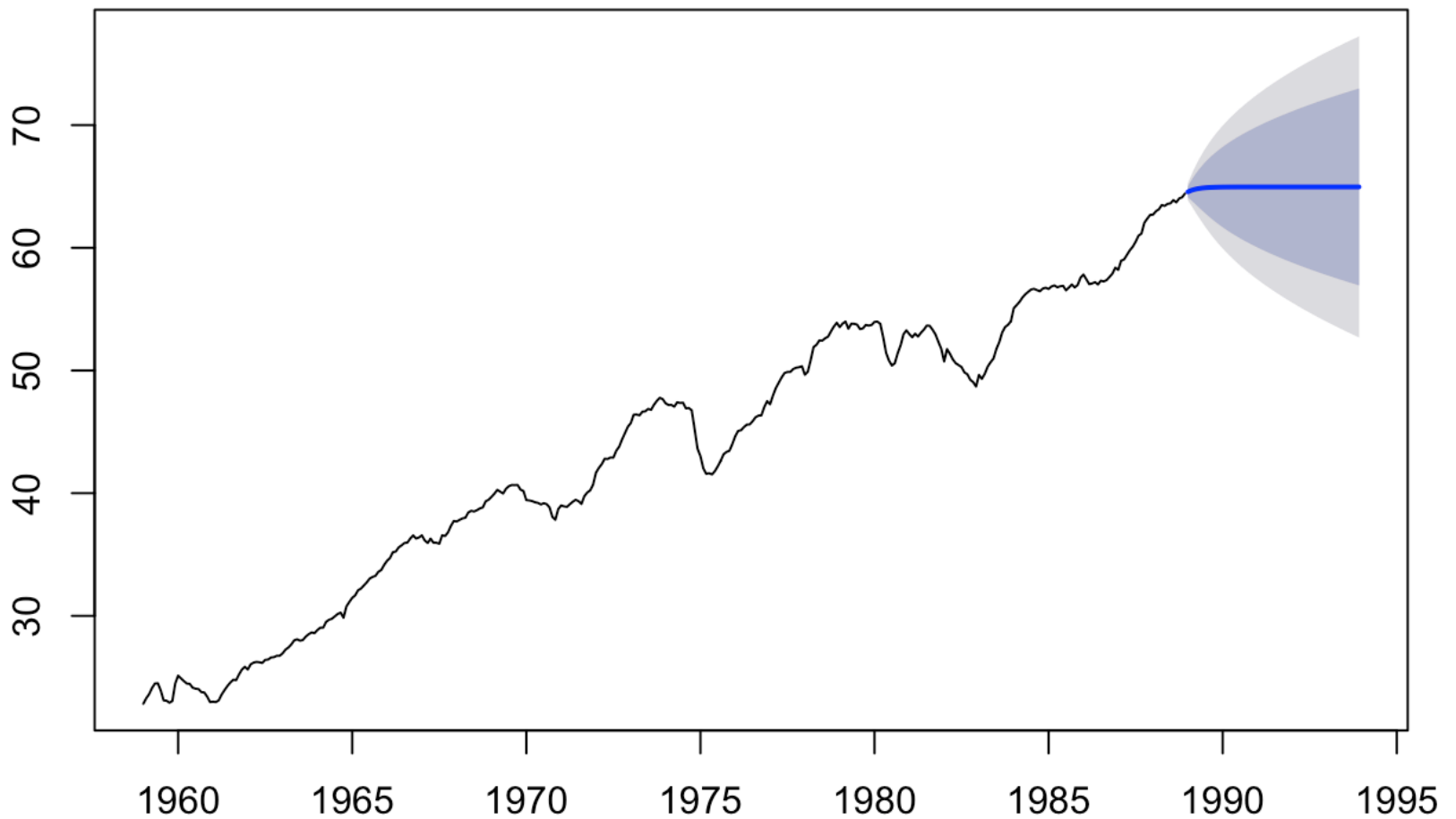


```
frcst_4<-forecast.Arima(model04,h=60)  
R2_Score(frcst_4$mean,test.ts)
```

```
## [1] -0.02447902
```

```
plot.forecast(frcst_4)
```

Forecasts from ARIMA(1,1,1)



С помощью информационного критерия Акаике(AIC) из всех этих моделей находим наилучшую: выбираем модель с минимальным значением критерия AIC.

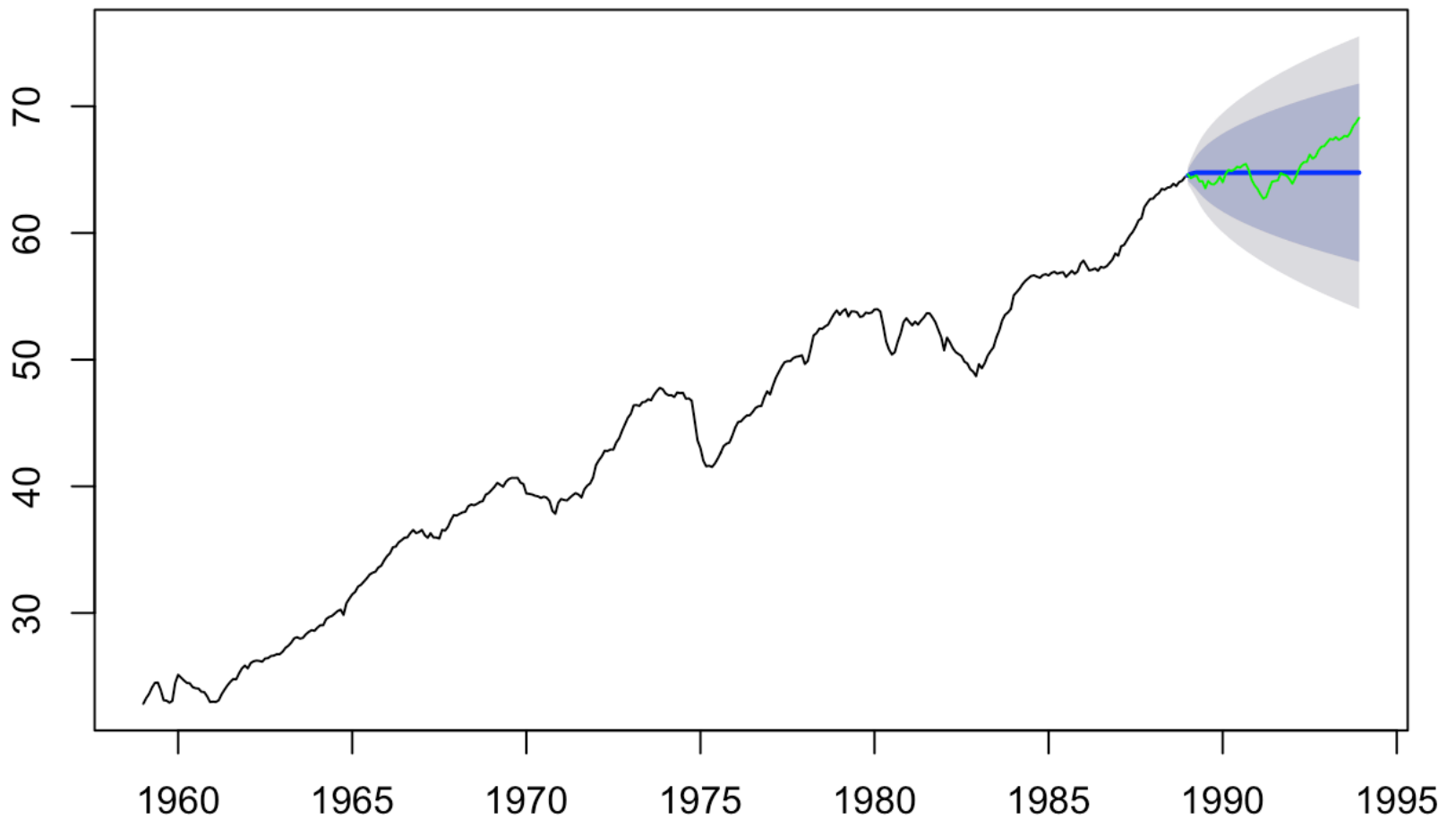
```
AIC(model01,model02,model03,model04)
```

##		df	AIC
##	model01	3	258.5099
##	model02	5	253.3399
##	model03	7	257.3009
##	model04	3	255.6413

Визуализируем `forecast.Arima` выбранной модели и получаем область прогноза дальнейших значений временного ряда. Строим график тестовой выборки.

```
plot.forecast(frcst_2)  
lines(test.ts,col='green')
```

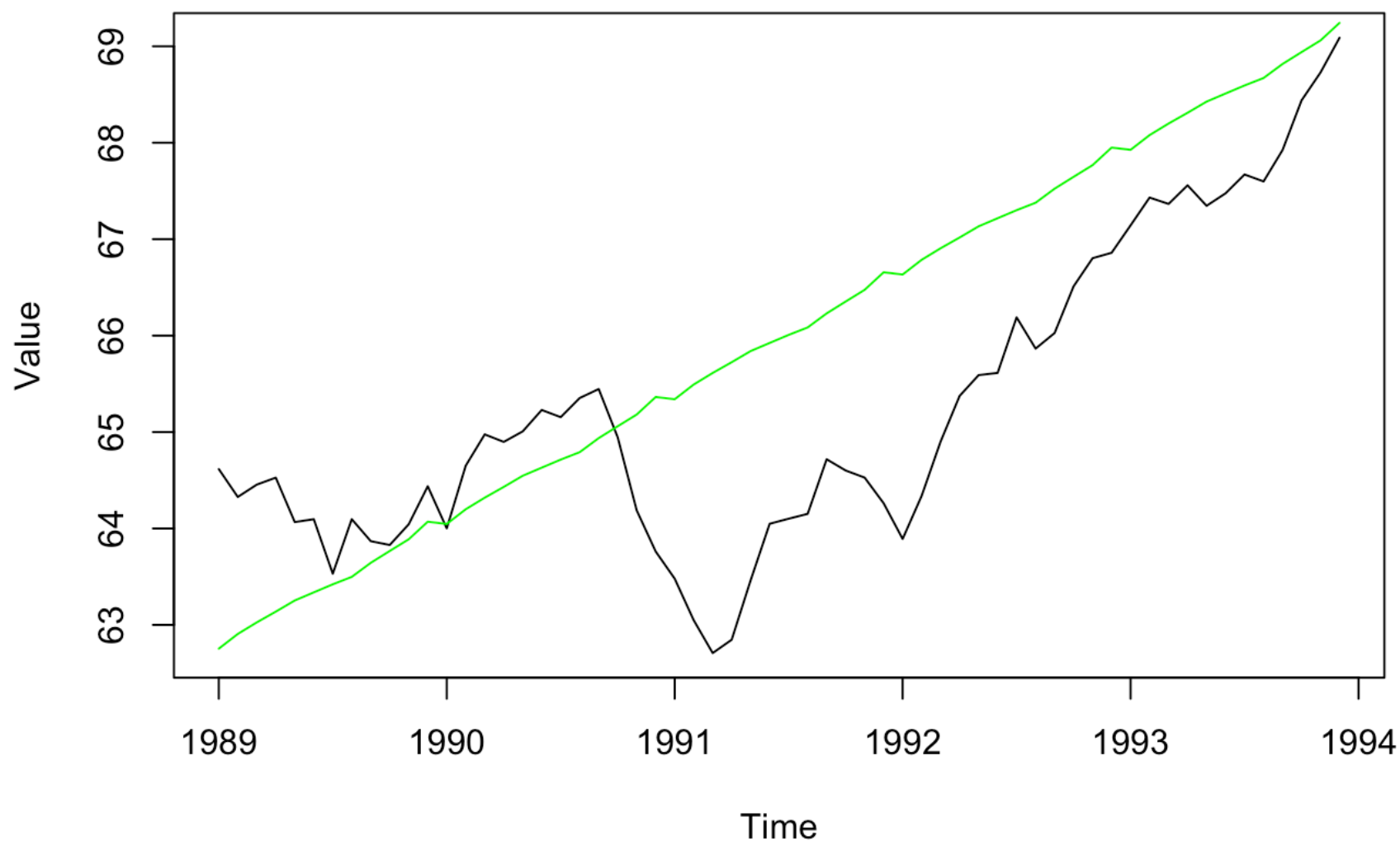
Forecasts from ARIMA(0,1,4)



Мы убедились, что график лежит внутри спрогнозируемой области. Однако, r^2 -score для наших моделей - отрицательный. Отсюда можно сделать вывод, что прогнозирование на 60 месяцев с помощью модели ARIMA не подходит. Вопрос в том, что же нам подойдет?

Давайте рассмотрим линейную модель, разбив ряд на тренд и сезонность:

```
dates_rows.lin <- tslm(dates_rows.ts ~ trend + season)
frcst_lin <- forecast.lm(dates_rows.lin, h = 60)
plot(test.ts)
lines(frcst_lin$mean, col = 'green')
```



```
R2_Score(frcst_lin$mean,test.ts)
```

```
## [1] 0.1968743
```

В линейной модели получили, что $r2_score$ у нас больше нуля, отсюда можно сделать вывод, что данная модель является применимой для тестовых данных. Так же, хочется заметить, что 60 месяцев, предоставленных к прогнозу по заданию, являются неподходящим сроком для прогнозирования моделью ARIMA.