Austres

Ewa Bojke

Zbiór danych

Pracujemy z danymi o liczbie mieszkańców Australii. Liczby są podawane w tysiącach. Mierzone są one kwartalnie od marca 1971 r. do marca 1994 r. Obiekt austres jest klasy „ts”, czyli jest to szereg czasowy.

Dane początkowe i końcowe z “Austres”

```
head(austres)
```

```
## [1] 13067.3 13130.5 13198.4 13254.2 13303.7 13353.9
```

```
tail(austres)
```

```
## [1] 17447.3 17482.6 17526.0 17568.7 17627.1 17661.5
```

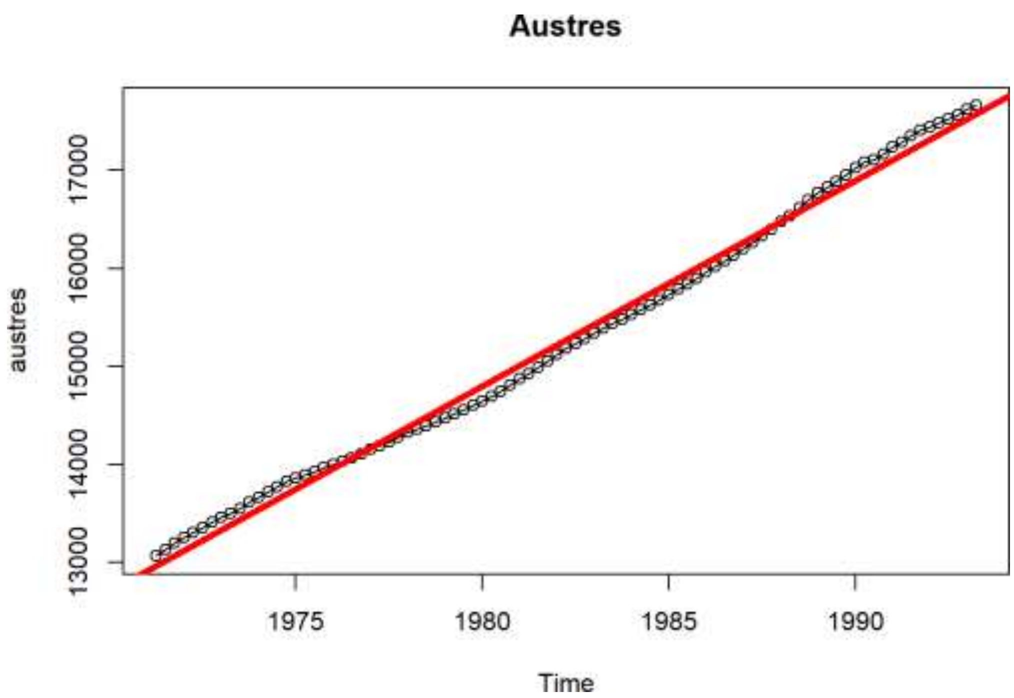
Ćwiczenie I

Dopasowujemy model liniowy do danych.

```
t<-time(austres)
model<-lm(austres~t)
```

Rysujemy dane oraz dodajemy regresje liniową.

```
plot(austres,col="black",type="o",main="Austres")
abline(model,lwd=4,col="red")
```



Wnioski i obserwacje:

Na wykresie regresji możemy zaobserwować, że funkcja predykcji(funkcja liniowa rysowana kolorem czerwonym) jest bardzo zbliżona do danych. Występuje wysoka dokładność predykcji.

Ćwiczenie II

Badamy stacjonarność szeregu. Jeżeli p-value < 0.05 to szereg jest stacjonarny, jeżeli jest odwrotnie to szereg jest niestacjonarny. Drugą metodą jest badanie trendu i sezonowości.

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.5.3
```

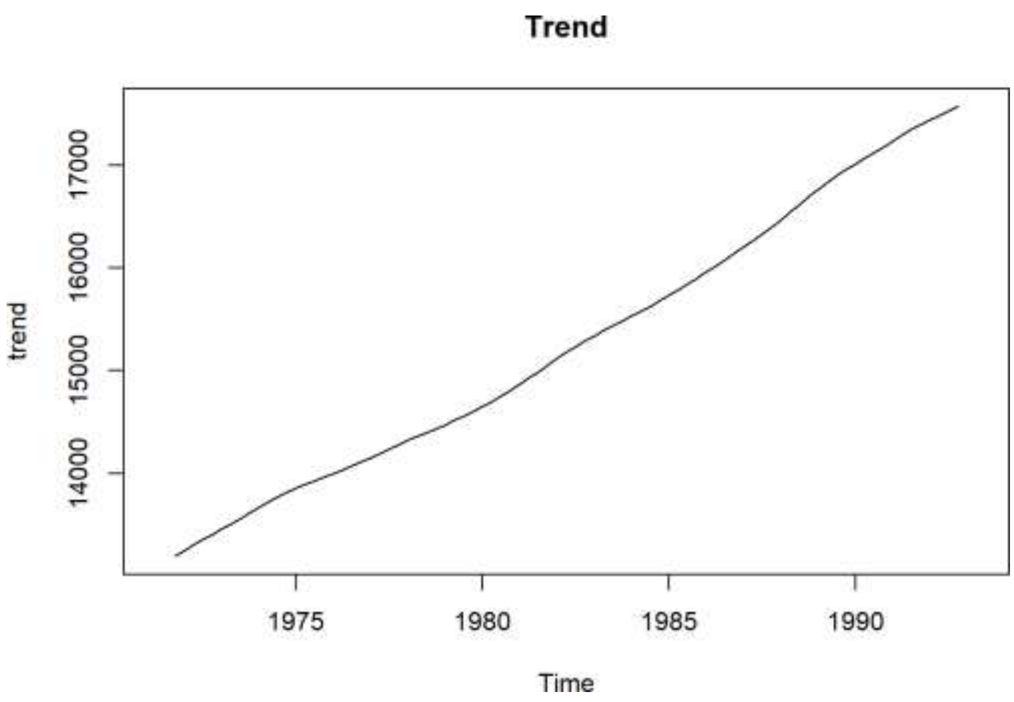
```
p.value=adf.test(austres)$p.value
p.value
```

```
## [1] 0.3493465
```

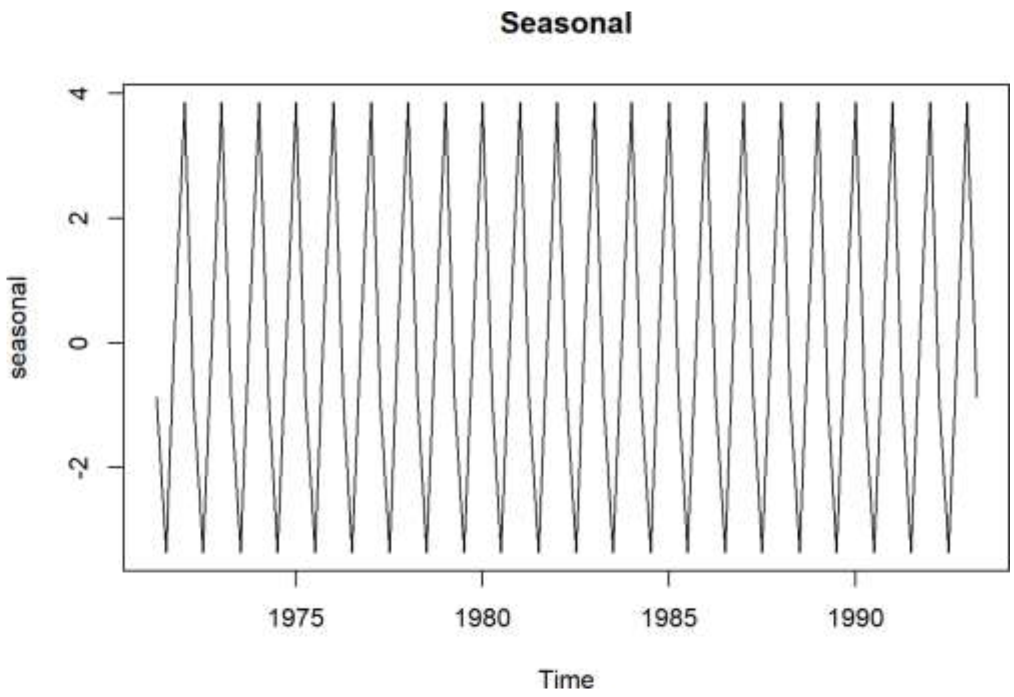
```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.5.3
```

```
trend=ma(austres,order=4,centre=TRUE) #usuwamy trend ă¤Lăąłrednić Lă¸ą
plot(trend, main="Trend")
```




```
seasonal=decompose(austres)$seasonal #wycić Lă¸ągamy sezonowoă¤Lăąłć"Lă¸¶ z danych
plot(seasonal, main="Seasonal")
```



Wnioski i obserwacje: Szereg czasowy z danymi austres jest niestacjonarny, ponieważ p-value > 0.05. Druga metoda również wskazuje na to żeszereg jest niestacjorany, ponieważ szereg ma trend i sezonowość.

Szereg przekształcamy w szereg stacjonarny za pomocą eliminacji trendu i sezonowości.Podwójnie różniczkujemy dane.

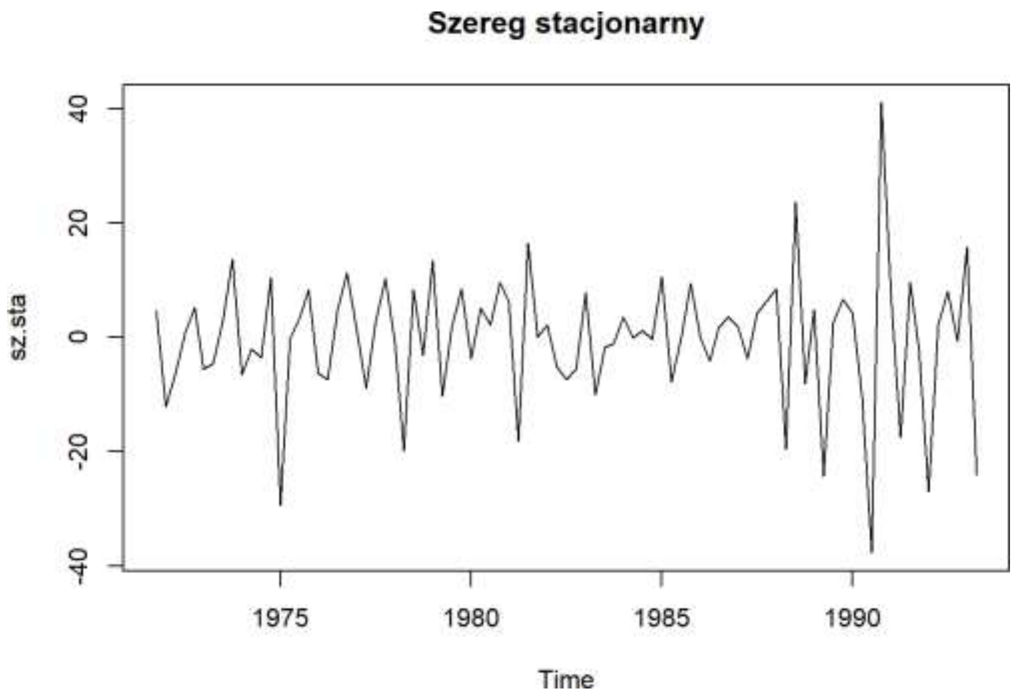
```
sz.sta= diff(diff(austres))
p.value1=adf.test(sz.sta)$p.value
```

```
## Warning in adf.test(sz.sta): p-value smaller than printed p-value
```

```
p.value1
```

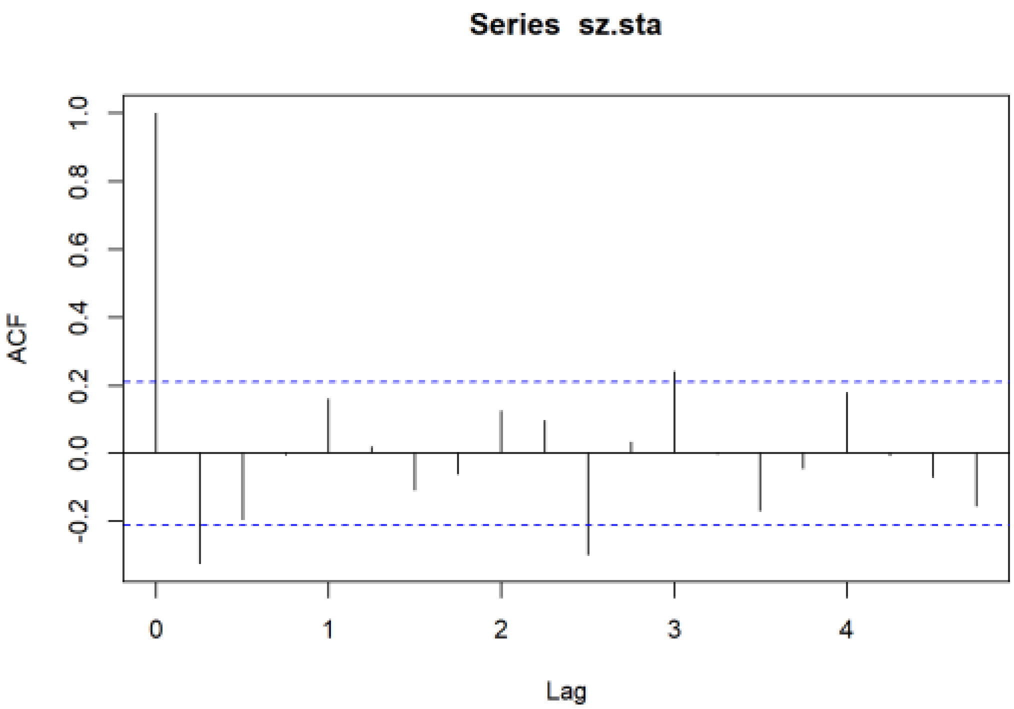
```
## [1] 0.01
```

```
plot(sz.sta,main="Szereg stacjonarny",type="l")
```

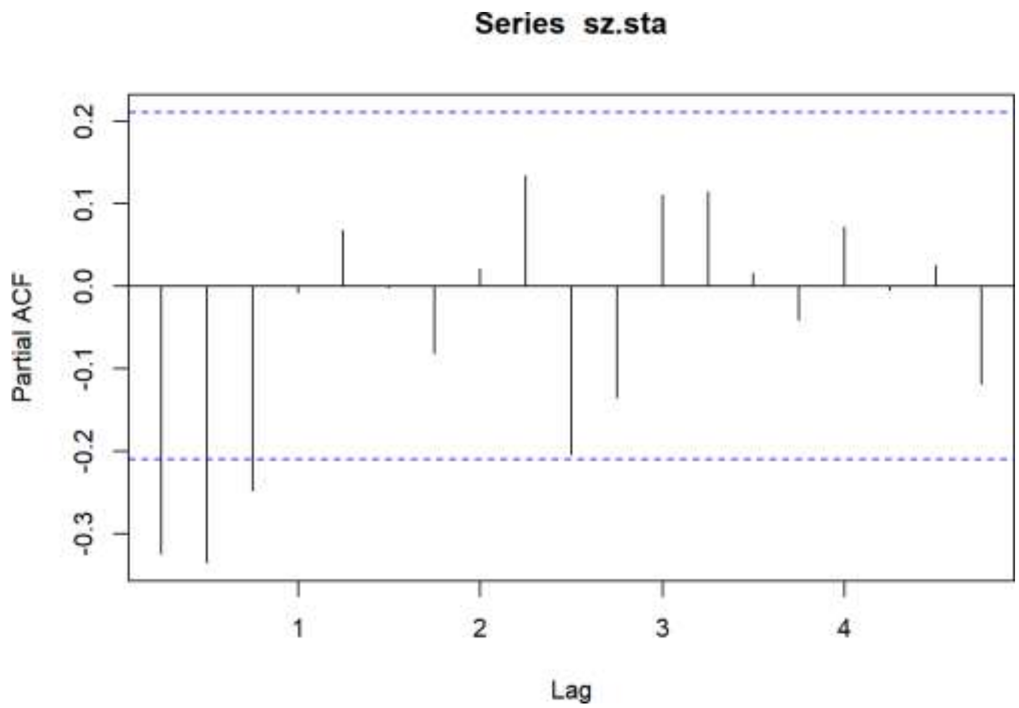


Wnioski i obserwacje: P-value wynosi mniej niż 0.01, czyli otrzymaliśmy szereg stacjonarny. Szereg stacjonarny składa sie tylko z losowych wartości. Zbadajmy jego autokorelację i cząstkową autokorelacje.

```
acf(sz.sta)
```




```
pacf(sz.sta)
```



Model jaki dopasujemy do tego szeregu to AR(2) dla p=2. Patrząc na autokorelację częściową możemy stwierdzić, że pierwsze dwie autokorelacje są większe od zera, dlatego je odrzucamy badając hipotezę: pacf(h)=0, gdzie h > p. Natomiast pozostałe autokorelacje są mniejsze i statystycznie ich nie odrzucamy,i dlatego też wybraliśmy model AR(2).

Wyliczamy wartości krytyczne dla 2/3 danych i przedstawiamy reszty z regresji liniowej szeregu stacjonarnego dla porównania dla modelu AR(2) i AR(3).

```
library("astsa")
```

```
## Warning: package 'astsa' was built under R version 3.5.3
```

```
##
## Attaching package: 'astsa'
```

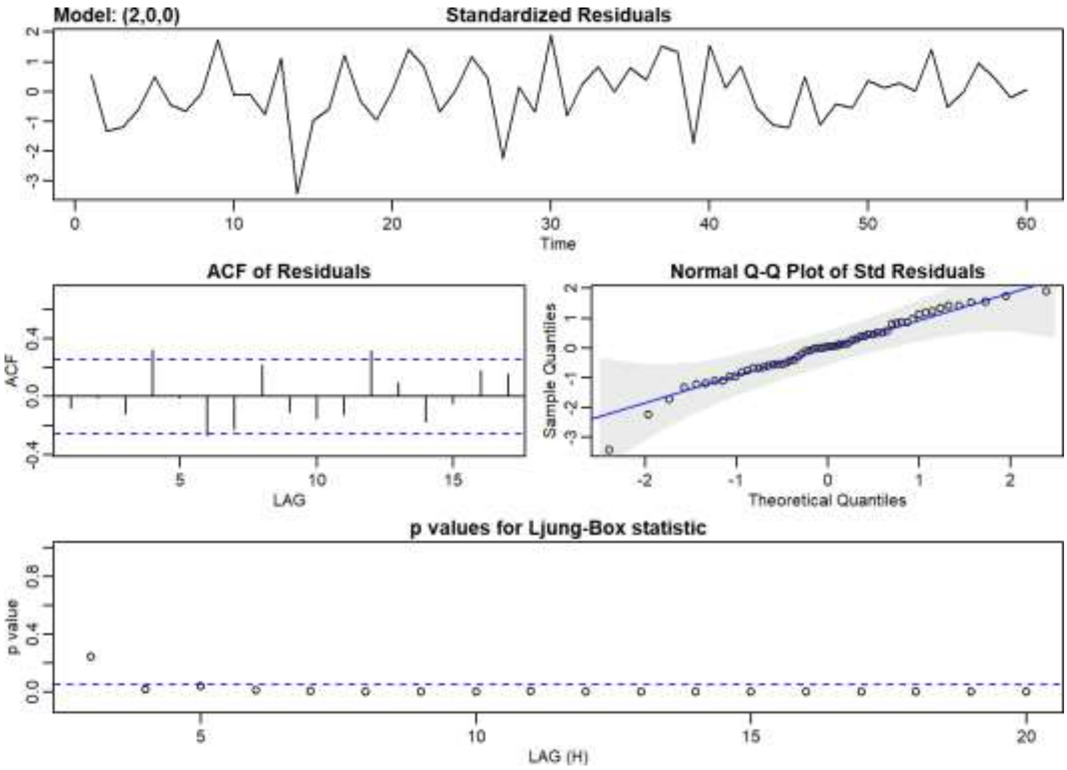
```
## The following object is masked from 'package:forecast':
##
##     gas
```

```
r<-diff(diff(resid(lm(austres~time(austres)))))
r[1:60]->trening
r[61:89]->prognoza
```

Model AR(2)

```
sarima(trening,2,0,0)-> mod.ex
```

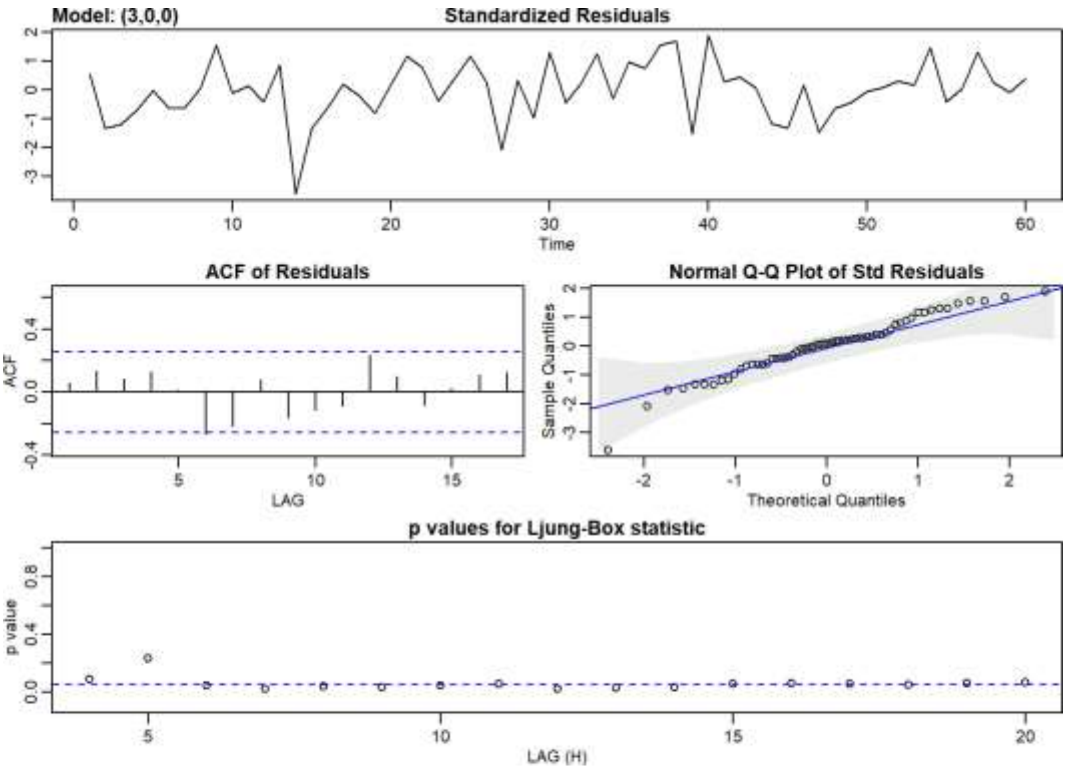
```
## initial  value 2.121512
## iter   2 value 2.050726
## iter   3 value 2.047735
## iter   4 value 2.046172
## iter   5 value 2.045001
## iter   6 value 2.044939
## iter   7 value 2.044935
## iter   7 value 2.044935
## iter   7 value 2.044935
## final  value 2.044935
## converged
## initial  value 2.047931
## iter   2 value 2.047848
## iter   3 value 2.047841
## iter   4 value 2.047834
## iter   4 value 2.047834
## iter   4 value 2.047834
## final  value 2.047834
## converged
```



Model AR(3)

```
sarima(trening,3,0,0)-> mod2.ex
```

```
## initial  value 2.125389
## iter   2 value 2.027613
## iter   3 value 2.010658
## iter   4 value 2.001104
## iter   5 value 1.997352
## iter   6 value 1.996914
## iter   7 value 1.996857
## iter   8 value 1.996856
## iter   8 value 1.996856
## iter   8 value 1.996856
## final  value 1.996856
## converged
## initial  value 2.007058
## iter   2 value 2.006813
## iter   3 value 2.006465
## iter   4 value 2.006352
## iter   5 value 2.006330
## iter   6 value 2.006330
## iter   6 value 2.006330
## iter   6 value 2.006330
## final  value 2.006330
## converged
```



Wartośi krytyczne

```
AIC1 = mod.ex$AIC
AIC2=mod2.ex$AIC
AIC1
```

```
## [1] 7.066879
```

```
AIC2
```

```
## [1] 7.017203
```

AIC2(kryterium Akaike) dla modelu AR(3) jest mniejsze od AIC1(kryterium Akaike) dla modelu Ar(2), dlatego możemy powiedzieć, że model AR(2) jest dokładniejszy.

Szacowanie parametrów za pomocą regresji liniowej używając dwóch metod.

```
ws=coef(arima(trening,order=c(2,0,0)))
ws
```

```
##         ar1         ar2   intercept
## -0.35001280 -0.26560384 -0.07491922
```

```
ar.mle(trening,order.max=2)
```

```
##
## Call:
## ar.mle(x = trening, order.max = 2)
##
## Coefficients:
##       1        2
## -0.3500 -0.2656
##
## Order selected 2  sigma^2 estimated as   59.85
```

Szacowanie parametrów za pomocąNW.

```
tren<-diff(diff(austres)[1:60])
ar.mle(tren,order.max=2)
```

```
##
## Call:
## ar.mle(x = tren, order.max = 2)
##
## Coefficients:
##       1        2
## -0.3494 -0.2653
##
## Order selected 2  sigma^2 estimated as   60.86
```

Szacowanie parametrów za pomocąYW.

```
ar.yw(diff(diff(austres)[1:60]),order.max=2)
```

```
##
## Call:
## ar.yw.default(x = diff(diff(austres)[1:60]), order.max = 2)
##
## Coefficients:
##       1        2
## -0.3537 -0.2670
##
## Order selected 2  sigma^2 estimated as  64.25
```

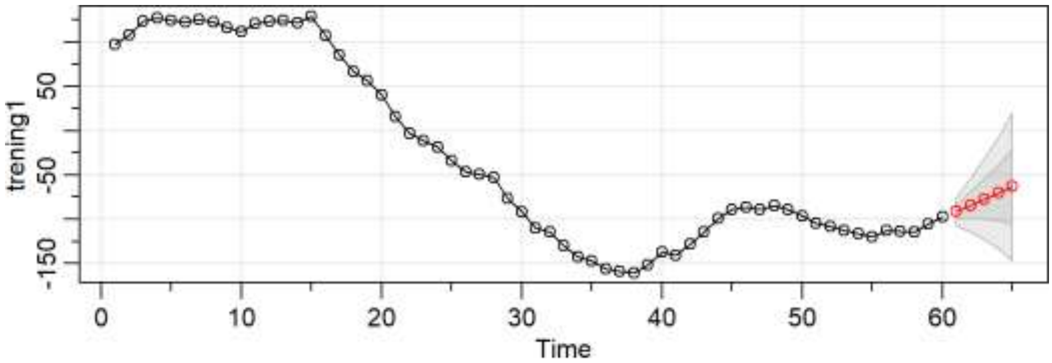
Wnioski i obserwacje: Używając każdej z metod możemy powiedzieć,że parametry sąbardzo zbliżone do siebie. Model jest poprawny.

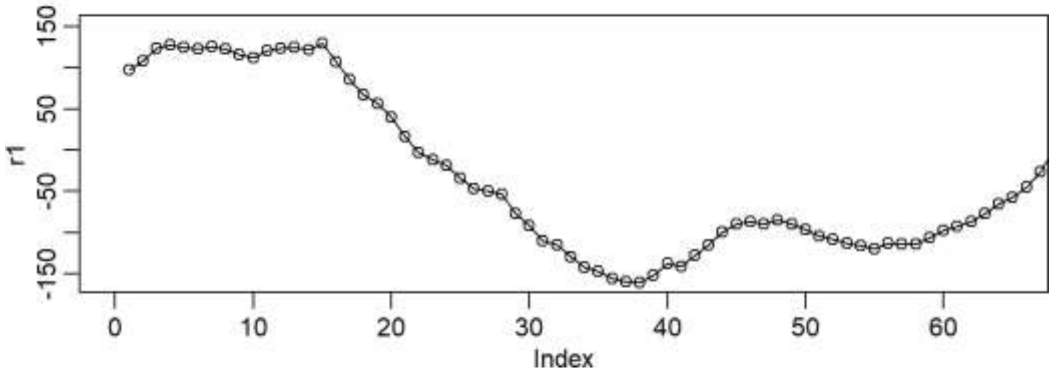
Sprawdzenie dokładnośi predykcji.

```
r1<-resid(lm(austres~time(austres)))
r1[1:60]->trening1
r1[61:89]->prognoza1
par(mfrow=c(2,1))
sarima.for(trening1,5,2,2,0) #predykcja
```

```
## $pred
## Time Series:
## Start = 61
## End = 65
## Frequency = 1
## [1] -91.23517 -84.30992 -77.03680 -70.11050 -63.15438
##
## $se
## Time Series:
## Start = 61
## End = 65
## Frequency = 1
## [1]  7.86713 15.19176 22.81564 31.79762 41.83054
```

```
plot(r1,type="o",xlim=c(0,65))#reszty regresji
```





Wniosek i obserwacje: Na podstawie 2/3 danych model przewiduje kolejne dane. Wnioskując po obserwacjach z obydwóch wykresów możemy stwierdzić, że nasz model dobrze przewidział kolejne dane. Występuje duża dokładność predykcji.