Projekt - Prognozowanie i symulacja zjawisk gospodarczych

Piotr Pasierb, Oskar Paśko, Michał Żychowski

2024-06-21

Projekt - Prognozowanie i symulacja zjawisk gospodarczych

Wczytanie bibliotek

Wczytuje potrzebne biblioteki do projektu.

```
library(dplyr)
library(zoo)
library(xts)
library(tsbox)
library(skedastic)
library(lmtest)
library(urca)
library(forecast)
library(ggplot2)
```

Wczytanie danych

Wczytuje dane wykorzystywane do analizy w projekcie.

```
setwd("C:/Users/micha/Desktop/Projekt_Prognozowanie_i_symulacja_zjawisk_gospodarczych")
data <- read.csv("Data/Dane.csv", sep = ";", dec = ",", colClasses = c("character","numeric","numeric",
data2 <- read.csv("Data/Dane2.csv", sep = ",")
data3 <- read.csv("Data/Dane3.csv", sep = ";")
```

Dalej dane są podzielone na 6 zbiorów. Pierwsze cztery zbiory to "Wskaźniki makroekonomiczne" z strony Głównego Urzędu Statystycznego. Dane zbiorów 5 i 6 są z strony kaggle.com. Zbiór piąty to dzienne dane pogodowe Indii. Szósty zbiór to godzinowe dane pogodowe Półwyspu Gallipoli.

Zbiór 1

```
zb1 <- data.frame(data$Data,data$Podaż.pieniądza.ogółem.M3..w.mln.zł.)
colnames(zb1) <- c("Data", "Podaż pieniądza ogółem M3 w mln zł.")
head(zb1, n=10)
```

```
##            Data Podaż pieniądza ogółem M3 w mln zł.
## 1    I kw. 2000                            269788.1
```

```
## 2   II kw. 2000                            291886.9
## 3  III kw. 2000                            289140.2
## 4   IV kw. 2000                            300757.3
## 5    I kw. 2001                            309465.8
## 6   II kw. 2001                            315025.5
## 7  III kw. 2001                            327153.5
## 8   IV kw. 2001                            329704.7
## 9    I kw. 2002                            321319.3
## 10  II kw. 2002                            325076.4
```

Zbiór 2

```r
zb2 <- data.frame(data$Data,data$Wskaźnik.ogólnego.klimatu.koniunktury.w.budownictwie)
colnames(zb2) <- c("Data", "Wskaźnik ogólnego klimatu koniunktury w budownictwie")
head(zb2, n=10)
```

```
##             Data Wskaźnik ogólnego klimatu koniunktury w budownictwie
## 1    I kw. 2000                                                  22.2
## 2   II kw. 2000                                                  24.2
## 3  III kw. 2000                                                   9.8
## 4   IV kw. 2000                                                 -24.4
## 5    I kw. 2001                                                   1.9
## 6   II kw. 2001                                                   8.1
## 7  III kw. 2001                                                  -1.9
## 8   IV kw. 2001                                                 -35.1
## 9    I kw. 2002                                                 -10.0
## 10  II kw. 2002                                                   6.8
```

Zbiór 3

```r
zb3 <- data.frame(data$Data,data$Mieszkania.oddane.do.użytkowania..w.tys)
colnames(zb3) <- c("Data", "Mieszkania oddane do użytkowania w tys")
head(zb3, n=10)
```

```
##             Data Mieszkania oddane do użytkowania w tys
## 1    I kw. 2000                                    18.0
## 2   II kw. 2000                                    16.3
## 3  III kw. 2000                                    20.5
## 4   IV kw. 2000                                    33.0
## 5    I kw. 2001                                    25.8
## 6   II kw. 2001                                    21.9
## 7  III kw. 2001                                    24.2
## 8   IV kw. 2001                                    34.1
## 9    I kw. 2002                                    21.9
## 10  II kw. 2002                                    20.9
```

Zbiór 4

```r
zb4 <- data.frame(data$Data,data$Dochody.budżetu.państwa.ogółem..w.mln.zł.)
colnames(zb4) <- c("Data", "Dochody budżetu państwa ogółem w mln zł.")
head(zb4, n=10)
```

```
##            Data Dochody budżetu państwa ogółem w mln zł.
## 1     I kw. 2000                                  30949.7
## 2    II kw. 2000                                  64244.2
## 3   III kw. 2000                                  97880.7
## 4    IV kw. 2000                                 135663.9
## 5     I kw. 2001                                  31623.2
## 6    II kw. 2001                                  67729.7
## 7   III kw. 2001                                 102775.5
## 8    IV kw. 2001                                 140526.9
## 9     I kw. 2002                                  31275.3
## 10   II kw. 2002                                  65111.0
```

Zbiór 5

```r
zb5 <- data.frame(data2$date,data2$meantemp)
zb5$data2.meantemp <- round(zb5$data2.meantemp, digits = 2)
colnames(zb5) <- c("Data", "Średnia temperatura w ciągu dnia.")
head(zb5, n=10)
```

```
##          Data Średnia temperatura w ciągu dnia.
## 1  2017-01-01                             15.91
## 2  2017-01-02                             18.50
## 3  2017-01-03                             17.11
## 4  2017-01-04                             18.70
## 5  2017-01-05                             18.39
## 6  2017-01-06                             19.32
## 7  2017-01-07                             14.71
## 8  2017-01-08                             15.68
## 9  2017-01-09                             14.57
## 10 2017-01-10                             12.11
```

Zbiór 6

```r
zb6 <- data.frame(data3$DateTime,data3$Temperature)
colnames(zb6) <- c("Data i godzina", "Temperatura.")
head(zb6, n=10)
```

```
##      Data i godzina Temperatura.
## 1  1.01.2008 03:00         -0.1
## 2  1.01.2008 04:00          2.5
## 3  1.01.2008 05:00          2.9
## 4  1.01.2008 06:00          3.9
## 5  1.01.2008 07:00          4.9
## 6  1.01.2008 08:00          5.0
```

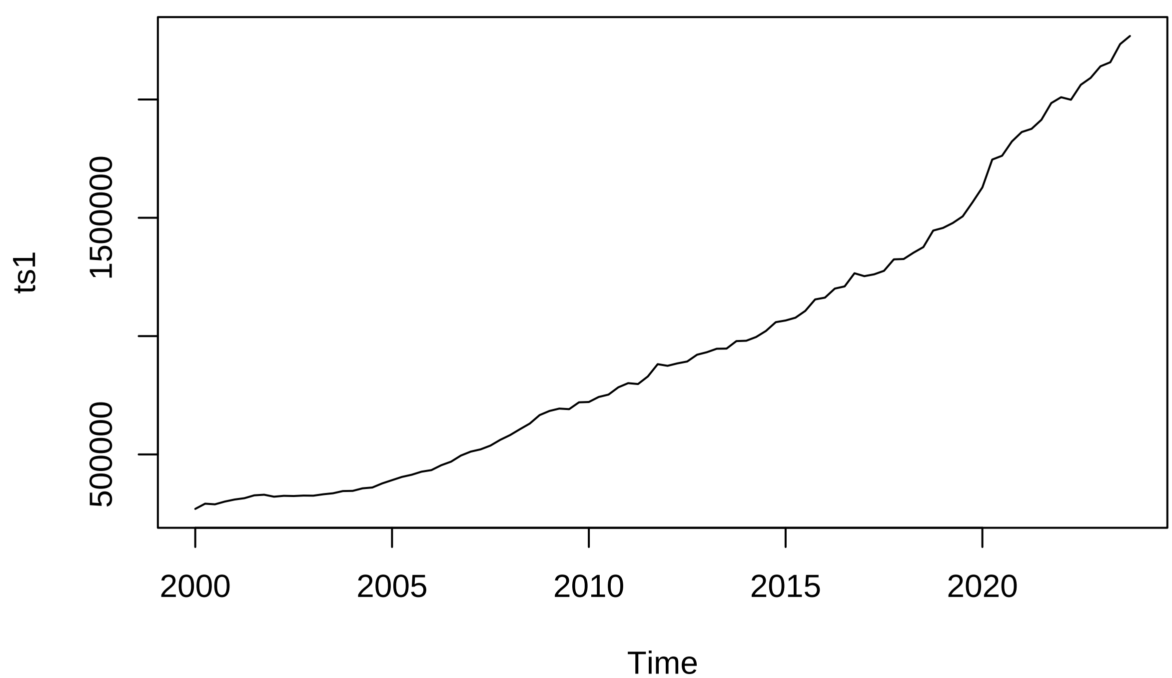
```
## 7  1.01.2008 09:00          5.2
## 8  1.01.2008 10:00          6.4
## 9  1.01.2008 11:00          7.6
## 10 1.01.2008 12:00          8.5
```

Szeregi czasowe

Utworzenie szeregów czasowych z wczytanych zbiorów danych.

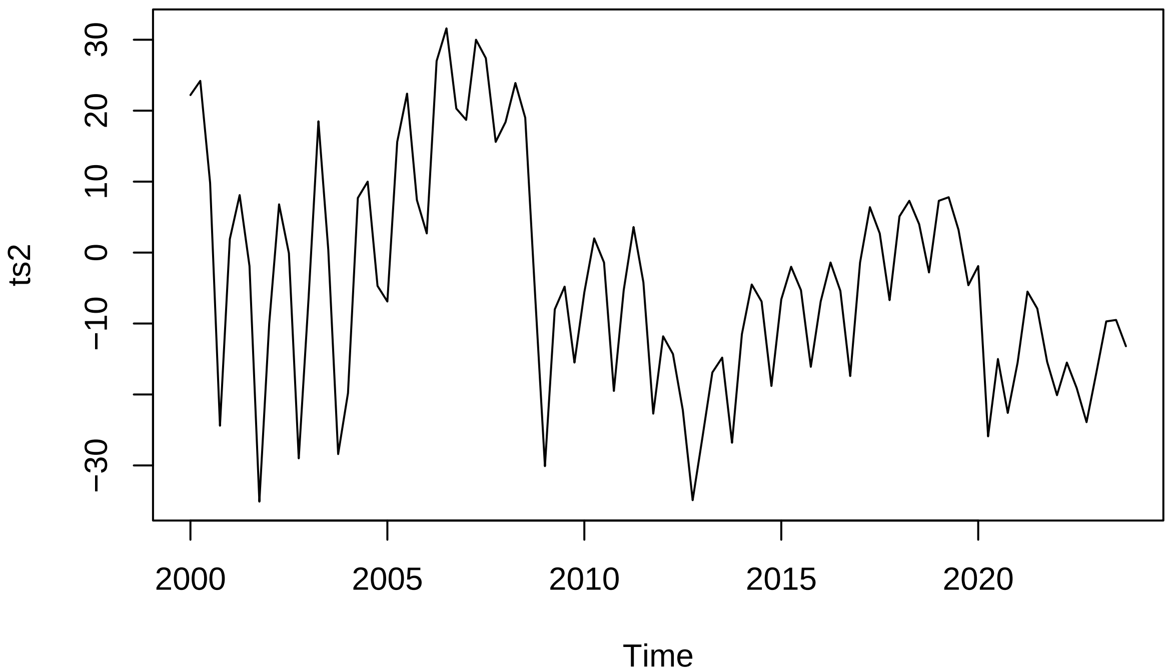
Szereg pierwszy

```
ts1 <- ts(zb1$`Podaż pieniądza ogółem M3 w mln zł.`, start = c(2000, 1), frequency = 4)
plot(ts1)
```



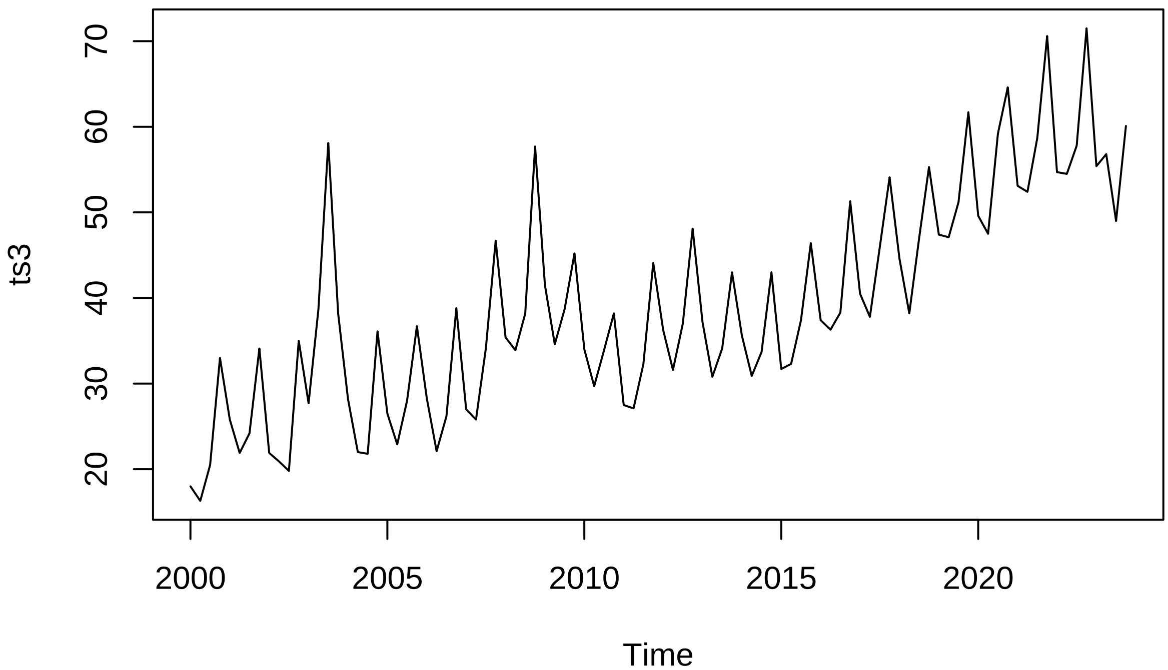
Szereg drugi

```
ts2 <- ts(zb2$`Wskaźnik ogólnego klimatu koniunktury w budownictwie`, start = c(2000, 1), frequency = 4
plot(ts2)
```



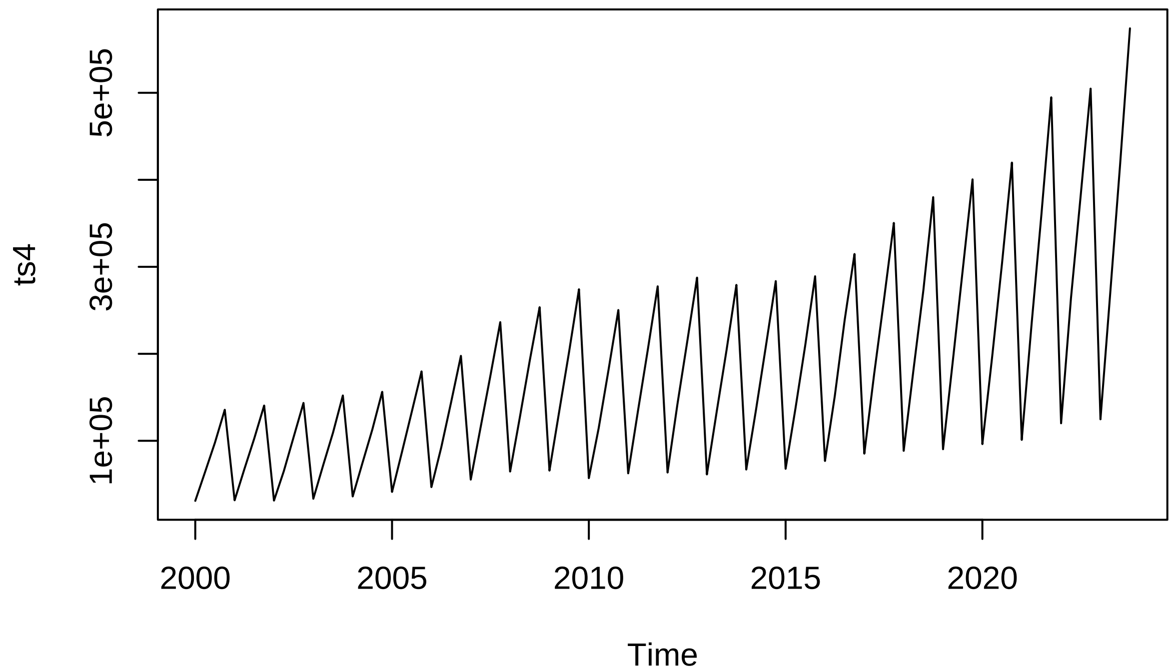
Szereg trzeci

```r
ts3 <- ts(zb3$`Mieszkania oddane do użytkowania w tys`, start = c(2000, 1), frequency = 4)
plot(ts3)
```



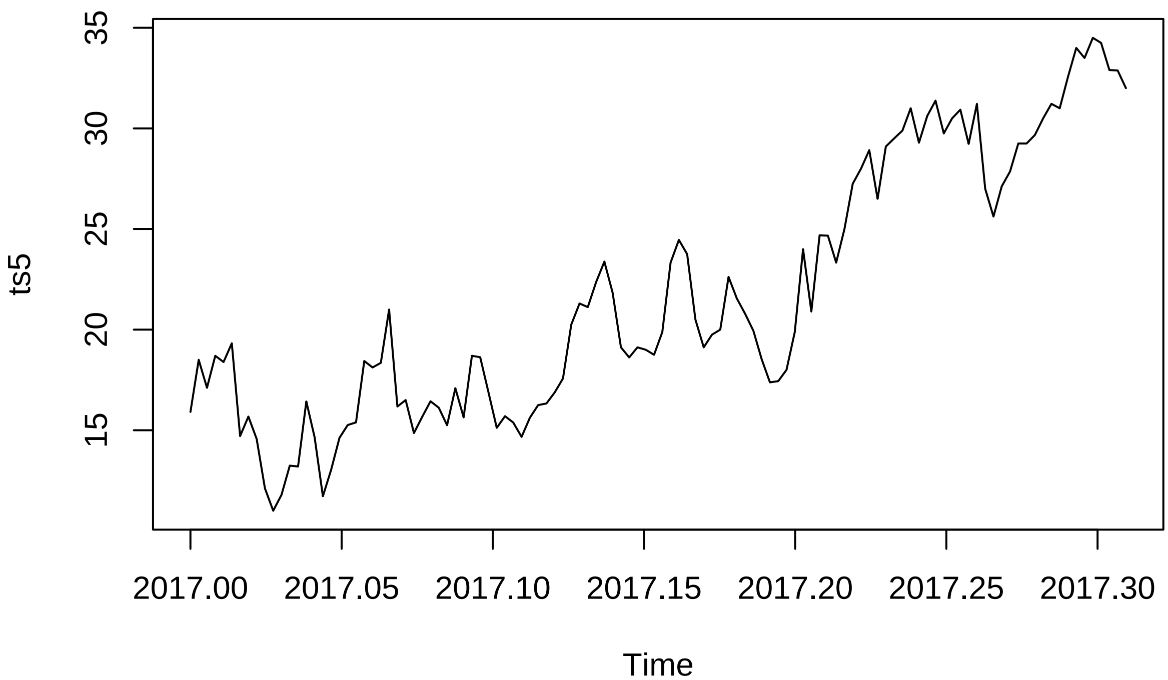
Szereg czwarty

```
ts4 <- ts(zb4$`Dochody budżetu państwa ogółem w mln zł.`, start = c(2000, 1), frequency = 4)
plot(ts4)
```



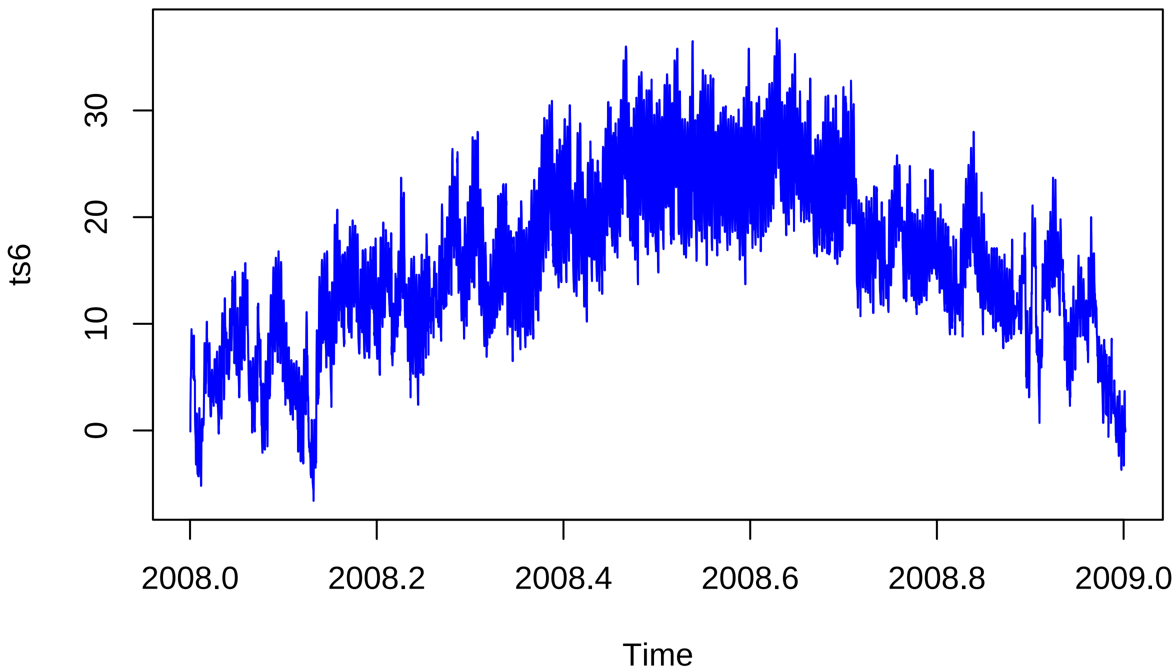
Szereg piąty

```
start_date <- as.Date("2017-01-01")
end_date <- as.Date("2017-04-24")
dates <- seq.Date(from = start_date, to = end_date, by = "day")
ts5 <- ts_ts(xts(zb5$`Średnia temperatura w ciągu dnia.`, order.by = dates))
plot(ts5)
```



Szereg szósty

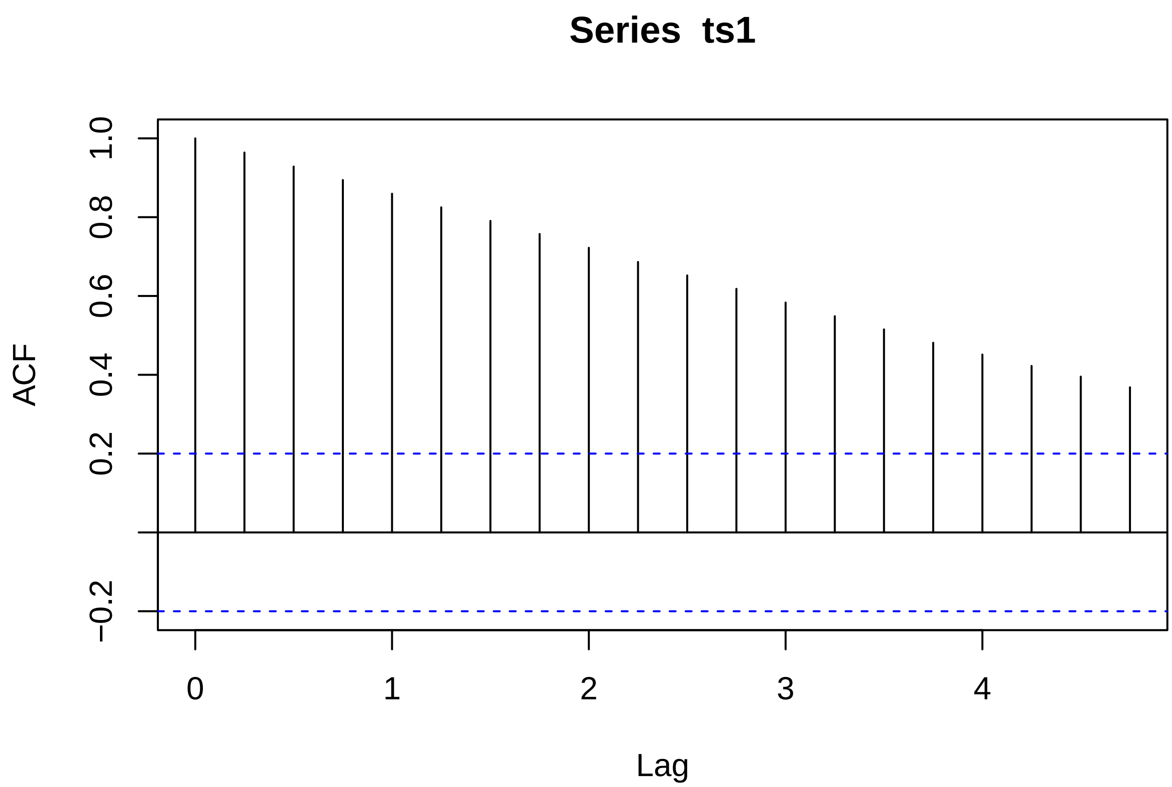
```
start_date2 <- as.POSIXct("2008-01-01 03:00:00")
end_date2 <- as.POSIXct("2008-12-31 23:00:00")
dates2 <- seq(from = start_date2, to = end_date2, by = "hour")
ts6 <- ts_ts(xts(zb6$Temperatura., order.by = dates2))
plot(ts6, col="blue")
```



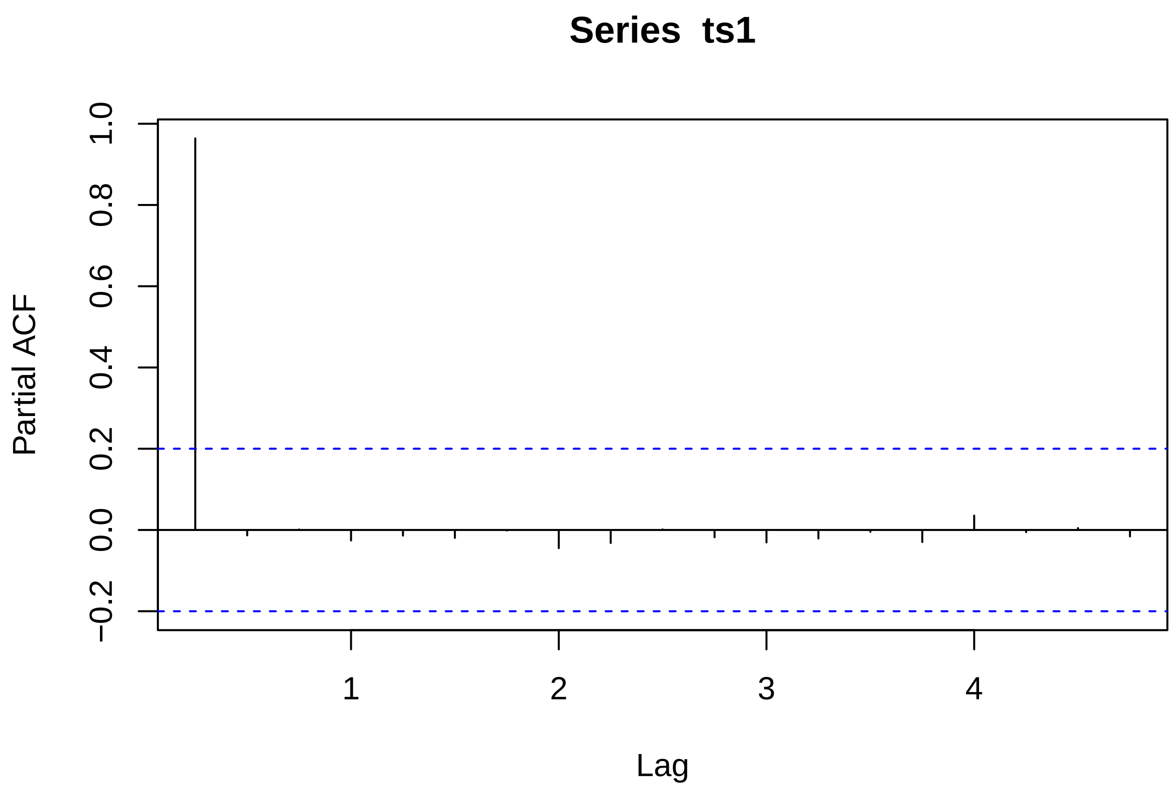
Autokorelacja

Teraz przeprowadzamy testy na autokorelację ACF i PACF oraz Durbina-Watsona dla każdego szeregu czasowego.

```
acf(ts1)
```




```
pacf(ts1)
```

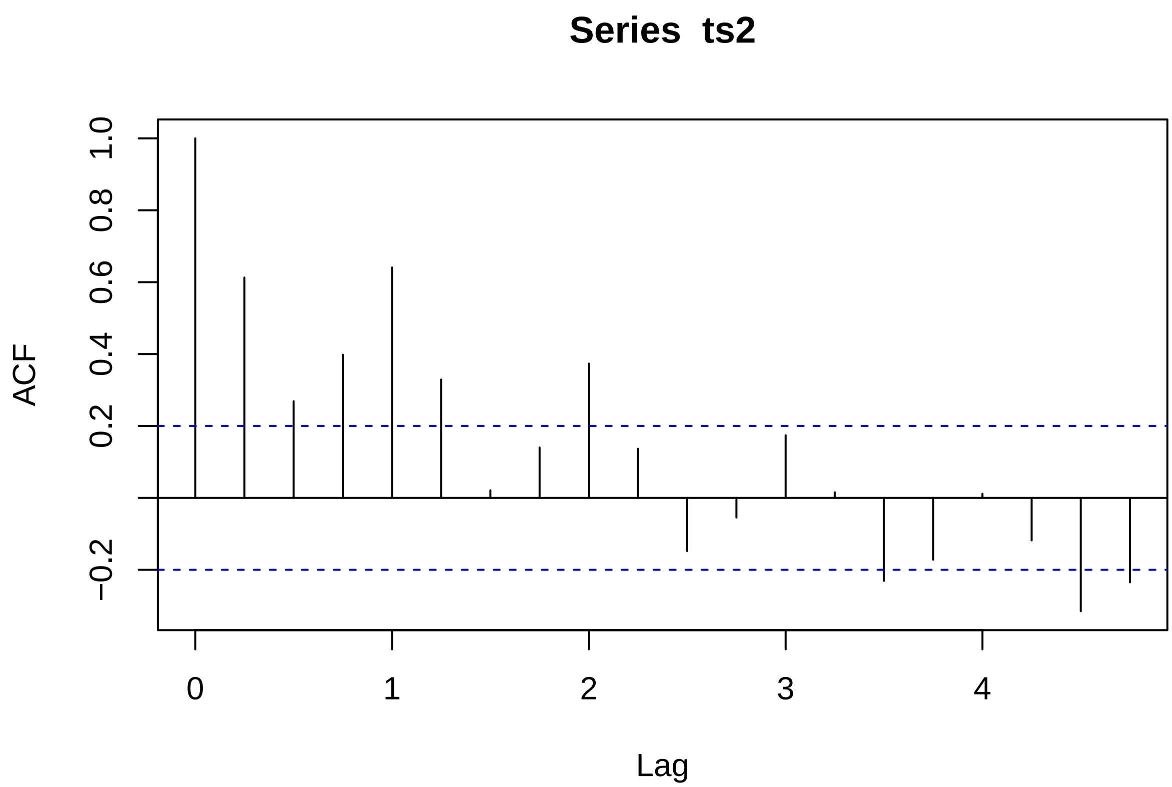



```
df1 = data.frame(time=1:length(ts1),ts1)
model1 <- lm(ts1~time,data = df1)
dwtest(formula = model1, order.by = NULL)
```

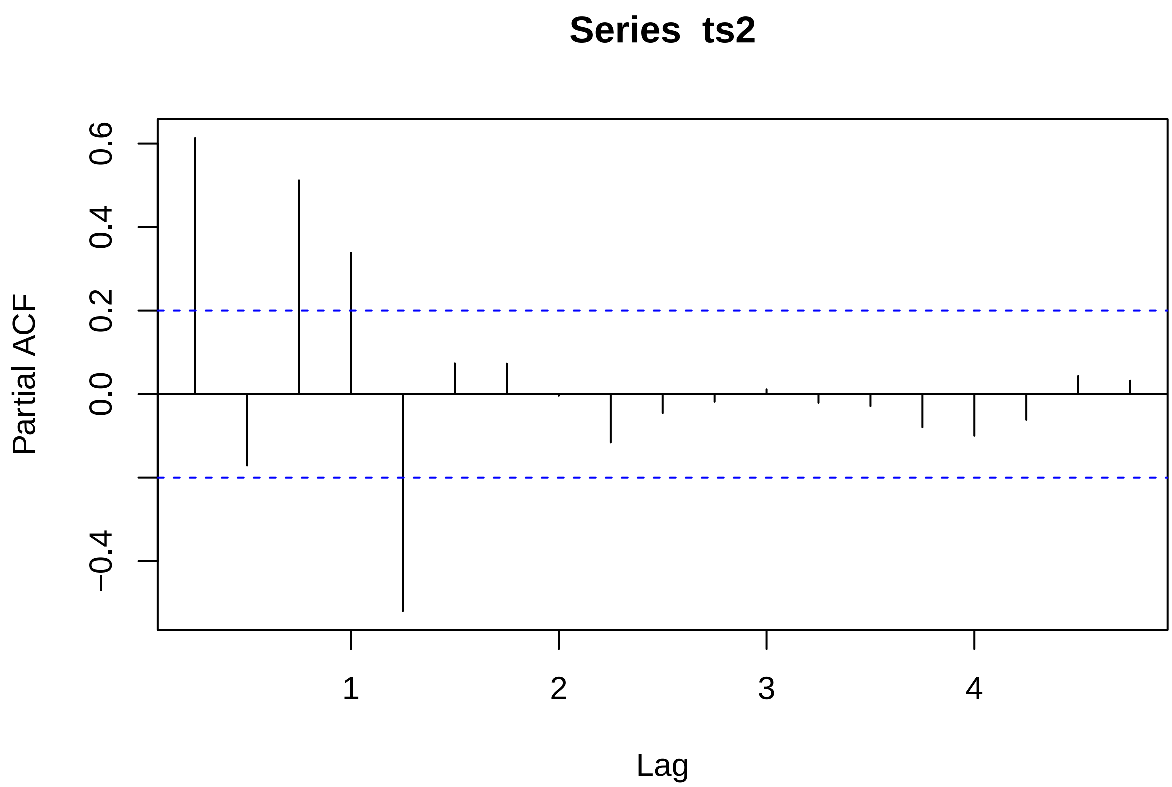
```
##
##  Durbin-Watson test
##
## data:  model1
## DW = 0.023264, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Z przeprowadzonego testu wynika, iż zachodzi autokorelacja.

```
acf(ts2)
```




```
pacf(ts2)
```

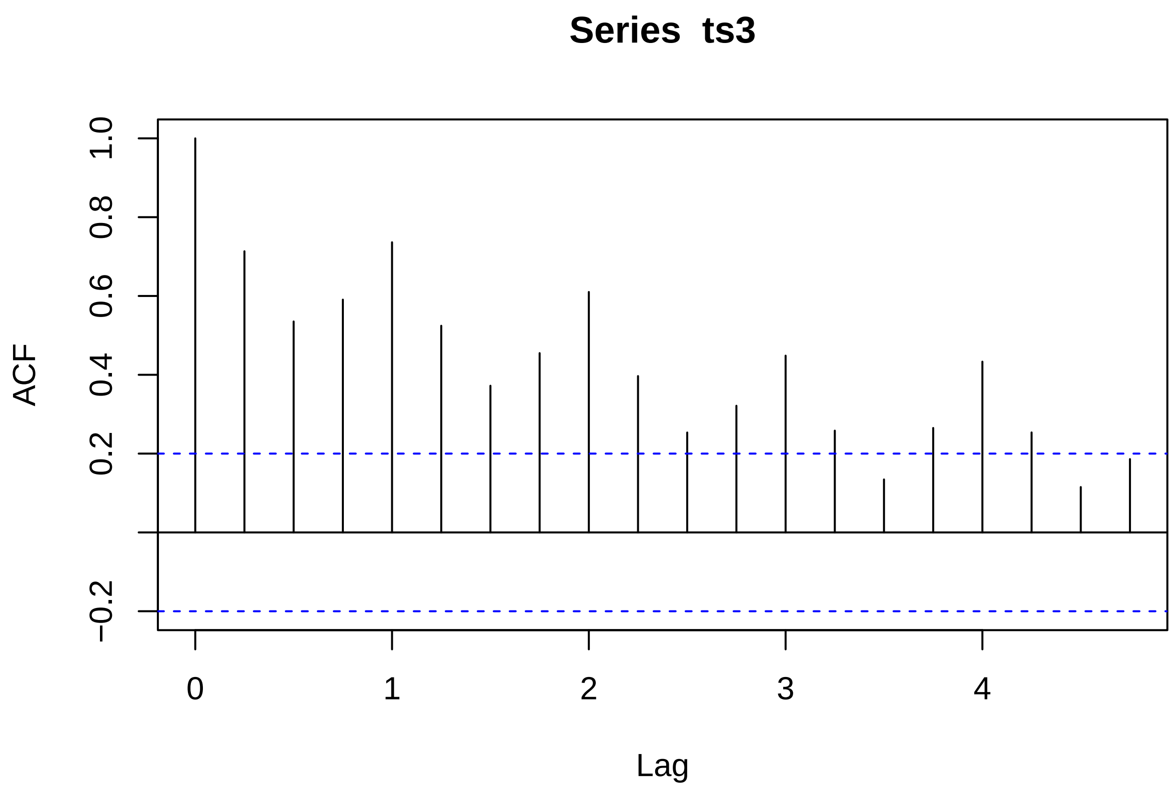



```r
df2 = data.frame(time=1:length(ts2),ts2)
model2 <- lm(ts2~time,data = df2)
dwtest(formula = model2, order.by = NULL)
```

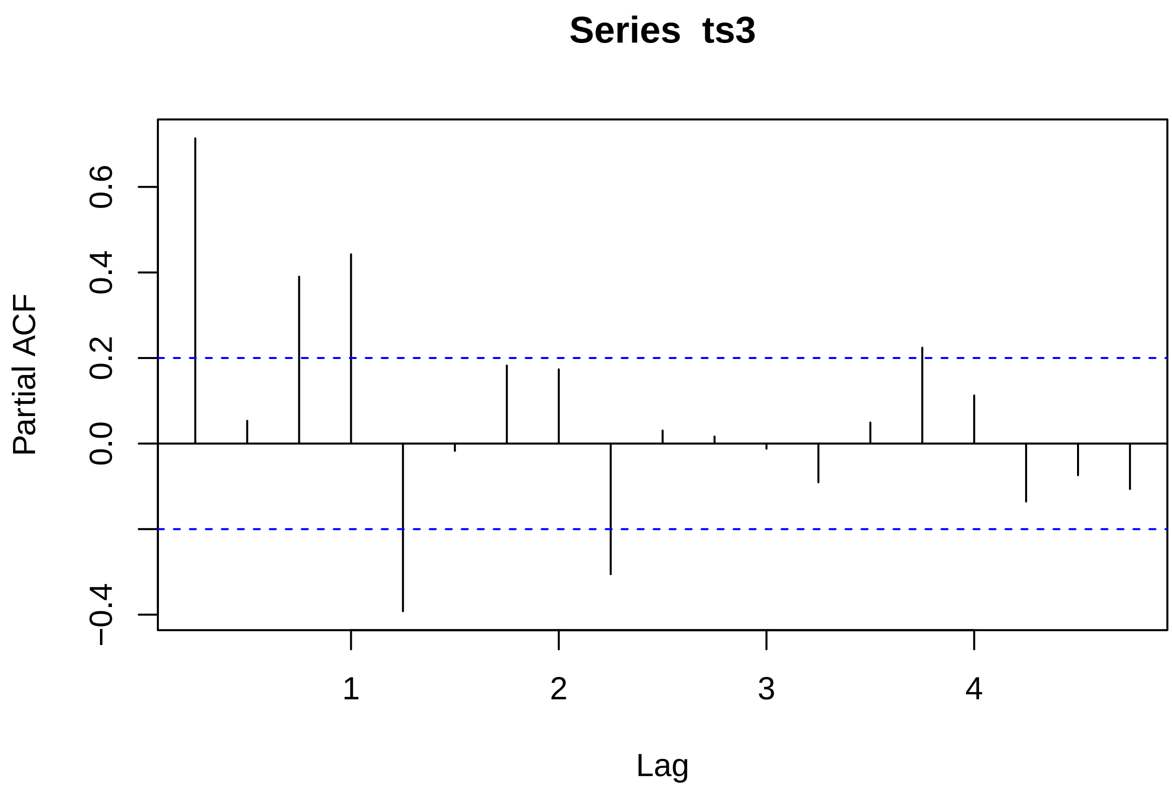
```
##
##  Durbin-Watson test
##
## data:  model2
## DW = 0.83581, p-value = 6.25e-11
## alternative hypothesis: true autocorrelation is greater than 0
```

Z przeprowadzonego testu wynika, iż zachodzi autokorelacja.

```r
acf(ts3)
```




```r
pacf(ts3)
```

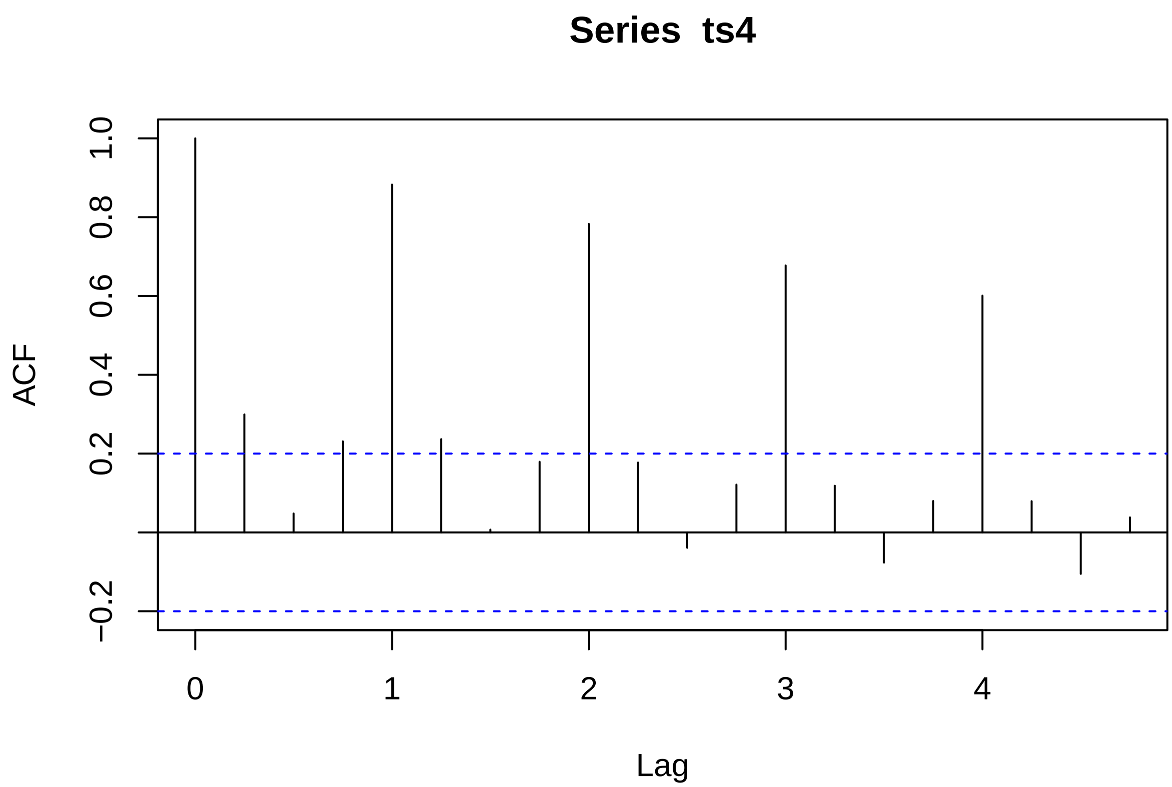



```r
df3 = data.frame(time=1:length(ts3),ts3)
model3 <- lm(ts3~time,data = df3)
dwtest(formula = model3, order.by = NULL)
```

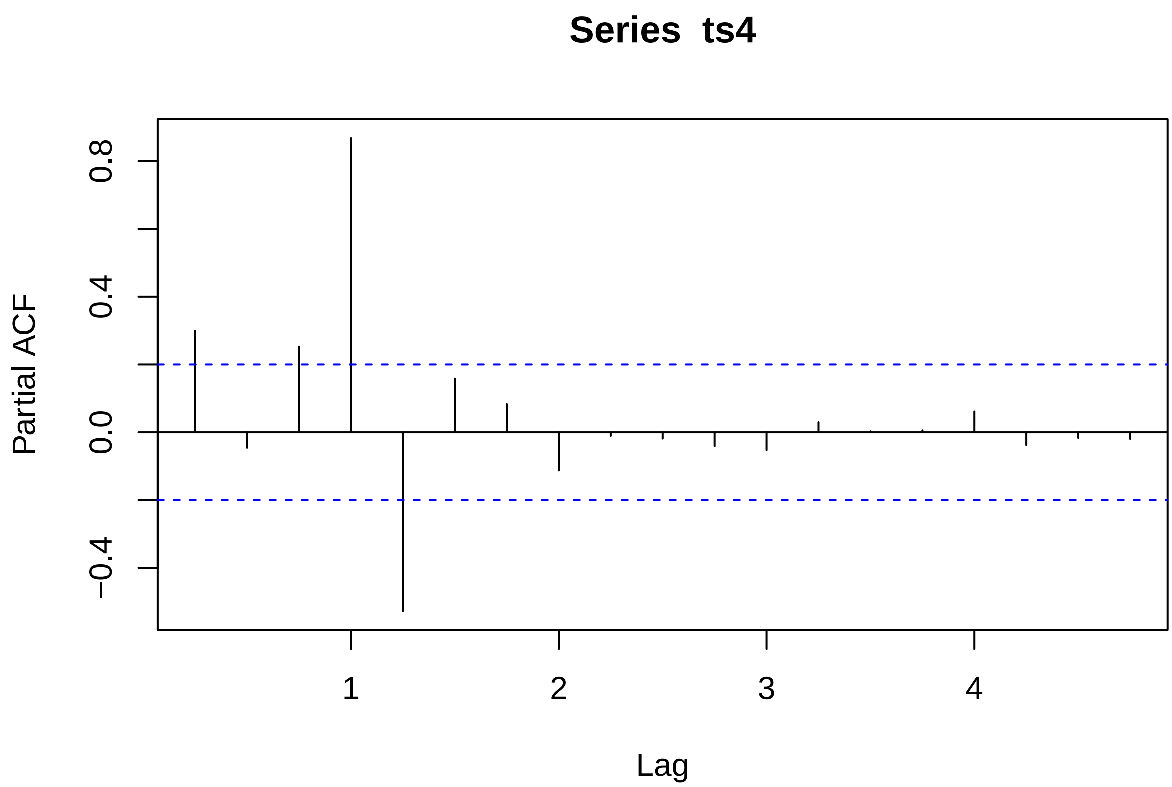
```
##
##  Durbin-Watson test
##
## data:  model3
## DW = 1.3064, p-value = 0.0001443
## alternative hypothesis: true autocorrelation is greater than 0
```

Z przeprowadzonego testu wynika, iż zachodzi autokorelacja.

```r
acf(ts4)
```




```r
pacf(ts4)
```

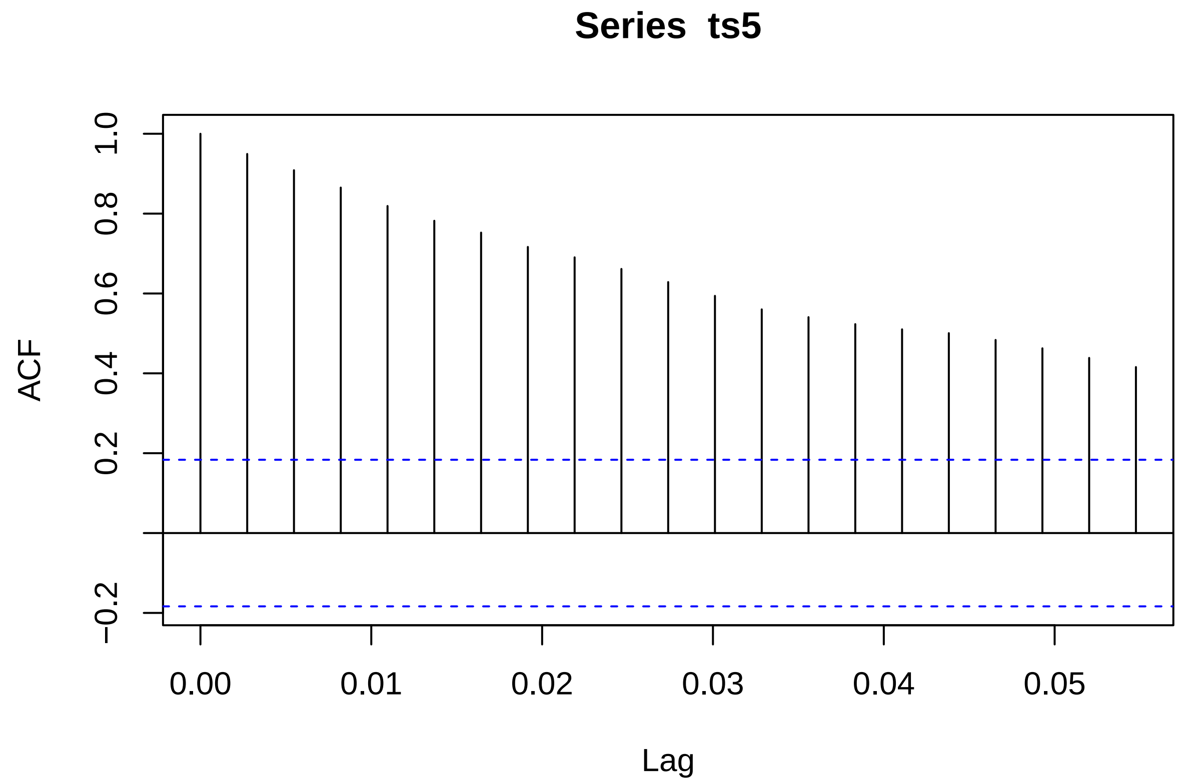



```
df4 = data.frame(time=1:length(ts4),ts4)
model4 <- lm(ts4~time,data = df4)
dwtest(formula = model4, order.by = NULL)
```

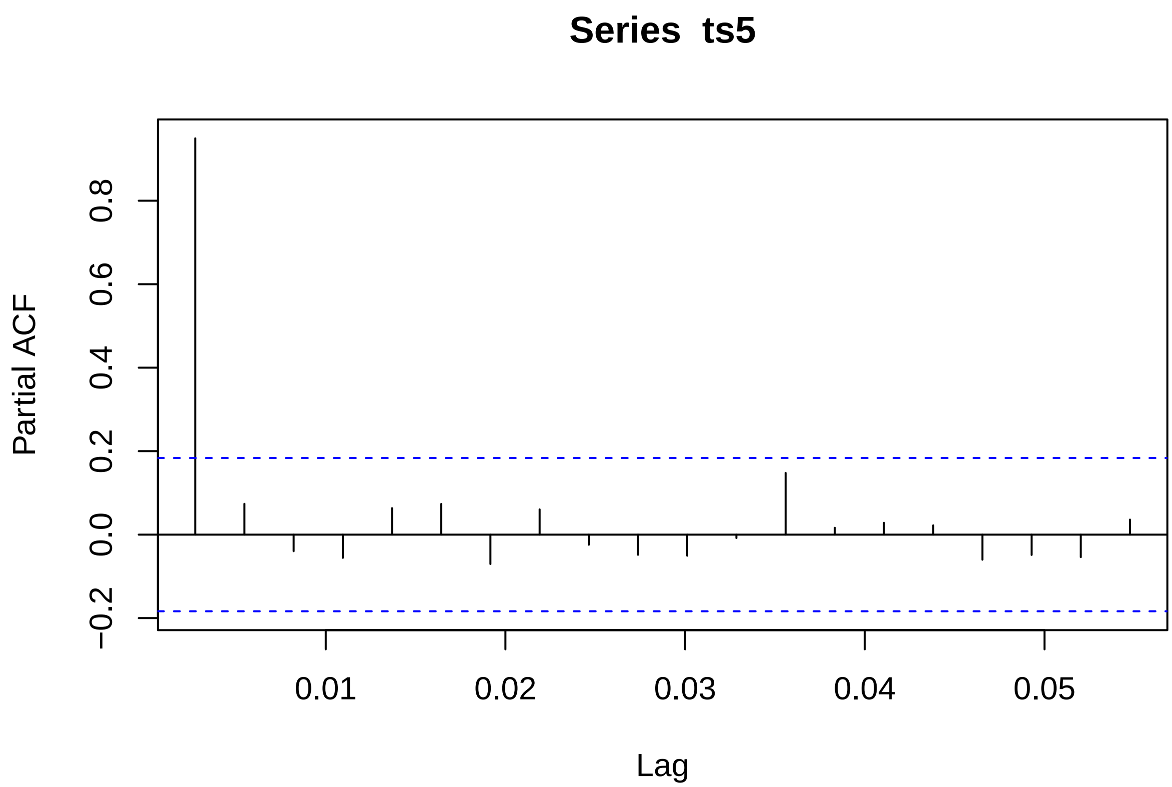
```
##
##  Durbin-Watson test
##
## data:  model4
## DW = 2.1013, p-value = 0.6527
## alternative hypothesis: true autocorrelation is greater than 0
```

Z przeprowadzonego testu wynika, iż zachodzi autokorelacja.

```
acf(ts5)
```




```r
pacf(ts5)
```

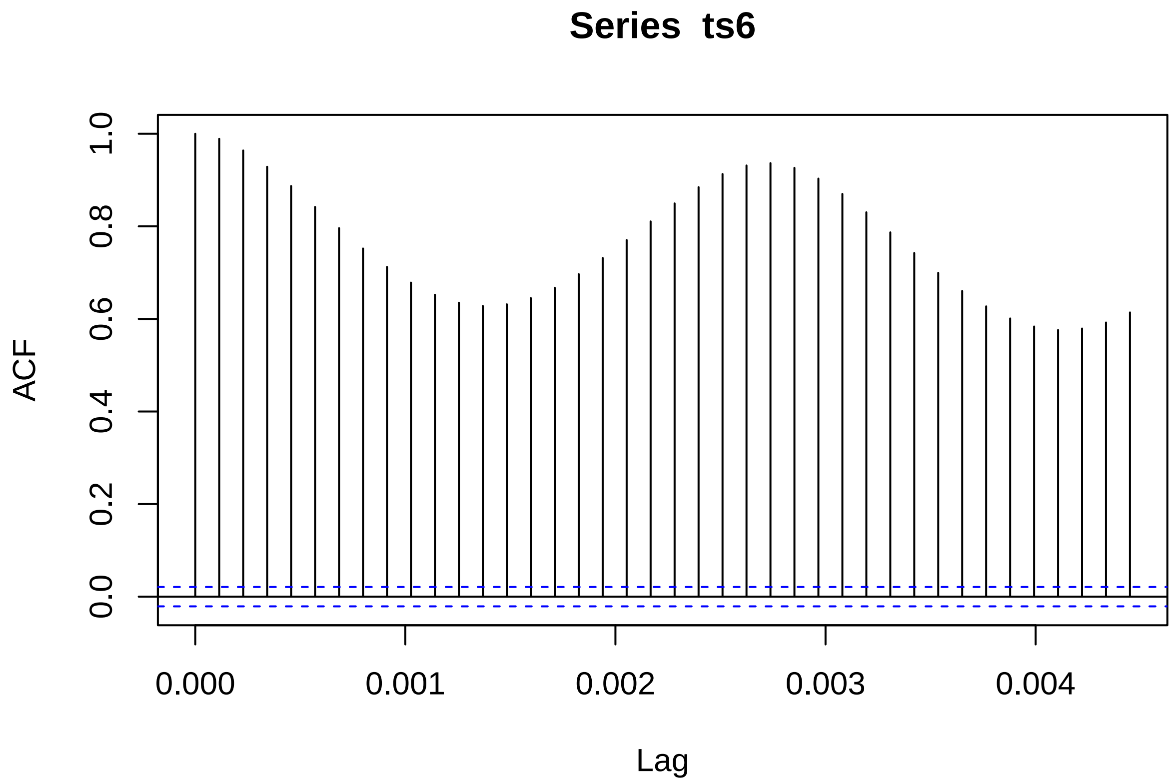



```r
df5 = data.frame(time=1:length(ts5),ts5)
model5 <- lm(ts5~time,data = df5)
dwtest(formula = model5, order.by = NULL)
```

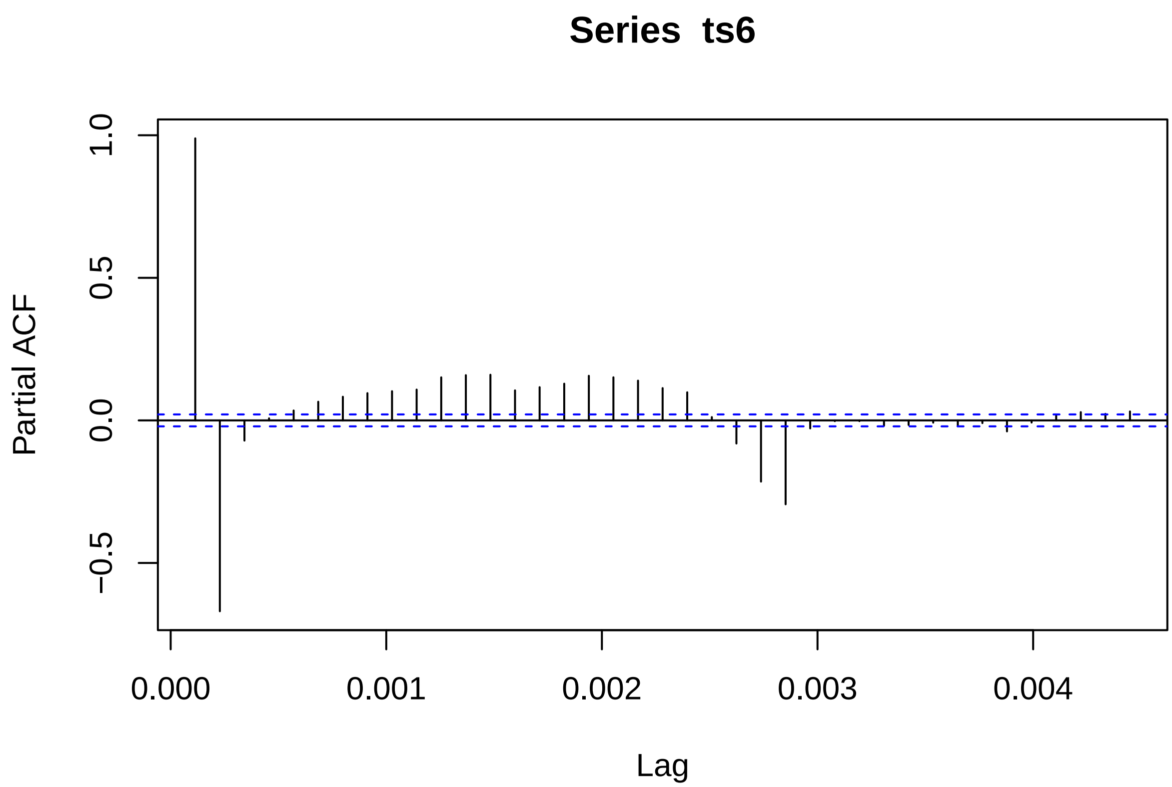
```
##
##  Durbin-Watson test
##
## data:  model5
## DW = 0.3574, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Z przeprowadzonego testu wynika, iż zachodzi autokorelacja.

```r
acf(ts6)
```




```r
pacf(ts6)
```




```r
df6 = data.frame(time=1:length(ts6),ts6)
model6 <- lm(ts6~time,data = df6)
dwtest(formula = model6, order.by = NULL)
```

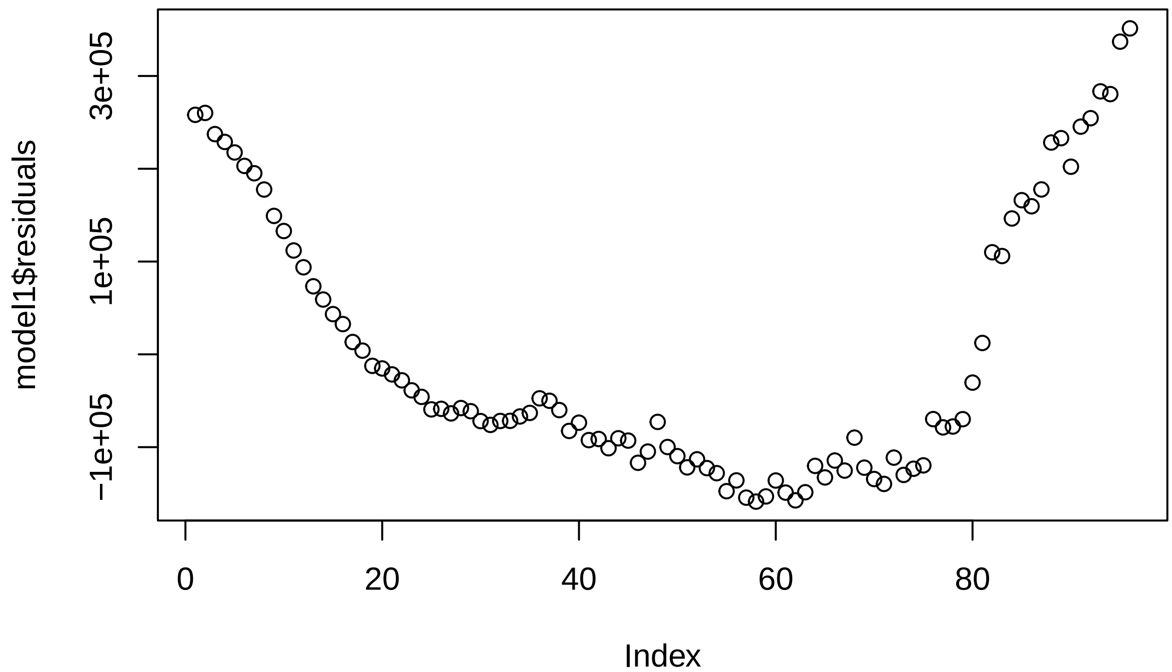
```
##
##  Durbin-Watson test
##
## data:  model6
## DW = 0.022242, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Z przeprowadzonego testu wynika, iż zachodzi autokorelacja.

Heteroskedastyczność

Teraz przeprowadzamy testy na heteroskedastyczność dla wszystkich siedmiu szeregów czasowych.

```r
plot(model1$residuals)
```

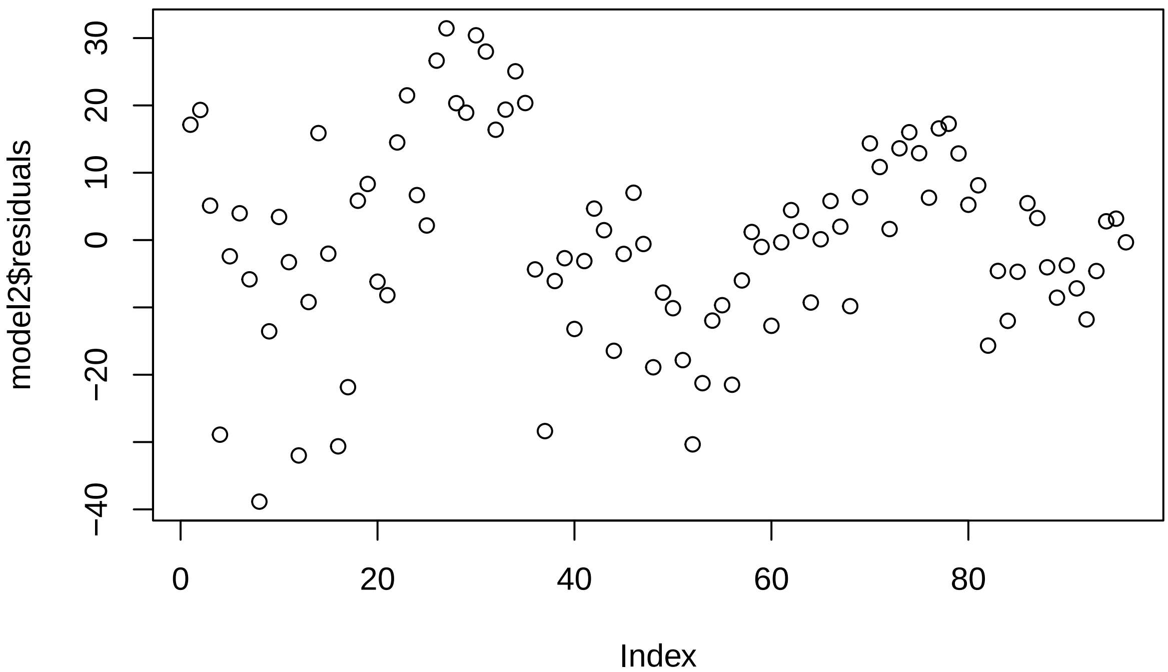



```r
bptest(model1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 8.8686, df = 1, p-value = 0.002901
```

Po przeprowadzeniu testu Breusch-Pagan'a, ponieważ p-value jest mniejsze od 0.05 odrzucamy hipotezę zerową i możemy stwierdzić, że występuje heteroskedastyczność.

```r
plot(model2$residuals)
```

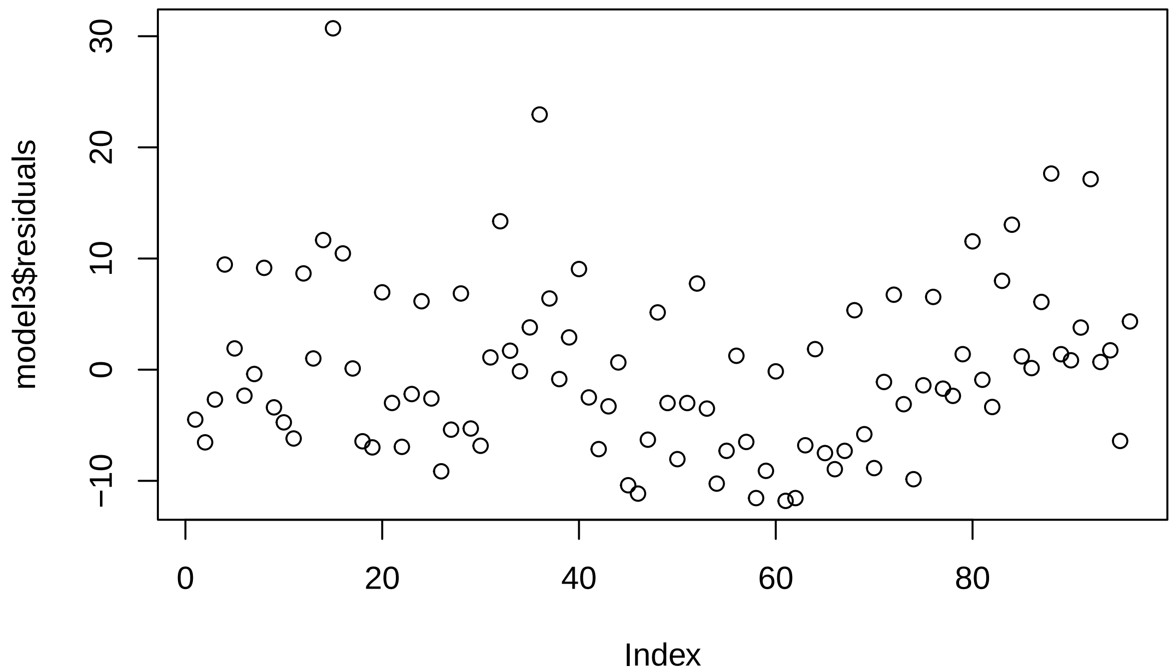



```r
bptest(model2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 12.71, df = 1, p-value = 0.0003636
```

Po przeprowadzeniu testu Breusch-Pagan'a, ponieważ p-value jest mniejsze od 0.05 odrzucamy hipotezę zerową i możemy stwierdzić, że występuje heteroskedastyczność.

```r
plot(model3$residuals)
```




```r
bptest(model3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model3
## BP = 0.30476, df = 1, p-value = 0.5809
```

Po przeprowadzeniu testu Breusch-Pagan'a, ponieważ p-value jest większe od 0.05 nie możemy odrzucić hipotezy zerowej i nie możemy stwierdzić, że występuje heteroskedastyczność. Musimy przeprowadzić kolejne testy.

```r
white(mainlm = model3, interactions = FALSE, statonly = FALSE)
```

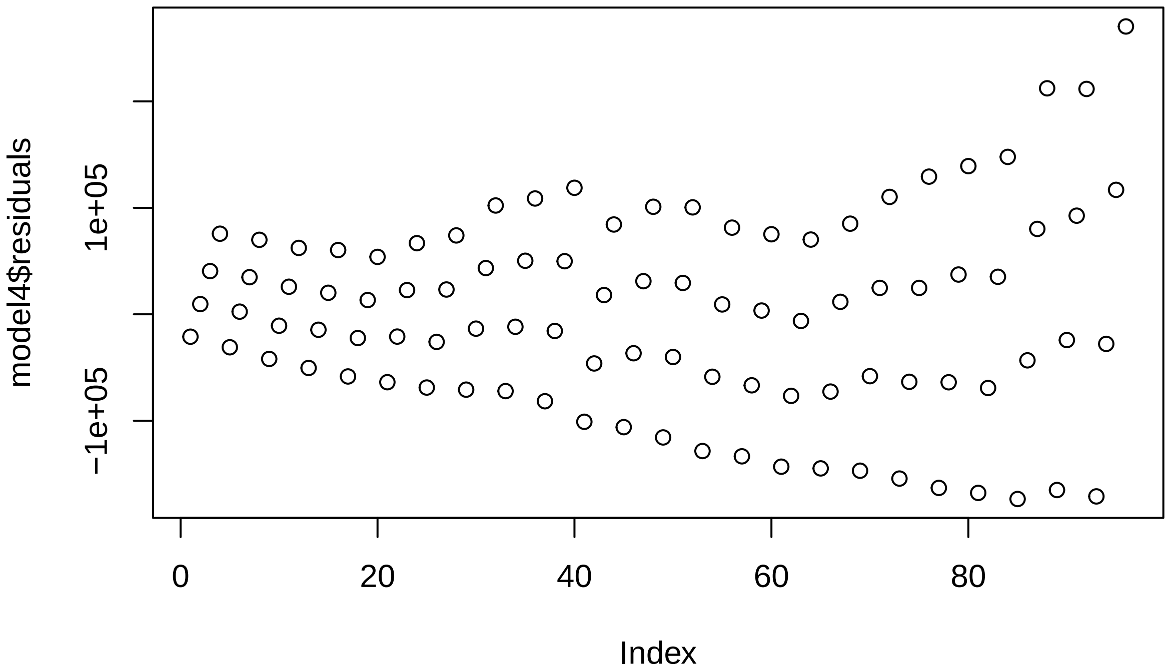
```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1     0.308   0.857         2 White's Test greater
```

```r
goldfeld_quandt(mainlm = model3, method = "parametric", deflator = NA, prop_central = 1/3, group1prop =
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method                   alternative
##       <dbl>   <dbl>     <int> <chr>                    <chr>
## 1     0.567   0.126        30 Goldfeld-Quandt F Test two.sided
```

Po przeprowadzeniu testu White'a i testu Goldfeld-Quandt'a dalej p-value jest większe od 0.05, więc możemy stwierdzić, że nie występuje heteroskedastyczność.

```
plot(model4$residuals)
```

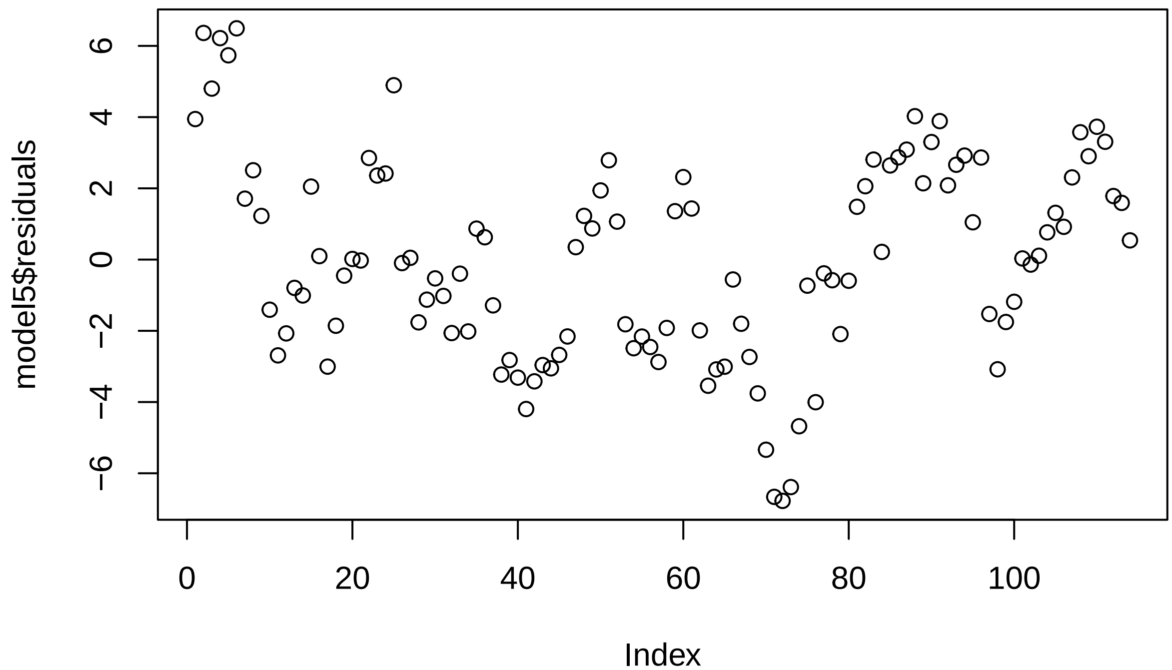



```
bptest(model4)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model4
## BP = 26.246, df = 1, p-value = 3.006e-07
```

Po przeprowadzeniu testu Breusch-Pagan'a, ponieważ p-value jest mniejsze od 0.05 odrzucamy hipotezę zerową i możemy stwierdzić, że występuje heteroskedastyczność.

```
plot(model5$residuals)
```




```r
bptest(model5)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model5
## BP = 1.7705, df = 1, p-value = 0.1833
```

Po przeprowadzeniu testu Breusch-Pagan'a, ponieważ p-value jest większe od 0.05 nie możemy odrzucić hipotezy zerowej i nie możemy stwierdzić, że występuje heteroskedastyczność. Musimy przeprowadzić kolejne testy.

```r
white(mainlm = model5, interactions = FALSE, statonly = FALSE)
```

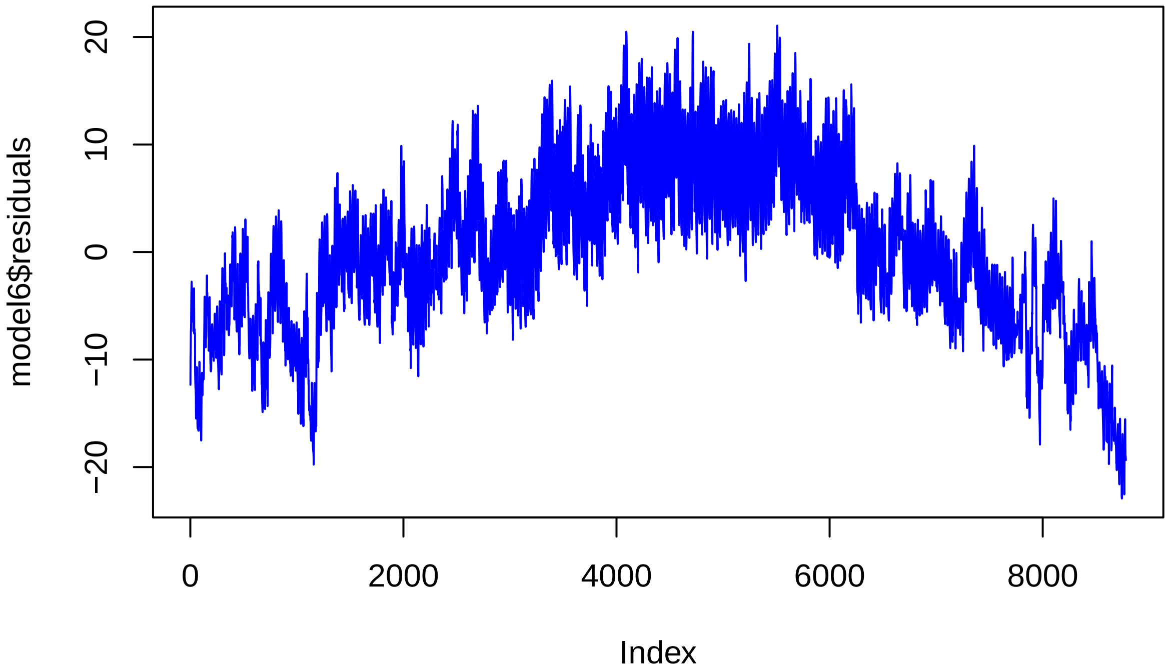
```
## # A tibble: 1 x 5
##   statistic p.value parameter method       alternative
##       <dbl>   <dbl>     <dbl> <chr>        <chr>
## 1      2.91   0.233         2 White's Test greater
```

```r
goldfeld_quandt(mainlm = model5, method = "parametric", deflator = NA, prop_central = 1/3, group1prop =
```

```
## # A tibble: 1 x 5
##   statistic p.value parameter method                   alternative
##       <dbl>   <dbl>     <int> <chr>                    <chr>
## 1     0.641   0.187        36 Goldfeld-Quandt F Test two.sided
```

Po przeprowadzeniu testu White'a i testu Goldfeld-Quandt'a dalej p-value jest większe od 0.05, więc możemy stwierdzić, że nie występuje heteroskedastyczność.

```r
plot(model6$residuals, type = "l", col="blue")
```




```r
bptest(model6)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model6
## BP = 146.36, df = 1, p-value < 2.2e-16
```

Po przeprowadzeniu testu Breusch-Pagan'a, ponieważ p-value jest mniejsze od 0.05 odrzucamy hipotezę zerową i możemy stwierdzić, że występuje heteroskedastyczność.

Stacjonarność

Teraz przeprowadzamy test na stacjonarność naszych szeregów.

```r
urca::ur.kpss(ts1) %>% summary()
```

```
##
```

```
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 2.3684
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
urca::ur.kpss(ts2) %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.5768
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
urca::ur.kpss(ts3) %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 1.9773
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
urca::ur.kpss(ts4) %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 2.2983
```

```
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
urca::ur.kpss(ts5) %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 2.0514
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
urca::ur.kpss(ts6) %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 12 lags.
##
## Value of test-statistic is: 16.9625
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Z przeprowadzonego testu KPSS wychodzi, iż pierwsze sześć szeregów czasowych jest niestacjonarne, a ostatni siódmy szereg jest stacjonarny.

Tworzenie modeli

Na podstawie naszych wcześniejszych testów stworzymy teraz model predykcji dla każdego szeregu czasowego.

Model 1

Jako, że pierwszy szereg czasowy wyszedł niestacjonarny musimy go zróżnicować.

```
diff(ts1, lag = 4) %>% urca::ur.kpss() %>% summary()
```

```
##
## #######################
```

```
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 1.6481
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Jako, że zróżnicowanie szeregu nie przyniosło oczekiwanych rezultatów (szereg dalej jest niestacjonarny). Spróbujemy pozbyć się trendu.

```
model1$residuals %>% urca::ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.5324
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Pozbycie się trendu nie spowodowało, że nasz szereg jest stacjonarny. Szereg nie jest możliwy do sprowadzenia do postaci stacjonarnej.

```
ts1arima = auto.arima(ts1, seasonal = TRUE)
ts1arima
```

```
## Series: ts1
## ARIMA(0,2,1)(1,0,1)[4]
##
## Coefficients:
##           ma1    sar1     sma1
##       -0.9307  0.8821  -0.6707
## s.e.   0.0426  0.1226   0.2207
##
## sigma^2 = 328205035:  log likelihood = -1054.73
## AIC=2117.46   AICc=2117.91   BIC=2127.64
```

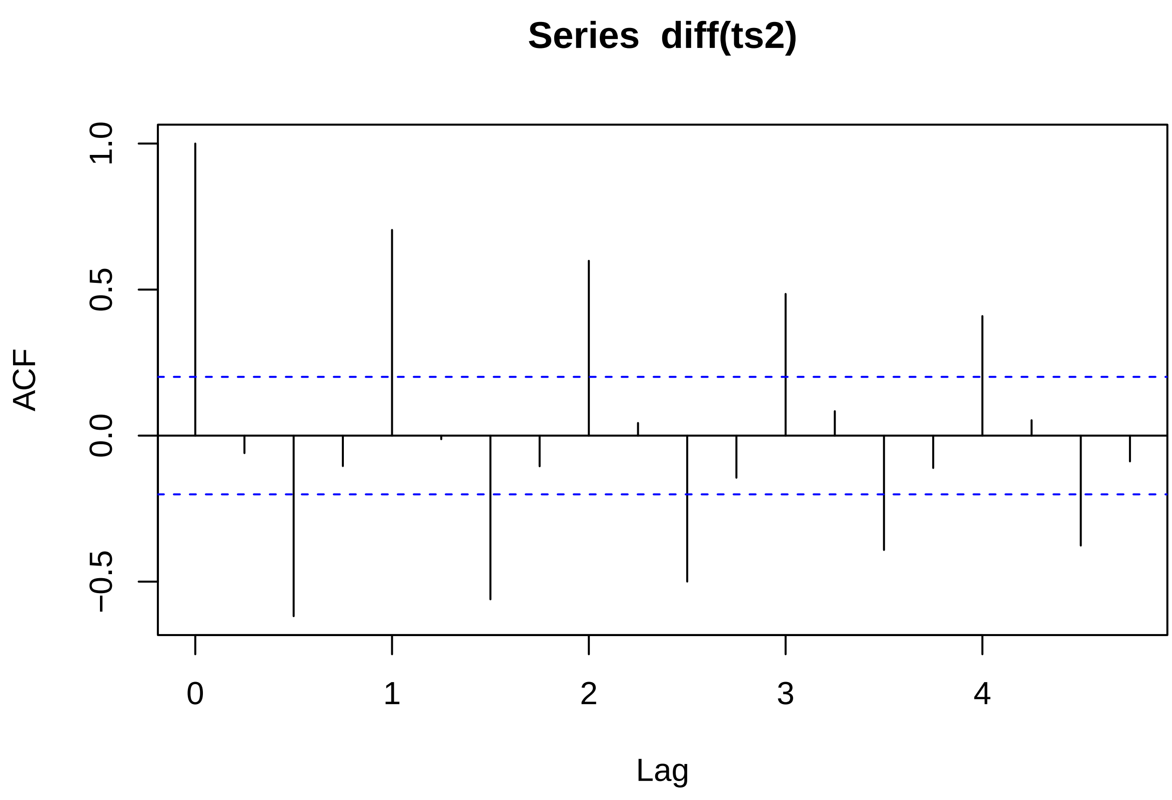
Model 2

Jako, że drugi szereg czasowy też wyszedł niestacjonarny musimy go zróżnicować.

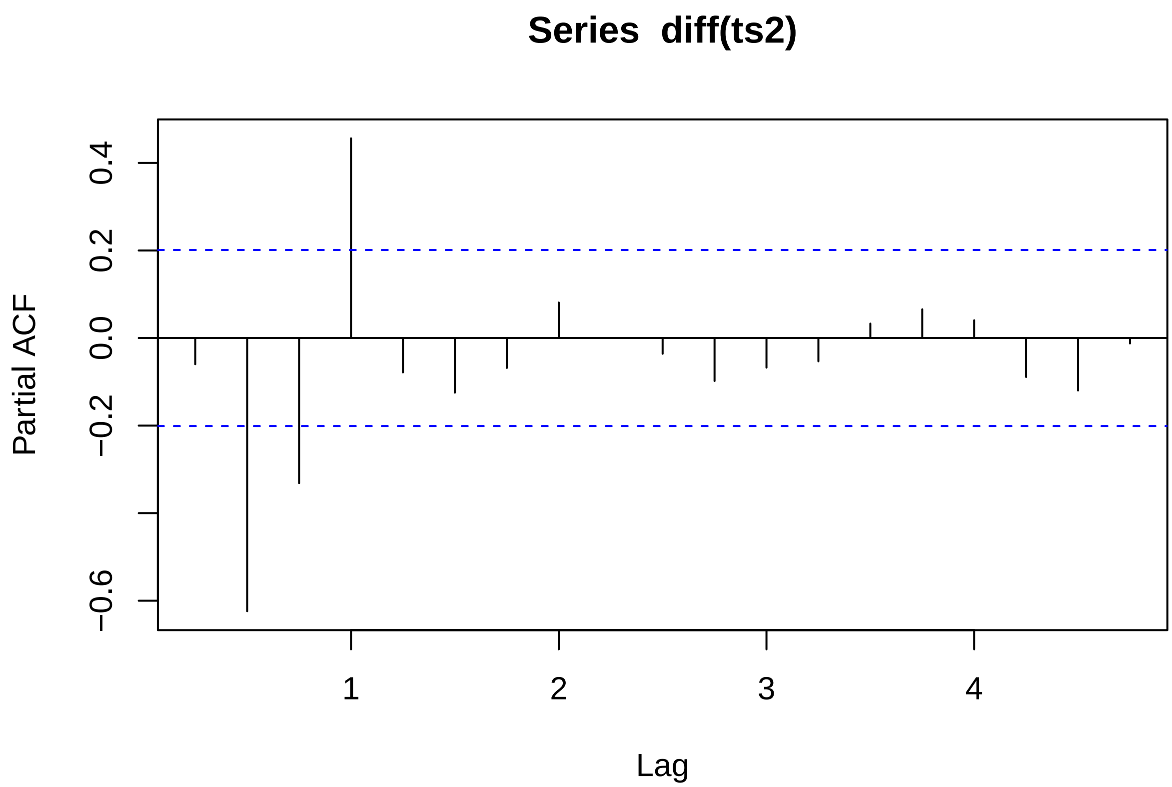
```r
diff(ts2) %>% urca::ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0765
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
acf(diff(ts2))
```




```r
pacf(diff(ts2))
```



Z testu ACF wychodzi nam, iż q może wychodzić 1. Z testu PACF wychodzi, iż p jest też równe 1. Szereg czasowy wydaje się być sezonowy.

```
Arima(y=ts2, order = c(1,1,1), lambda = NULL)
```

```
## Series: ts2
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.3939  -0.8014
## s.e.  0.1868   0.1346
##
## sigma^2 = 152.6:  log likelihood = -372.82
## AIC=751.64   AICc=751.91   BIC=759.3
```

```
Arima(y=ts2, seasonal = c(1,1,1), lambda = NULL)
```

```
## Series: ts2
## ARIMA(0,0,0)(1,1,1)[4]
##
## Coefficients:
##         sar1     sma1
##       0.6558  -0.9418
## s.e.  0.1167   0.1053
```

```
##
## sigma^2 = 136.4:  log likelihood = -357.22
## AIC=720.45   AICc=720.72   BIC=728.01
```

```
ts2arima = Arima(y=ts2, seasonal = c(1,1,1), lambda = "auto")
ts2arima
```

```
## Series: ts2
## ARIMA(0,0,0)(1,1,1)[4]
## Box Cox transformation: lambda= 0.7573308
##
## Coefficients:
##         sar1     sma1
##       0.6775  -0.9385
## s.e.  0.1161   0.1009
##
## sigma^2 = 52.43:  log likelihood = -313.05
## AIC=632.11   AICc=632.38   BIC=639.67
```

```
auto.arima(ts2, seasonal = TRUE)
```

```
## Series: ts2
## ARIMA(1,0,0)(0,1,1)[4]
##
## Coefficients:
##          ar1     sma1
##       0.7864  -0.4171
## s.e.  0.0680   0.1089
##
## sigma^2 = 65.53:  log likelihood = -322.62
## AIC=651.24   AICc=651.51   BIC=658.8
```

Model ts2arima okazał się najlepszy spośród testowany modeli.

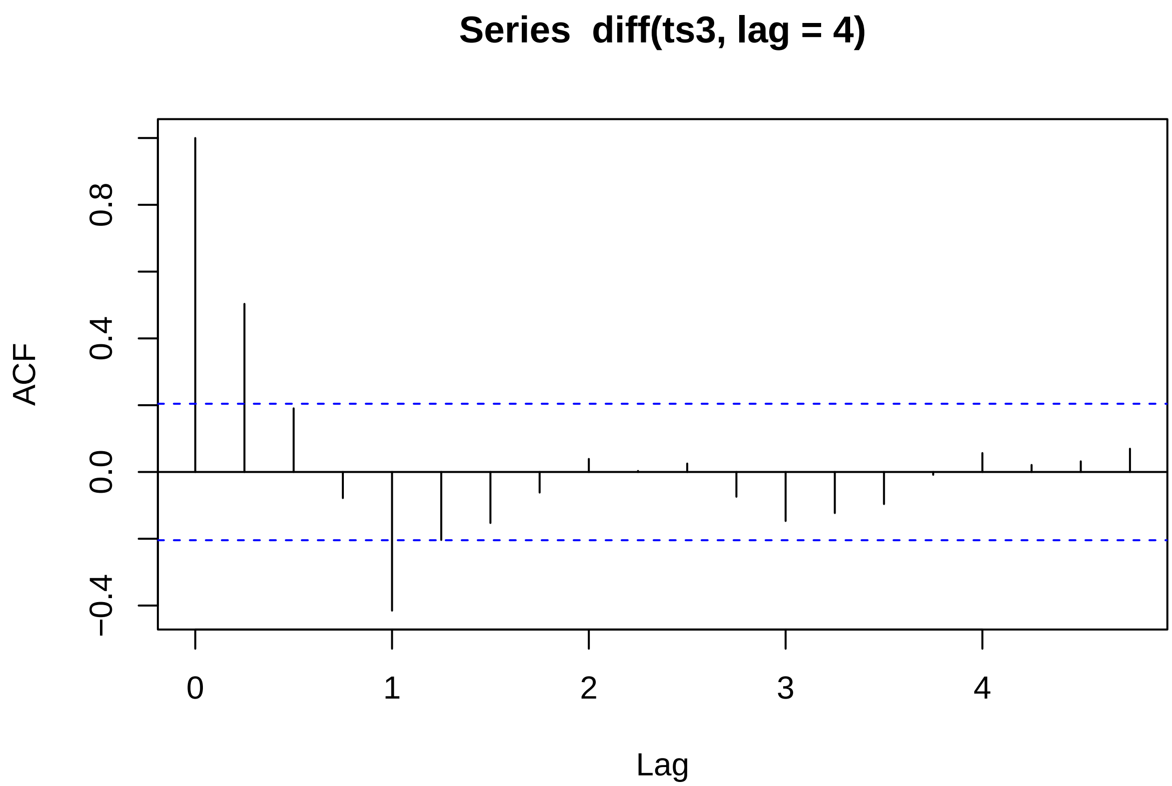
Model 3

Jako, że trzeci szereg czasowy również wyszedł niestacjonarny musimy go zróżnicować.

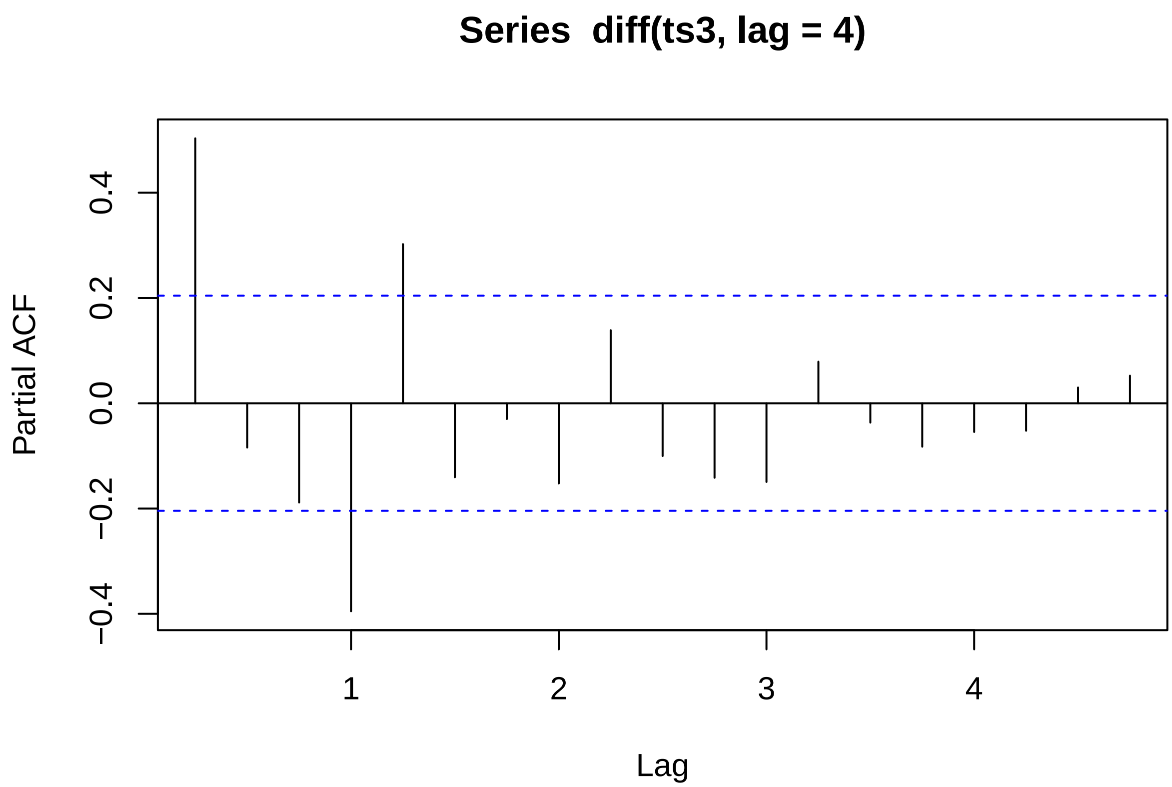
```
diff(ts3) %>% urca::ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0494
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
acf(diff(ts3, lag = 4))
```




```r
pacf(diff(ts3, lag = 4))
```



Z testu ACF wychodzi nam, iż q może wychodzić 1. Z testu PACF wychodzi, iż p jest też równe 1. Szereg czasowy wydaje się być sezonowy.

```
Arima(y=ts3, order = c(1,1,1), lambda = NULL)
```

```
## Series: ts3
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.2263  -0.8064
## s.e.  0.1253   0.0670
##
## sigma^2 = 60.41:  log likelihood = -328.94
## AIC=663.89   AICc=664.15   BIC=671.55
```

```
Arima(y=ts3, seasonal = c(1,1,1), lambda = NULL)
```

```
## Series: ts3
## ARIMA(0,0,0)(1,1,1)[4]
##
## Coefficients:
##          sar1     sma1
##       -0.0984  -0.3160
## s.e.   0.2529   0.2429
```

```
##
## sigma^2 = 50.79:  log likelihood = -310.56
## AIC=627.12   AICc=627.4   BIC=634.69
```

```r
ts3arima = Arima(y=ts3, seasonal = c(1,1,1), lambda = "auto")
ts3arima
```

```
## Series: ts3
## ARIMA(0,0,0)(1,1,1)[4]
## Box Cox transformation: lambda= 0.816127
##
## Coefficients:
##          sar1     sma1
##       -0.1235  -0.3096
## s.e.   0.2493   0.2426
##
## sigma^2 = 13.24:  log likelihood = -248.75
## AIC=503.5   AICc=503.77   BIC=511.06
```

```r
auto.arima(ts3, seasonal = TRUE)
```

```
## Series: ts3
## ARIMA(1,0,0)(2,1,0)[4] with drift
##
## Coefficients:
##          ar1     sar1     sar2   drift
##       0.6416  -0.6915  -0.2558  0.3618
## s.e.  0.0842   0.1046   0.1013  0.1971
##
## sigma^2 = 28.62:  log likelihood = -283.92
## AIC=577.84   AICc=578.54   BIC=590.45
```

Model ts3arima okazał się najlepszy spośród testowany modeli.

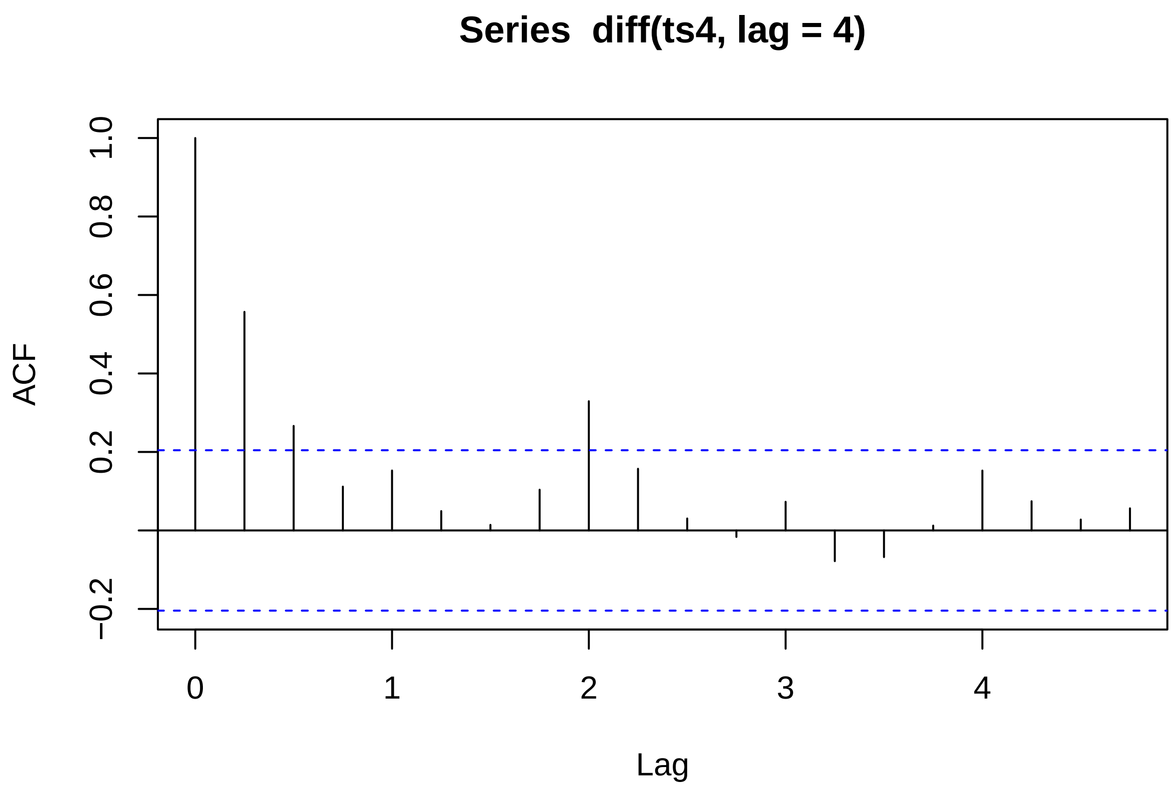
Model 4

Jako, że czwarty szereg czasowy wyszedł niestacjonarny musimy go zróżnicować.

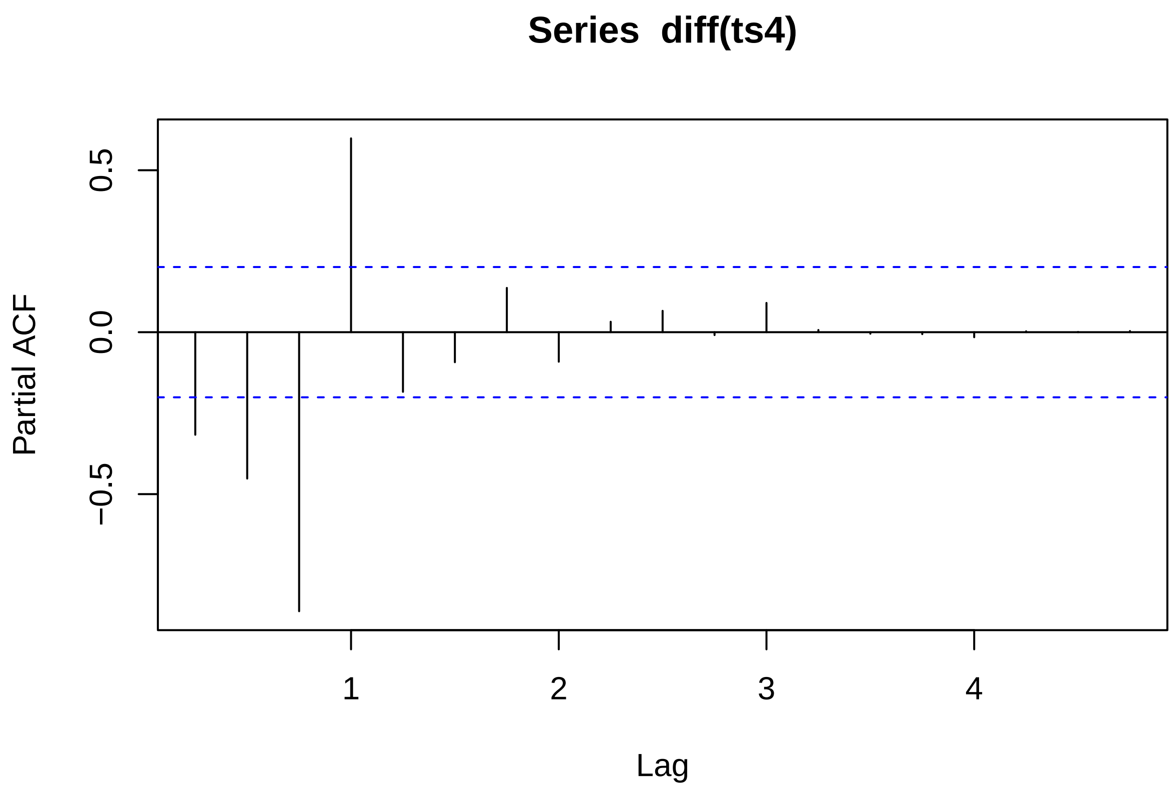
```r
diff(ts4) %>% urca::ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.3658
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
acf(diff(ts4, lag = 4))
```




```r
pacf(diff(ts4))
```



Z testu ACF wychodzi nam, iż q może wychodzić 2. Z testu PACF wychodzi, iż p jest równe 1. Szereg czasowy wydaje się być sezonowy.

```
Arima(y=ts4, order = c(1,1,2), lambda = NULL)
```

```
## Series: ts4
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1     ma2
##       0.1239  -1.7359  0.9228
## s.e.  0.1210   0.0548  0.0466
##
## sigma^2 = 5.938e+09:  log likelihood = -1204.78
## AIC=2417.56   AICc=2418.01   BIC=2427.78
```

```
Arima(y=ts4, order = c(8,2,8), lambda = NULL)
```

```
## Series: ts4
## ARIMA(8,2,8)
##
## Coefficients:
##          ar1      ar2     ar3     ar4      ar5     ar6     ar7     ar8      ma1
##       0.3753  -0.4572  0.2688  0.1428  -0.6865  0.1474  -0.581  0.5435  -1.5837
## s.e.     NaN      NaN     NaN     NaN   0.1345     NaN     NaN     NaN      NaN
```

```
##         ma2      ma3     ma4      ma5     ma6      ma7     ma8
##       0.918  -1.2561  1.9789  -0.9017  0.0998  -0.5694  0.3912
## s.e.    NaN   0.0904     NaN      NaN  0.3005   0.3485     NaN
##
## sigma^2 = 141435739:  log likelihood = -1017.93
## AIC=2069.86   AICc=2077.91   BIC=2113.1
```

```
Arima(y=ts4, seasonal = c(4,2,4), lambda = NULL)
```

```
## Series: ts4
## ARIMA(0,0,0)(4,2,4)[4]
##
## Coefficients:
##         sar1     sar2     sar3    sar4     sma1    sma2     sma3     sma4
##       0.7441  -0.4061  -0.2430  0.4893  -1.7460  1.6994  -0.7384  -0.0560
## s.e.  0.2888   0.2193   0.2214  0.1803   0.2928  0.4580   0.4858   0.1999
##
## sigma^2 = 172880011:  log likelihood = -964.18
## AIC=1946.36   AICc=1948.67   BIC=1968.66
```

```
ts4arima = Arima(y=ts4, seasonal = c(4,0,4), lambda = "auto")
ts4arima
```

```
## Series: ts4
## ARIMA(0,0,0)(4,0,4)[4] with non-zero mean
## Box Cox transformation: lambda= 0.008732273
##
## Coefficients:
##         sar1    sar2     sar3   sar4     sma1     sma2    sma3     sma4
##       1.3981  0.5539  -1.3865  0.434  -0.3399  -0.9012  0.3426  -0.0226
## s.e.     NaN     NaN      NaN    NaN   0.1871   0.2666  0.2000   0.2509
##          mean
##       22.2098
## s.e.      NaN
##
## sigma^2 = 0.06338:  log likelihood = -9.29
## AIC=38.58   AICc=41.17   BIC=64.23
```

```
auto.arima(ts4, seasonal = TRUE)
```

```
## Series: ts4
## ARIMA(1,1,1)(2,1,0)[4]
##
## Coefficients:
##          ar1      ma1    sar1    sar2
##       0.6580  -0.9850  0.0990  0.4670
## s.e.  0.0885   0.0424  0.1003  0.1071
##
## sigma^2 = 133289278:  log likelihood = -979.66
## AIC=1969.31   AICc=1970.02   BIC=1981.87
```

Model ts4arima okazał się najlepszy spośród testowany modeli.

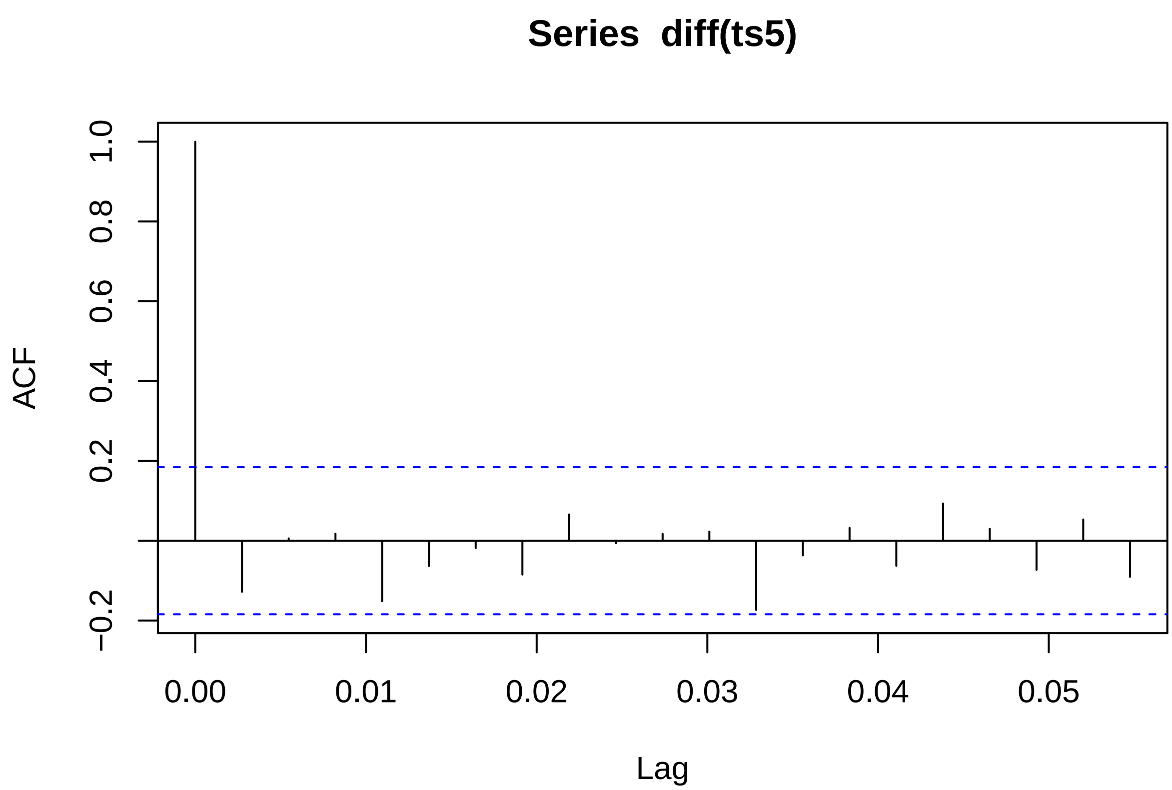
Model 5

Jako, że piąty szereg czasowy też wyszedł niestacjonarny musimy go zróżnicować.

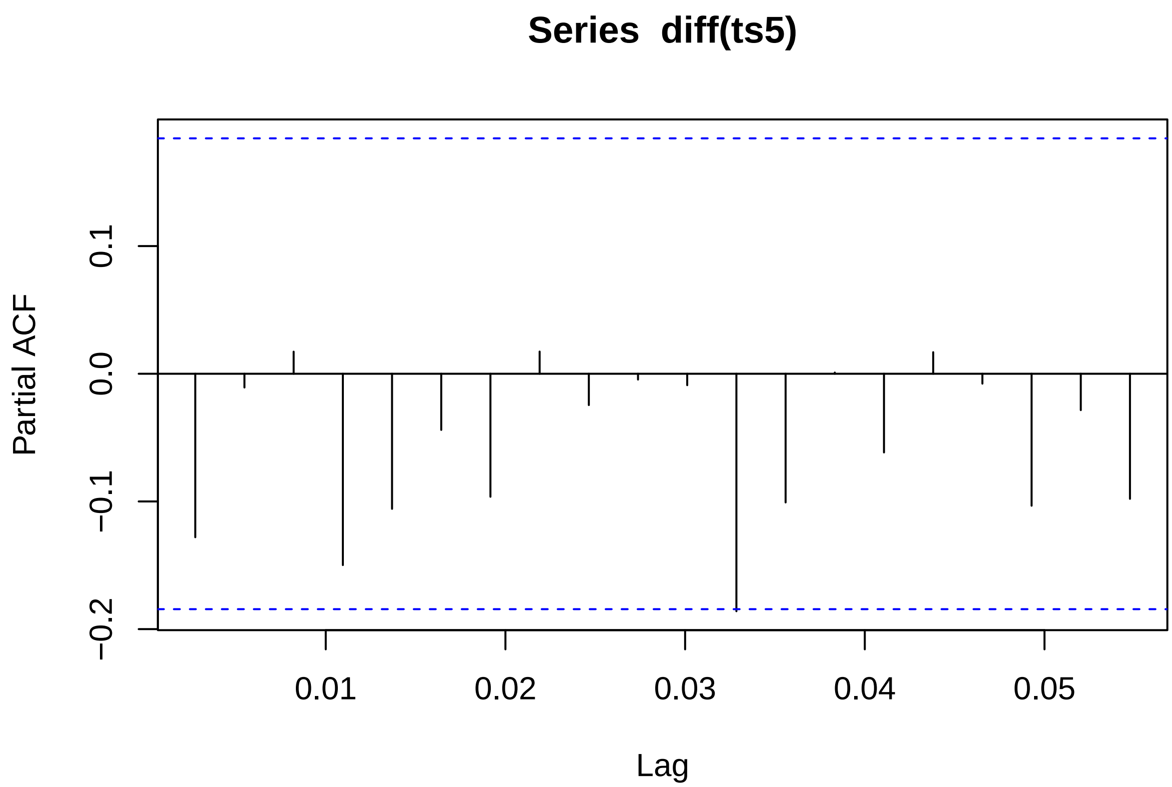
```
diff(ts5) %>% urca::ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0578
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
acf(diff(ts5))
```




```
pacf(diff(ts5))
```



Z testu ACF wychodzi nam, iż q może wychodzić 0. Z testu PACF wychodzi, iż p jest też równe 0.

```r
ts5arima = Arima(y=ts5, order = c(0,1,0), lambda = NULL)
ts5arima
```

```
## Series: ts5
## ARIMA(0,1,0)
##
## sigma^2 = 2.855:  log likelihood = -219.62
## AIC=441.23   AICc=441.27   BIC=443.96
```

```r
Arima(y=ts5, seasonal = c(0,0,0), lambda = NULL)
```

```
## Series: ts5
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##          mean
##       21.7128
## s.e.   0.5931
##
## sigma^2 = 40.45:  log likelihood = -372.17
## AIC=748.33   AICc=748.44   BIC=753.81
```

```
Arima(y=ts5, order = c(1,2,1), lambda = "auto")
```

```
## Series: ts5
## ARIMA(1,2,1)
## Box Cox transformation: lambda= 1
##
## Coefficients:
##           ar1     ma1
##       -0.1219  -1.000
## s.e.   0.0947   0.027
##
## sigma^2 = 2.864:  log likelihood = -219.31
## AIC=444.63   AICc=444.85   BIC=452.78
```

```
auto.arima(ts5, seasonal = FALSE)
```

```
## Series: ts5
## ARIMA(0,1,0)
##
## sigma^2 = 2.855:  log likelihood = -219.62
## AIC=441.23   AICc=441.27   BIC=443.96
```

Model ts5arima okazał się najlepszy spośród testowany modeli.

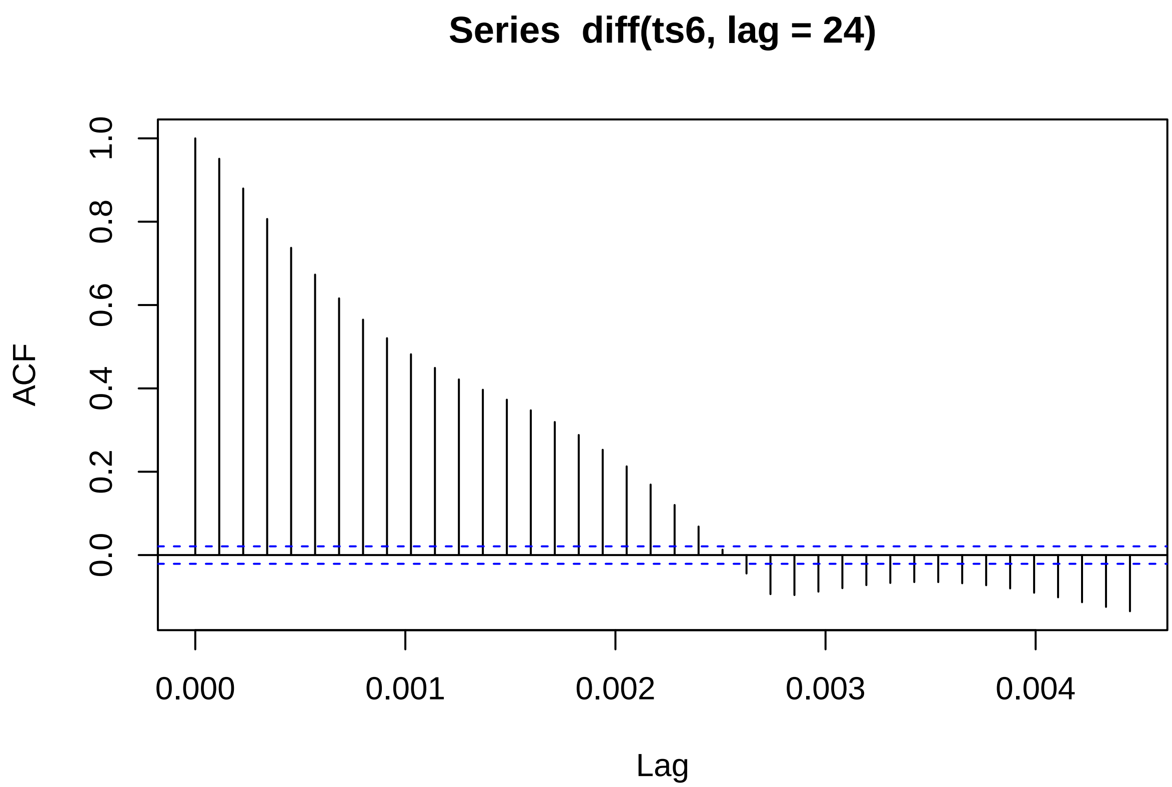
Model 6

Jako, że szósty szereg czasowy wyszedł niestacjonarny musimy go zróżnicować.

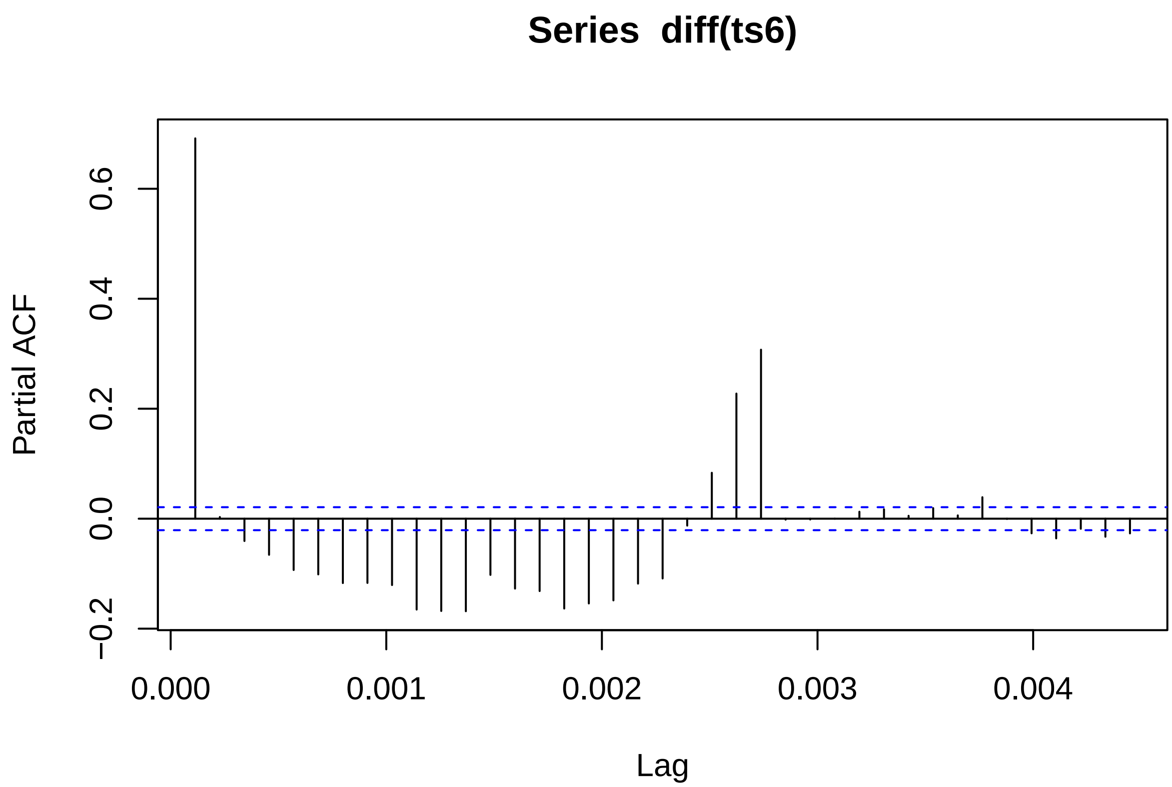
```
diff(ts6) %>% urca::ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 12 lags.
##
## Value of test-statistic is: 0.0098
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
acf(diff(ts6, lag = 24))
```




```r
pacf(diff(ts6))
```



Z testu ACF wychodzi nam, iż q może wychodzić 0. Z testu PACF wychodzi, iż p jest też równe 0. Szereg czasowy wydaje się być sezonowy.

```r
Arima(y=ts6, order = c(0,1,0), lambda = NULL)
```

```
## Series: ts6
## ARIMA(0,1,0)
##
## sigma^2 = 1.357:  log likelihood = -13800.05
## AIC=27602.1   AICc=27602.1   BIC=27609.18
```

```r
Arima(y=ts6, seasonal = c(0,0,0), lambda = NULL)
```

```
## Series: ts6
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##          mean
##       15.7491
## s.e.   0.0861
##
## sigma^2 = 65.12:  log likelihood = -30795.22
## AIC=61594.44   AICc=61594.45   BIC=61608.6
```

```r
Arima(y=ts6, seasonal = c(0,0,0), lambda = "auto")
```

```
## Series: ts6
## ARIMA(0,0,0) with non-zero mean
## Box Cox transformation: lambda= 1
##
## Coefficients:
##          mean
##       14.7491
## s.e.   0.0861
##
## sigma^2 = 65.12:  log likelihood = -30795.22
## AIC=61594.44   AICc=61594.45   BIC=61608.6
```

```r
ts6arima = auto.arima(ts6, seasonal = TRUE)
ts6arima
```

```
## Series: ts6
## ARIMA(5,1,1)
##
## Coefficients:
##          ar1      ar2      ar3     ar4      ar5      ma1
##       1.4922  -0.5278  -0.0160  0.0517  -0.1229  -0.9454
## s.e.  0.0111   0.0193   0.0199  0.0192   0.0109   0.0038
##
## sigma^2 = 0.6079:  log likelihood = -10271
## AIC=20556.01   AICc=20556.02   BIC=20605.57
```

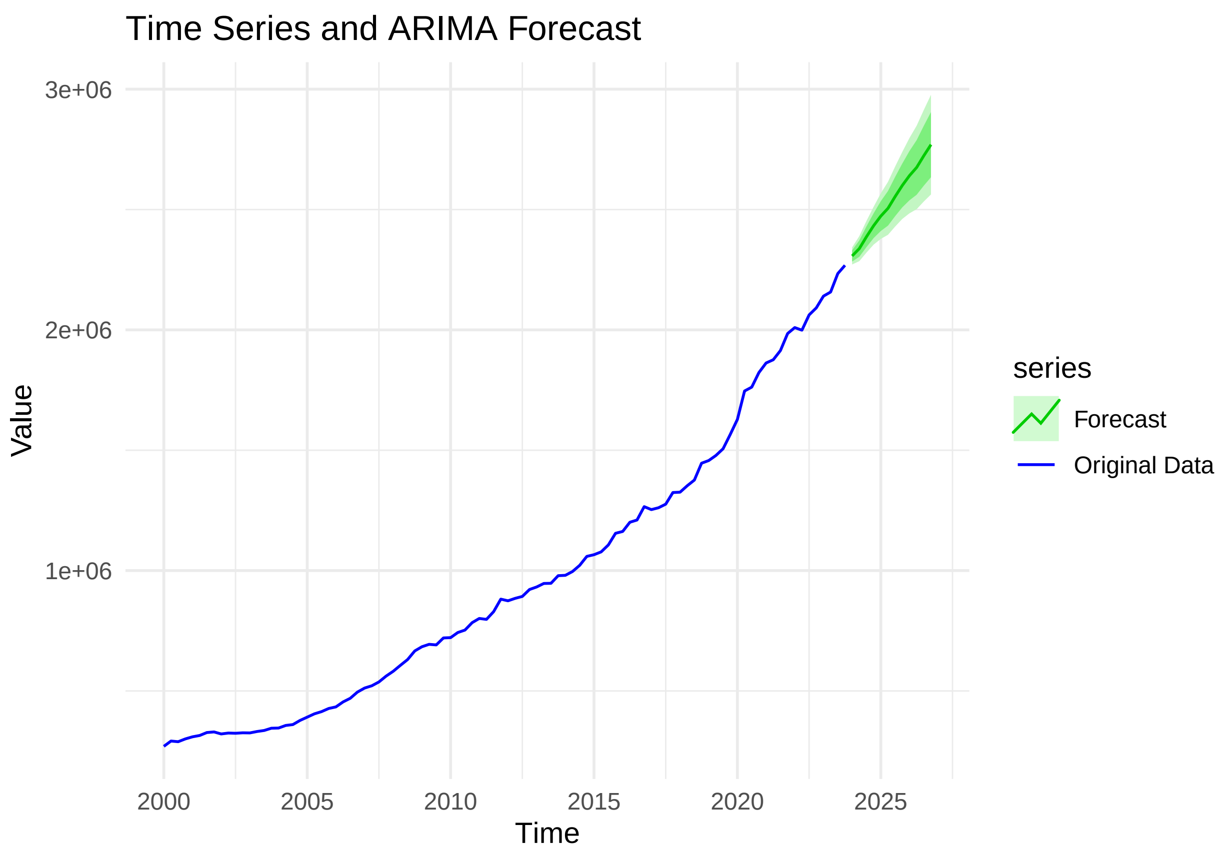
Model ts6arima okazał się najlepszy spośród testowany modeli.

Predykcja

Przeprowadzamy predykcję dla wybranych modeli.

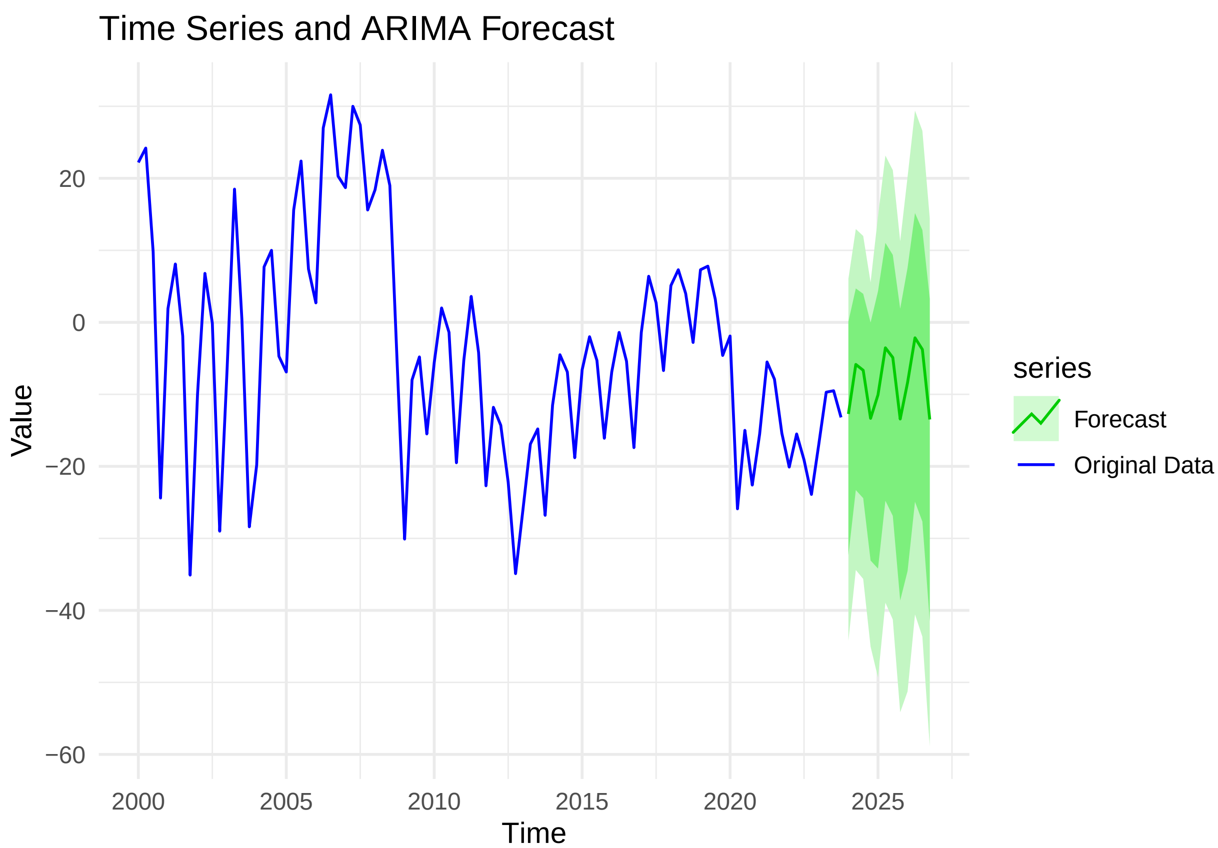
Predykcja 1

```r
prediction1 = forecast(ts1arima, h = 12)
autoplot(ts1, series="Original Data") +
  autolayer(prediction1, series="Forecast", PI=TRUE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="green"))
```



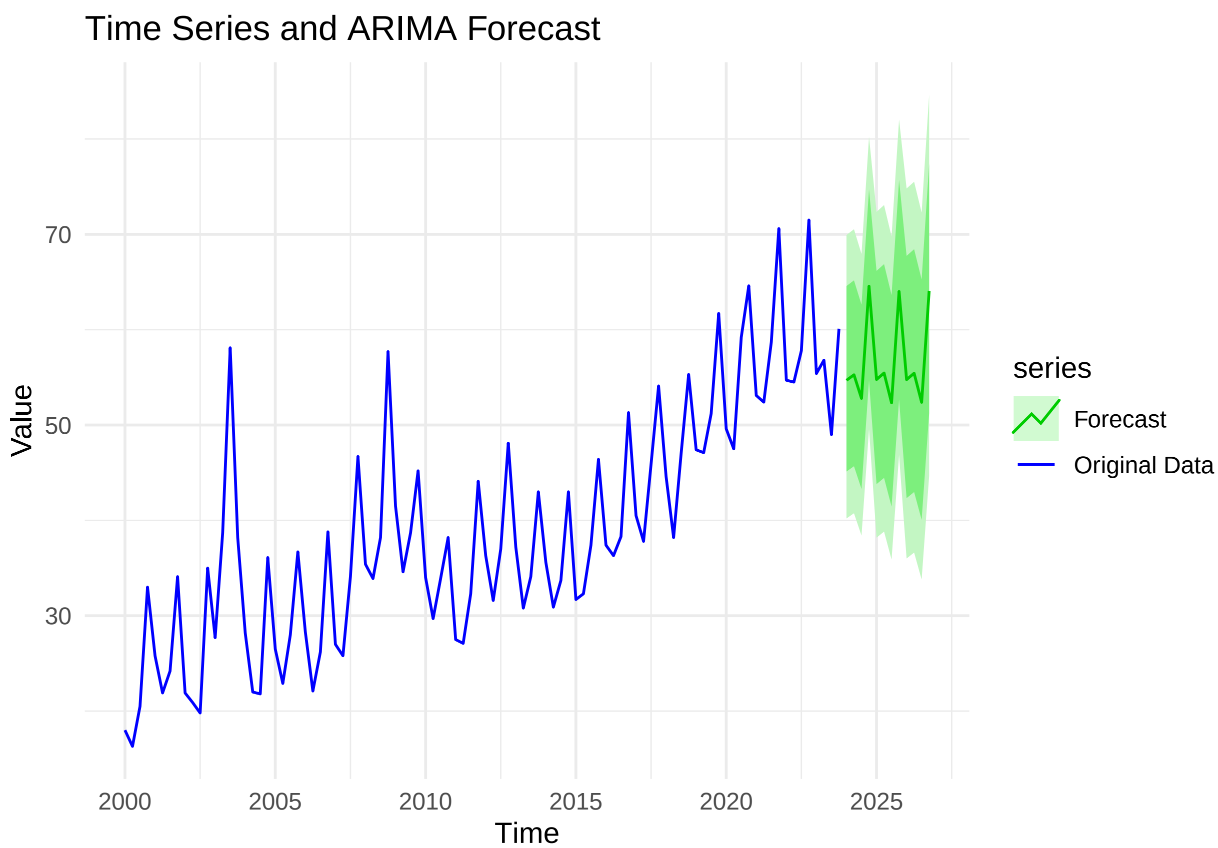
Predykcja 2

```
prediction2 = forecast(ts2arima, h = 12)
autoplot(ts2, series="Original Data") +
  autolayer(prediction2, series="Forecast", PI=TRUE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="green"))
```



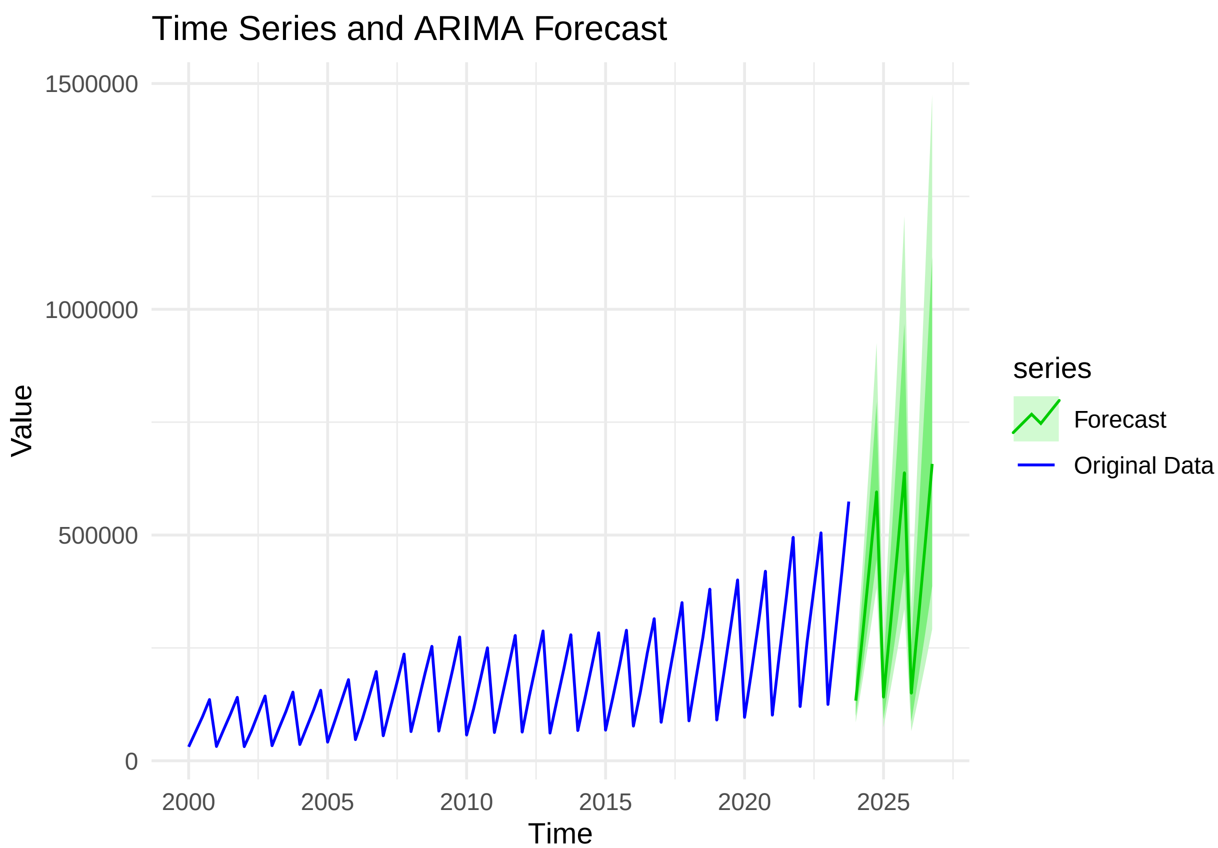
Predykcja 3

```
prediction3 = forecast(ts3arima, h = 12)
autoplot(ts3, series="Original Data") +
  autolayer(prediction3, series="Forecast", PI=TRUE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="green"))
```



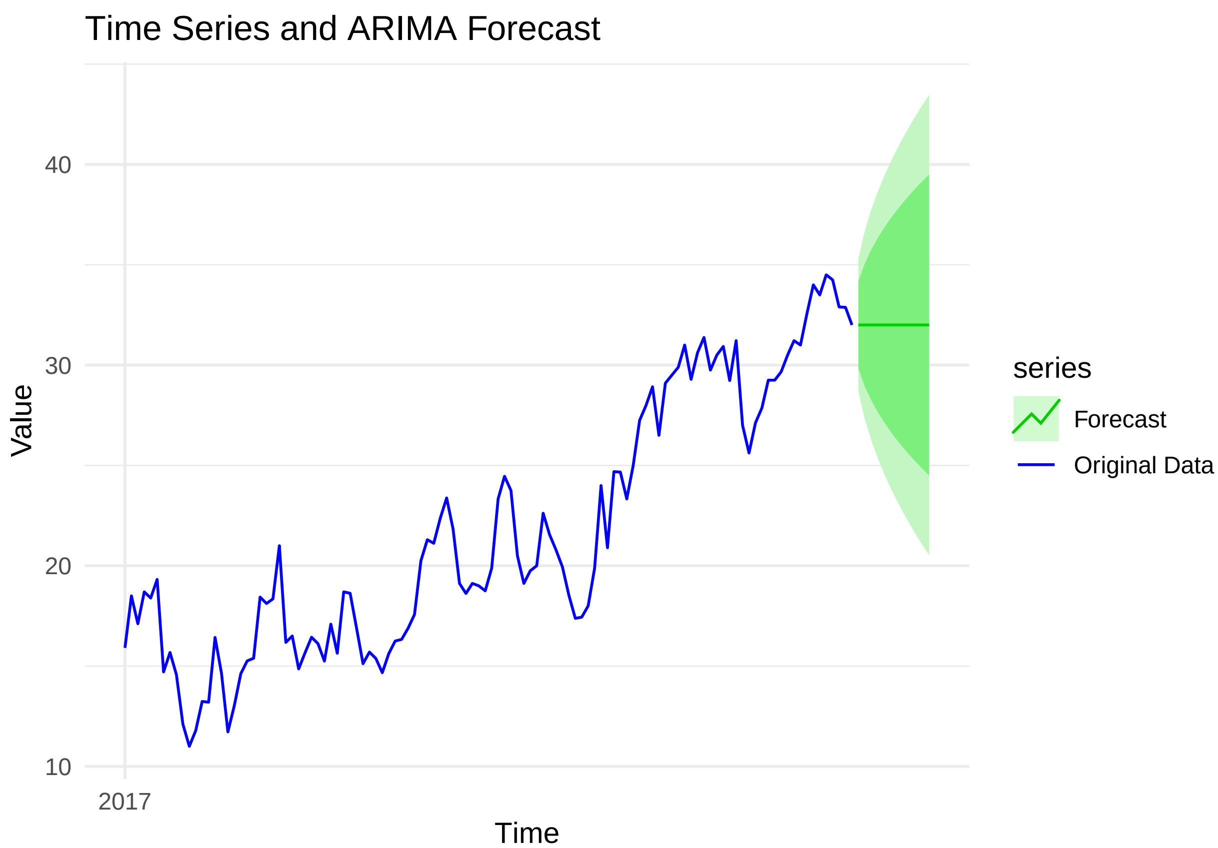
Predykcja 4

```
prediction4 = forecast(ts4arima, h = 12)
autoplot(ts4, series="Original Data") +
  autolayer(prediction4, series="Forecast", PI=TRUE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="green"))
```



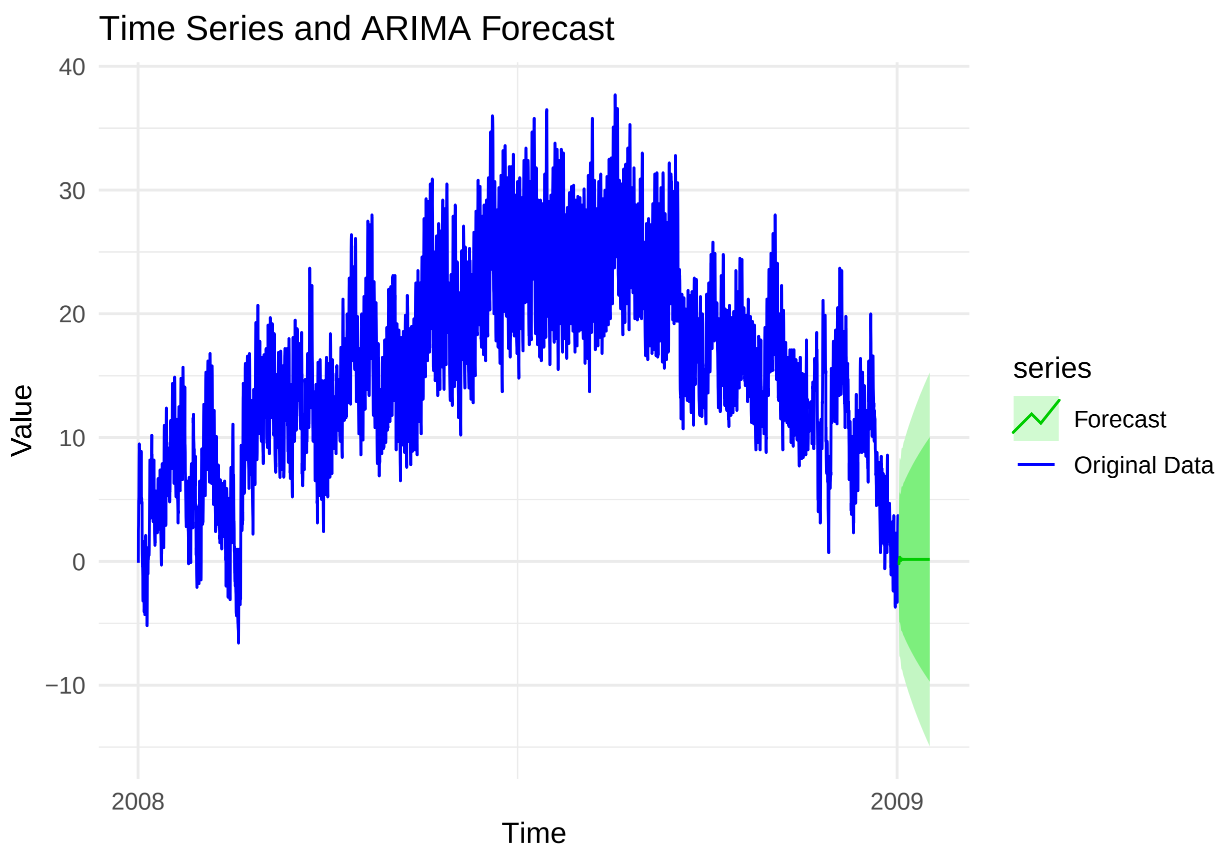
Predykcja 5

```
prediction5 = forecast(ts5arima, h = 12)
autoplot(ts5, series="Original Data") +
  autolayer(prediction5, series="Forecast", PI=TRUE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="green"))
```



Predykcja 6

```
prediction6 = forecast(ts6arima, h = 360)
autoplot(ts6, series="Original Data") +
  autolayer(prediction6, series="Forecast", PI=TRUE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="green"))
```

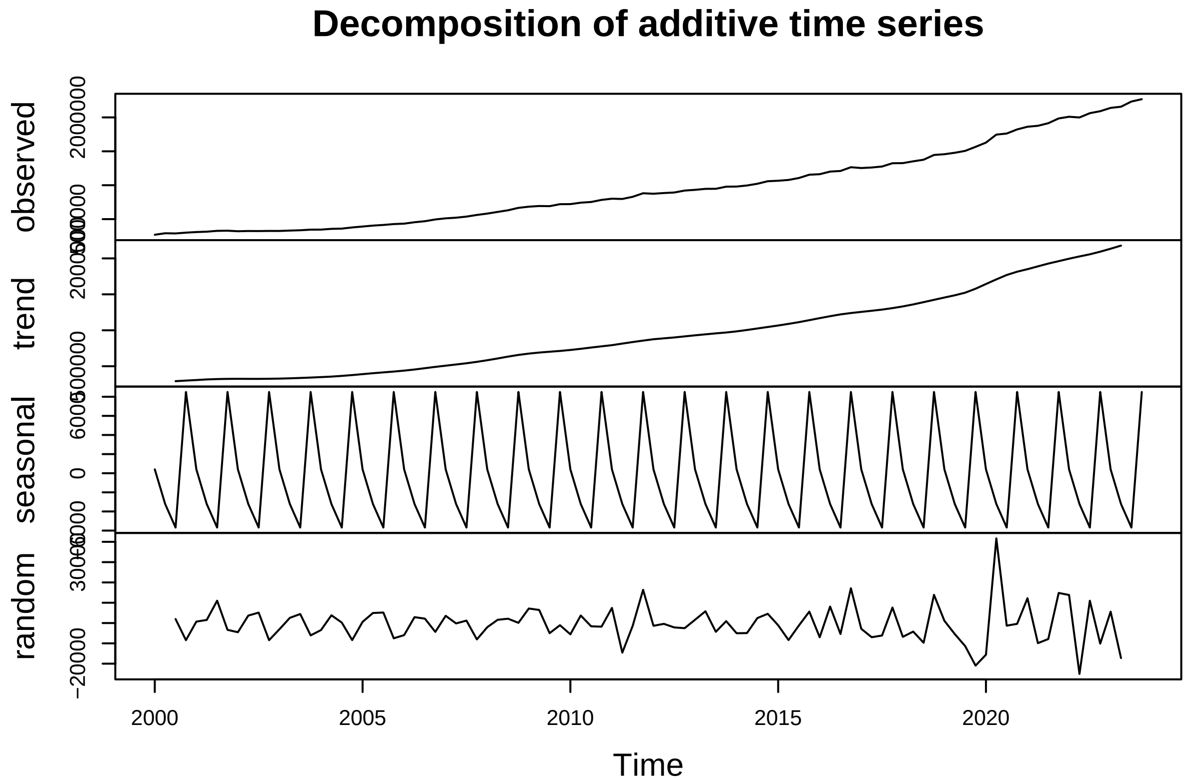


Dekompozycja

Teraz przeprowadzamy dekompozycje dla naszych szeregów.

Dekompozycja 1

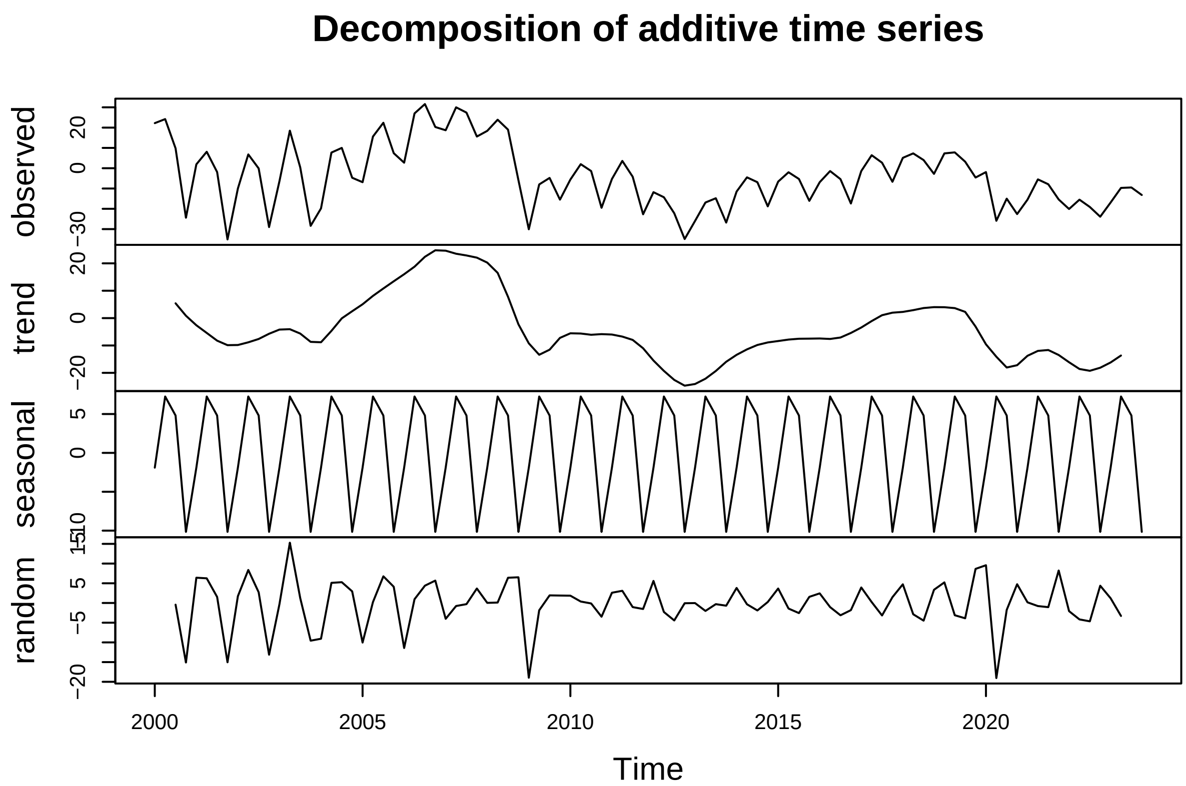
```
decompose(ts1) %>% plot()
```



Z przeprowadzonej dekompozycji dla pierwszego szeregu czasowego wynika, że posiadamy zauważalny trend, zachodzi tutaj sezonowość oraz występują pojedyńcze odchylenia losowe.

Dekompozycja 2

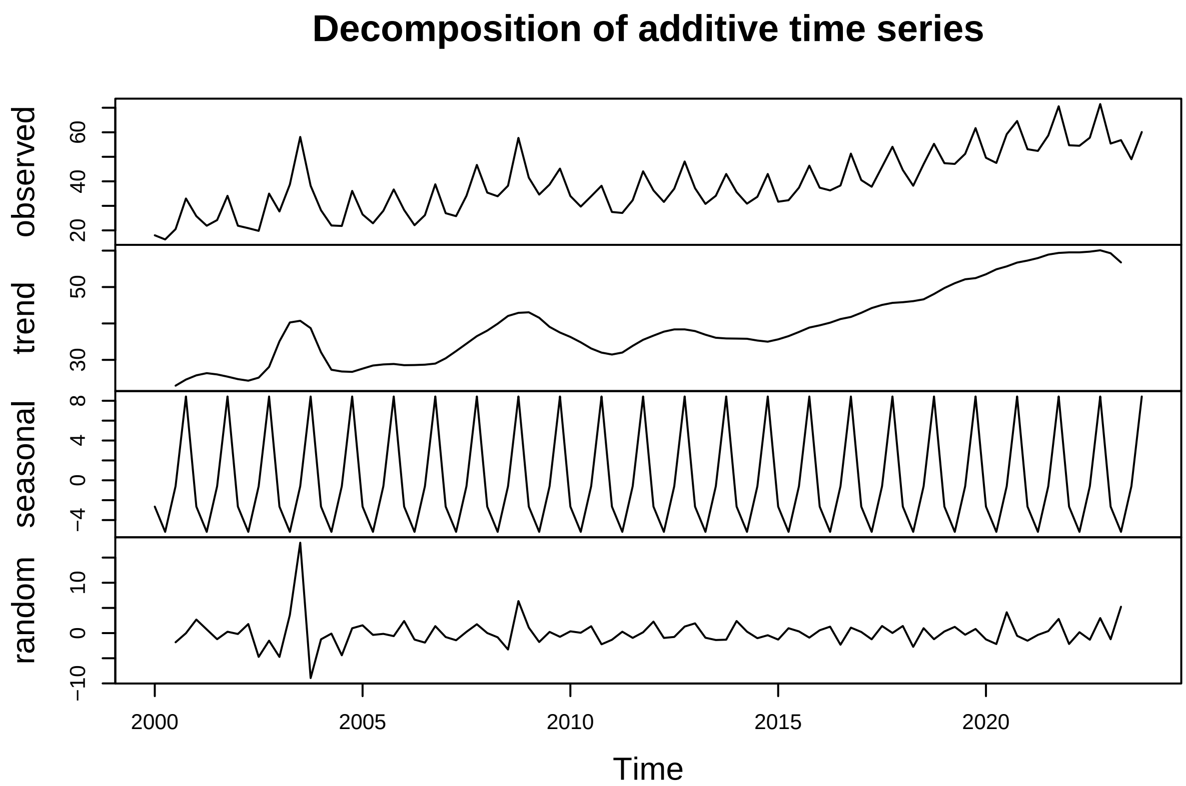
```
decompose(ts2) %>% plot()
```



Z przeprowadzonej dekompozycji dla drugiego szeregu czasowego wynika, że nie posiadamy zauważalnego tredu, zachodzi tutaj sezonowość oraz występują pojedyńcze odchylenia losowe.

Dekompozycja 3

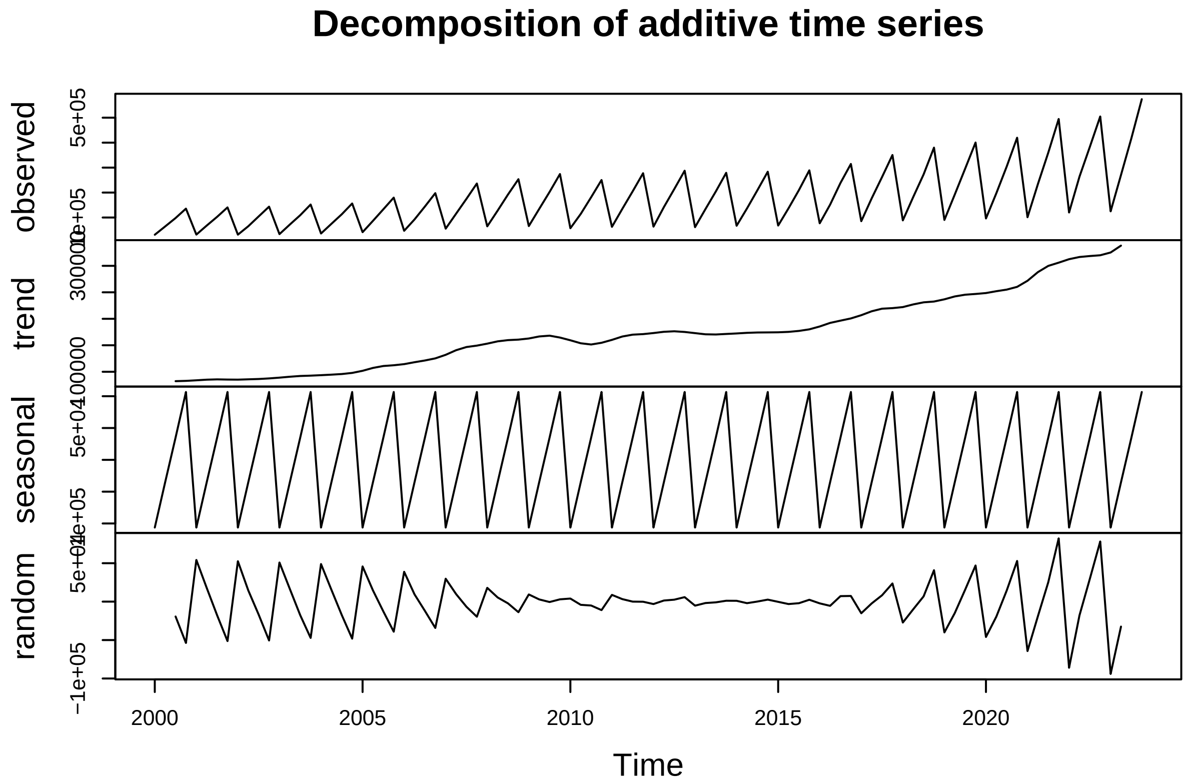
```
decompose(ts3) %>% plot()
```



Z przeprowadzonej dekompozycji dla trzeciego szeregu czasowego wynika, że posiadamy zauważalny tred, zachodzi tutaj sezonowość oraz występują pojedyńcze odchylenia losowe oscylujące wokół zera.

Dekompozycja 4

```
decompose(ts4) %>% plot()
```



Z przeprowadzonej dekompozycji dla czwartego szeregu czasowego wynika, że posiadamy zauważalny tred, zachodzi tutaj sezonowość oraz występują odchylenia losowe.

Dekompozycja 5 i 6

Dla szeregu 5 i 6 nie możemy przeprowadzić dekompozycji, gdyż w pierszym przypadku mamy niepełny okres a w drugim jeden okres.