

Analiza danych - Projekt v.0.1

Kamil Kukielka, Michał Zakielarz, Klaudia Kopec

2024-06-20

Zaczytanie danych i ich przygotowanie

W tym projekcie będziemy korzystać z kilku zestawów danych, tak aby wykazać, że pomimo danych z różnych dziedzin ich predykcja jest w jakimś sposób możliwa

Zbiór 1

Nasz pierwszy zestaw dotyczy różnych wskaźników pewnego przedsiębiorstwa, które zmieniają się w czasie. Dane te były aktualizowane co miesiąc i obejmują zakres od 01.01.2015 do 01.02.2020 Teraz przedstawimy fragment naszych danych aby wiedzieć z czym mamy doczynienia.

Zestaw 1

Zawiera on przychodów naszego przedsiębiorstwa.

##	Period	Revenue
## 1	01.01.2015	16010072.1195
## 2	01.02.2015	15807587.4498
## 3	01.03.2015	22047146.0236
## 4	01.04.2015	18814583.2943
## 5	01.05.2015	14021479.6117
## 6	01.06.2015	16783928.5221
## 7	01.07.2015	19161892.1949
## 8	01.08.2015	15204984.2967
## 9	01.09.2015	20603939.9751
## 10	01.10.2015	20992874.7801
## 11	01.11.2015	14993369.6576
## 12	01.12.2015	27791807.6398

Zestaw 2

Zawiera ilość sprzedaży w naszej firmie.

##	Period	Sales_quantity
## 1	01.01.2015	12729
## 2	01.02.2015	11636
## 3	01.03.2015	15922
## 4	01.04.2015	15227

## 5	01.05.2015	8620
## 6	01.06.2015	13160
## 7	01.07.2015	17254
## 8	01.08.2015	8642
## 9	01.09.2015	16144
## 10	01.10.2015	18135
## 11	01.11.2015	10841
## 12	01.12.2015	22113

Zestaw 3

Zawiera średni koszt produkcji w naszej firmie.

##	Period	Average_cost
## 1	01.01.2015	1257.76354148
## 2	01.02.2015	1358.50699981
## 3	01.03.2015	1384.69702447
## 4	01.04.2015	1235.60670482
## 5	01.05.2015	1626.62176470
## 6	01.06.2015	1275.37450776
## 7	01.07.2015	1110.57680508
## 8	01.08.2015	1759.42887025
## 9	01.09.2015	1276.25990926
## 10	01.10.2015	1157.58890434
## 11	01.11.2015	1383.02459714
## 12	01.12.2015	1256.80855786

Zestaw 4

Zawiera informację o średniej liczbie pracowników w regionie (rocznie).

##	Period	Average_annual_payroll_of_region
## 1	01.01.2015	30024676
## 2	01.02.2015	30024676
## 3	01.03.2015	30024676
## 4	01.04.2015	30024676
## 5	01.05.2015	30024676
## 6	01.06.2015	30024676
## 7	01.07.2015	30024676
## 8	01.08.2015	30024676
## 9	01.09.2015	30024676
## 10	01.10.2015	30024676
## 11	01.11.2015	30024676
## 12	01.12.2015	30024676

Zbiór 2

Obejmuje średnią dzienną temperaturę w Mumbaiu. Nasz zbiór zawiera więcej danych takich jak wilgoć, prędkość czy kierunek wiatru, jednak my skupimy się tylok na temperaturze.

##	Data	Temperatura
## 1	01-01-2016	28.4
## 2	02-01-2016	26.8
## 3	03-01-2016	25.5
## 4	04-01-2016	26.4
## 5	05-01-2016	27.1
## 6	06-01-2016	26.9
## 7	07-01-2016	26.1
## 8	08-01-2016	26.6
## 9	09-01-2016	26.3
## 10	10-01-2016	26.0
## 11	11-01-2016	26.1
## 12	12-01-2016	25.1

Zbiór 3

Zawiera on kwartalne dane o długu publicznym USA (podany w milionach USD).

##	Data	Dług
## 1	1966-01-01	320999
## 2	1966-04-01	316097
## 3	1966-07-01	324748
## 4	1966-10-01	329319
## 5	1967-01-01	330947
## 6	1967-04-01	322893
## 7	1967-07-01	335896
## 8	1967-10-01	344663
## 9	1968-01-01	349473
## 10	1968-04-01	345369
## 11	1968-07-01	354743
## 12	1968-10-01	358029

Zamiana na szereg czasowy

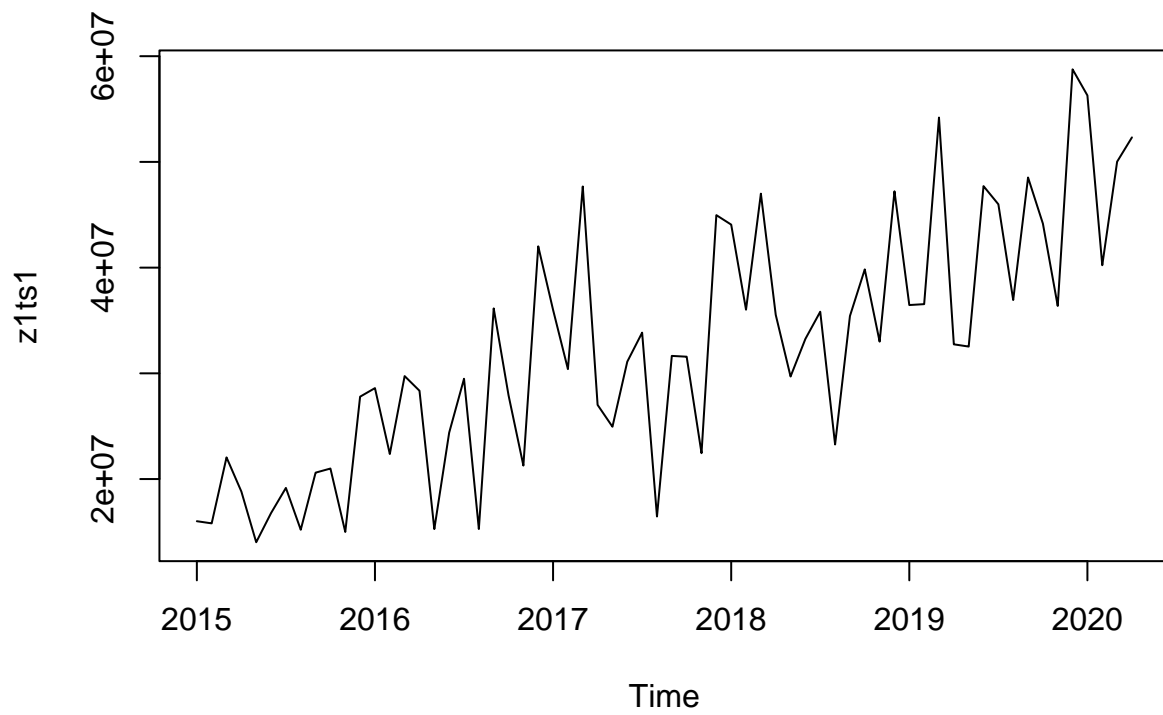
Teraz kiedy mamy już nasze dane musimy je zamienić na szeregi czasowe.

```
z1ts1 <- ts(z1df1$Revenue, start=c(2015,1), frequency = 12)
z1ts2 <- ts(z1df2$Sales_quantity, start=c(2015,1), frequency = 12)
z1ts3 <- ts(z1df3$Average_cost, start=c(2015,1), frequency = 12)
z1ts4 <- ts(z1df4$Average_annual_payroll_of_region, start=c(2015,1), frequency = 12)
z2ts1 <- ts(z2df1$Temperatura, start = c(2016,1,1), frequency = 365)
z3ts1 <- ts(z3df1$Dług, start = c(1966,1), frequency = 4)
```

Mając nasze dane w tej formie możemy narysować wykresy, aby zobaczyć jak wyglądają.

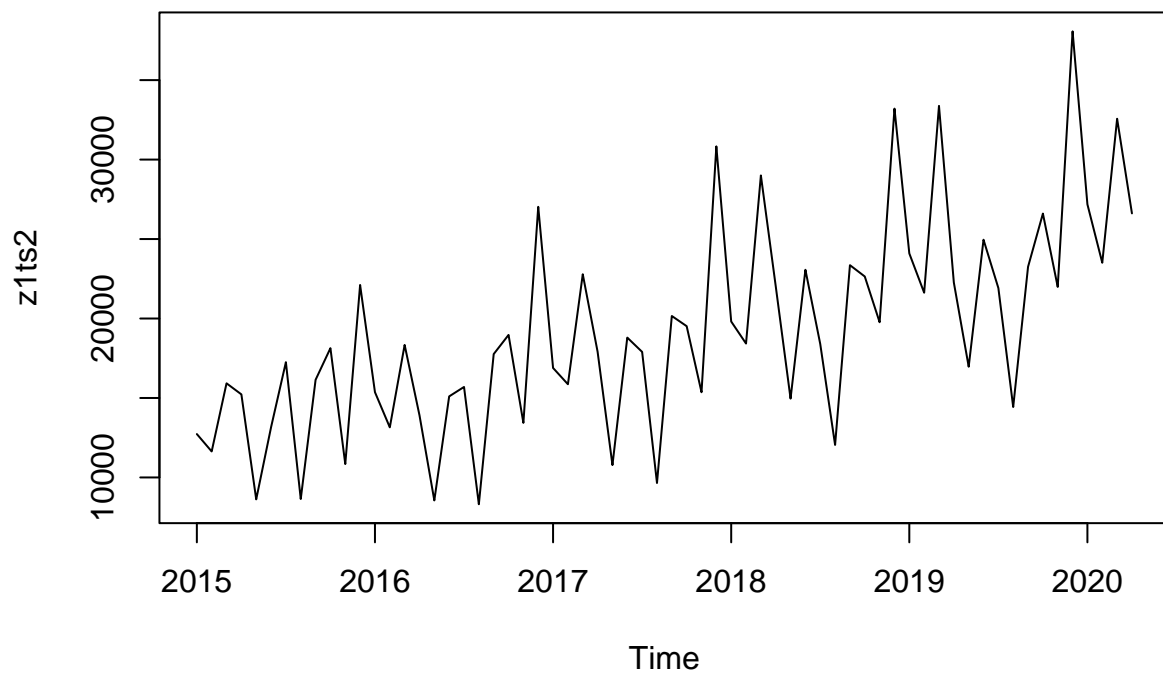
Zbiór 1 zestaw 1

```
plot(z1ts1)
```



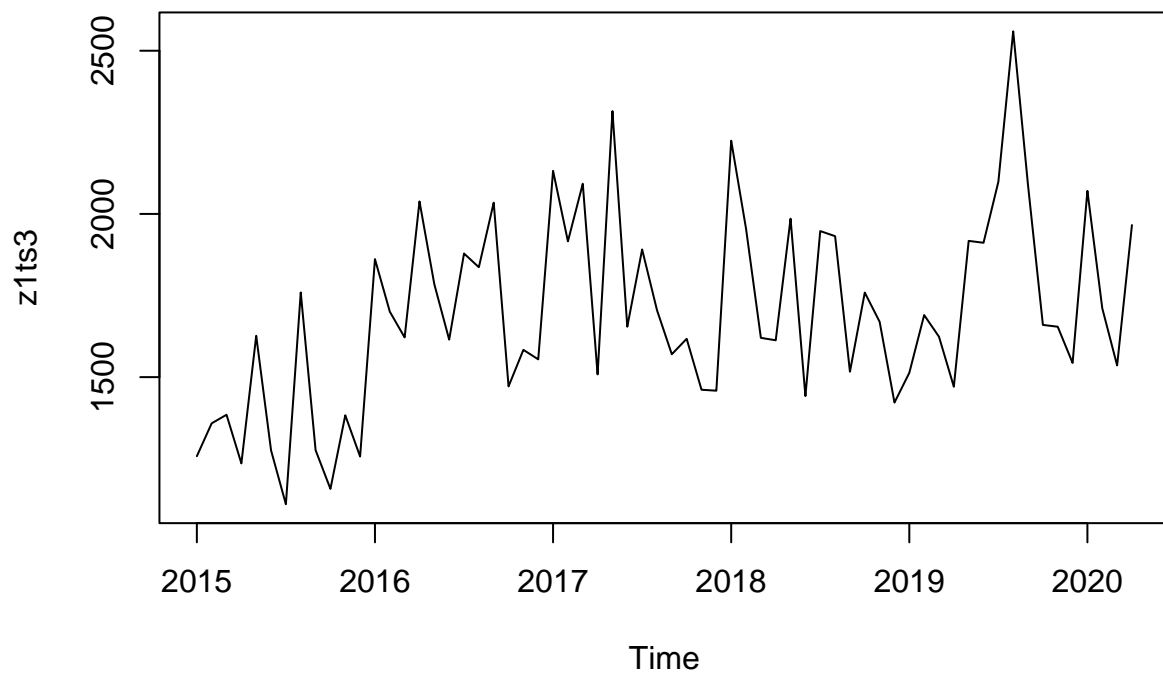
Zbiór 1 zestaw 2

```
plot(z1ts2)
```



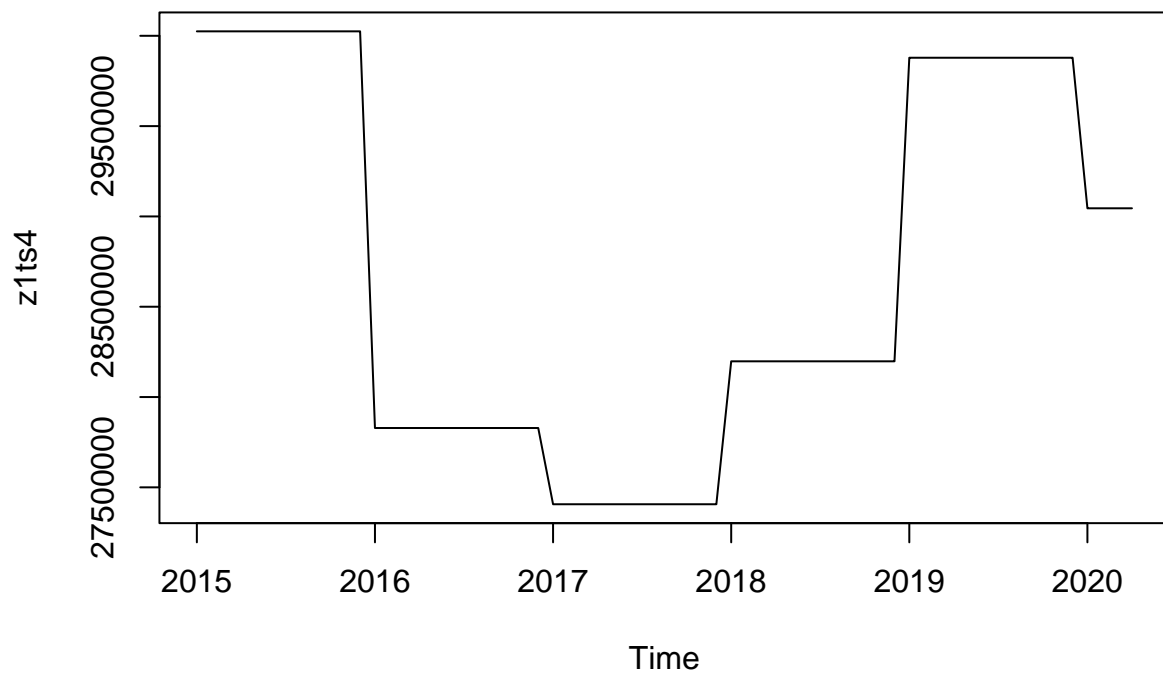
Zbiór 1 zestaw 3

```
plot(z1ts3)
```



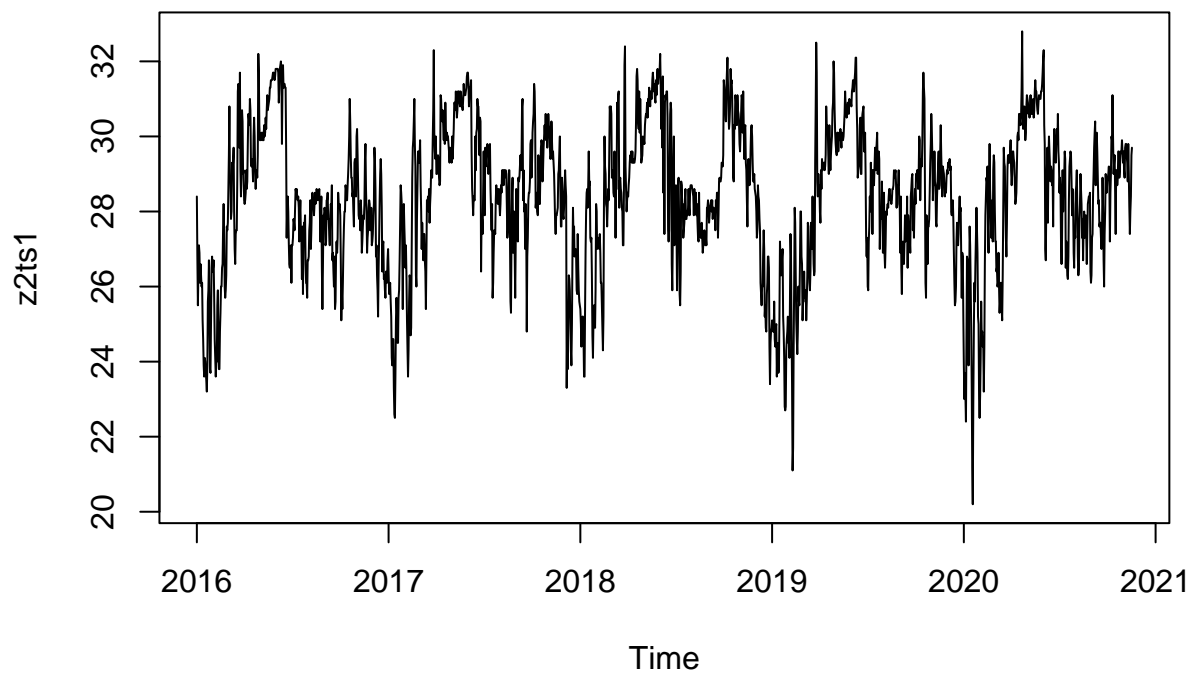
Zbiór 1 zestaw 4

```
plot(z1ts4)
```



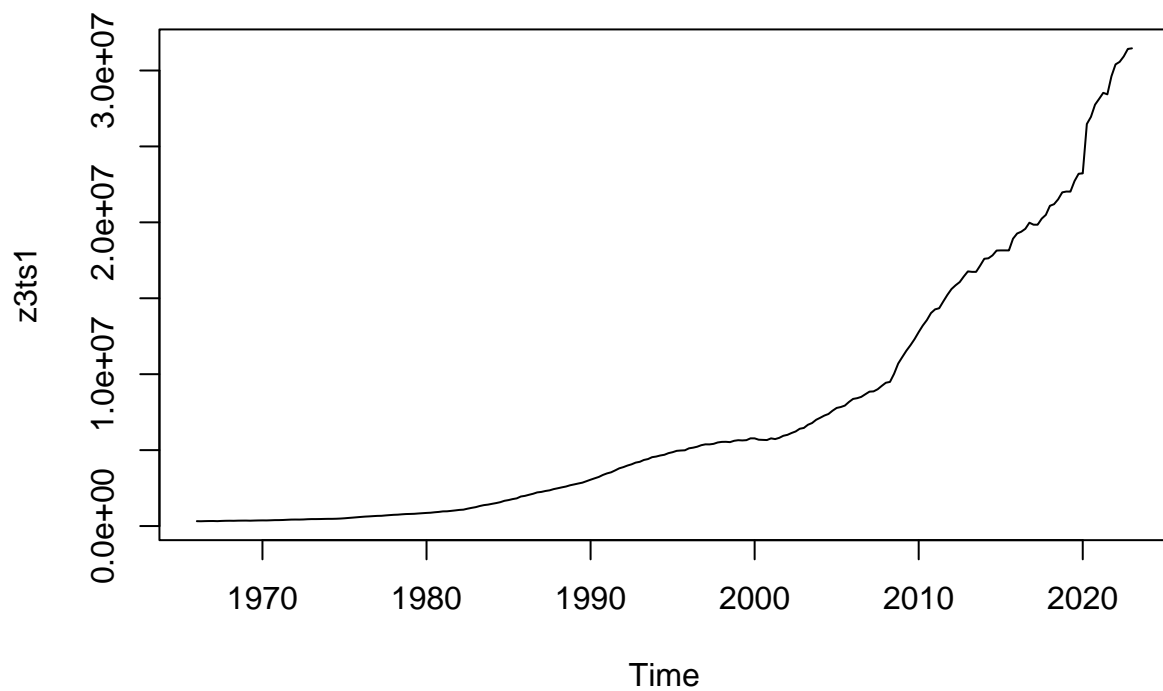
Zbiór 2

```
plot(z2ts1)
```



Zbiór 3

```
plot(z3ts1)
```

Przeprowadzenie testów

Autokorelacja

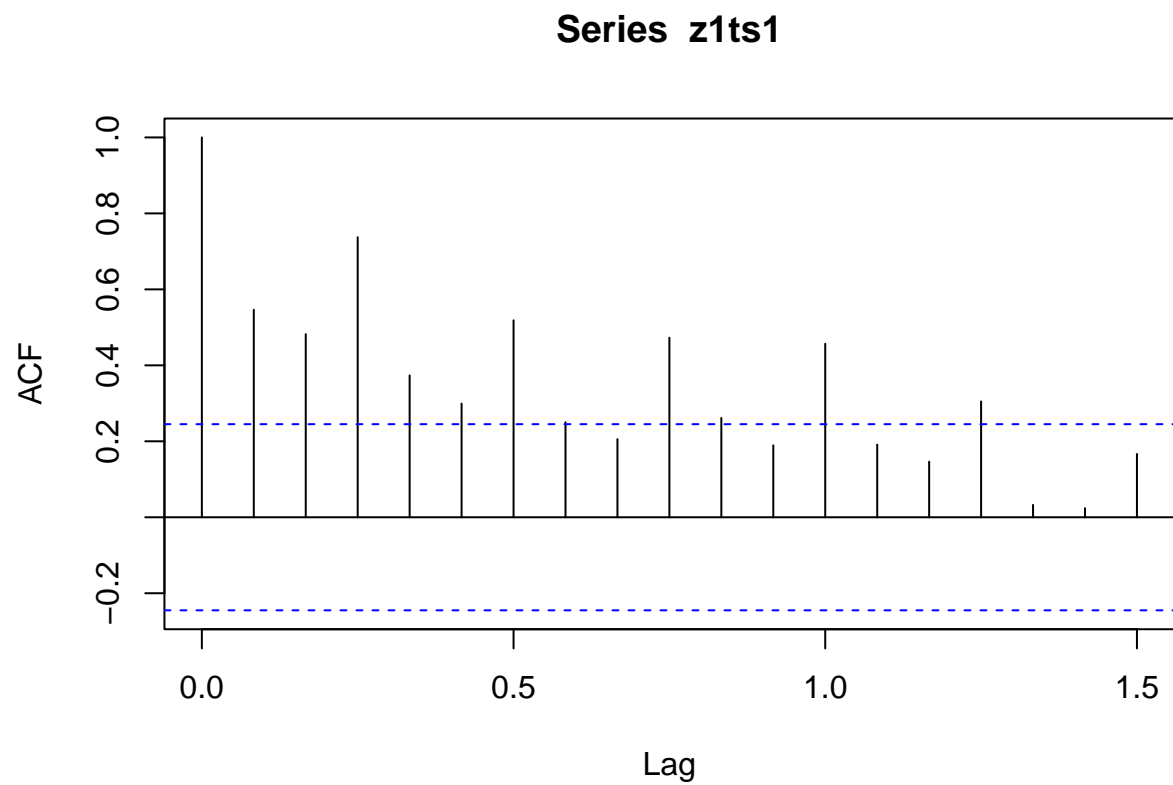
Przeprowadzimy teraz test na autokorelację, za pomocą korelogramów oraz testu Durбина-Watsona

Zbiór 1

Dla zestawu 1

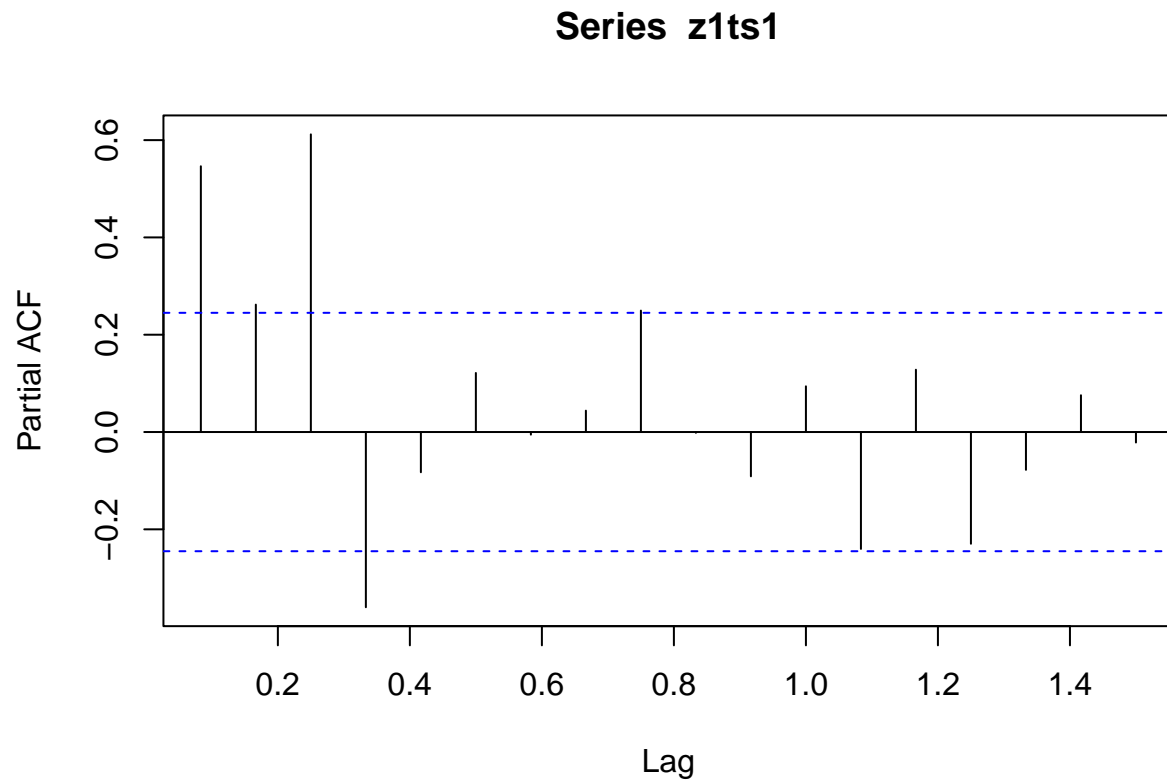
Autokorelacja przychodów przedsiębiorstwa.

```
acf(z1ts1)
```



Autokorelacja cząstkowa przychodów przedsiębiorstwa.

```
pacf(z1ts1)
```



Test Durbina-Watsona

```
df=data.frame(time=1:length(z1ts1),z1ts1)
z1lm1<-lm(z1ts1~time,data = df)
dwtest(z1lm1,order.by = NULL)
```

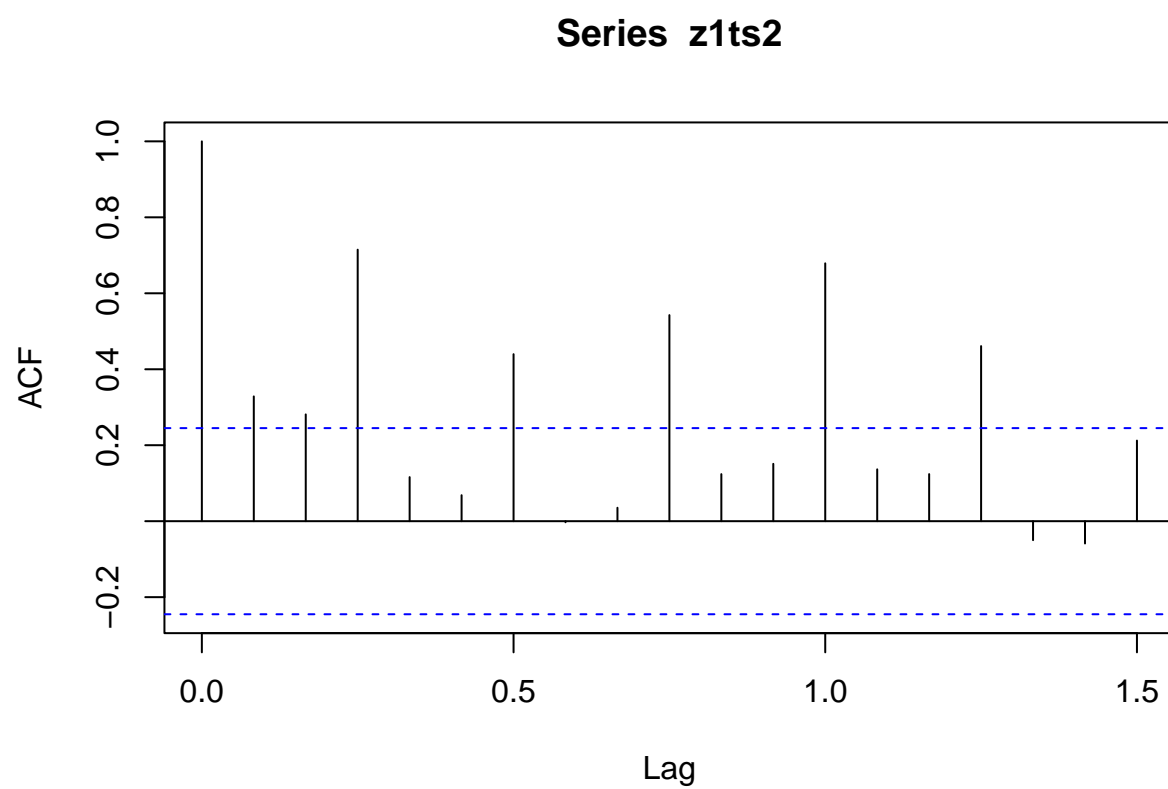
```
##
## Durbin-Watson test
##
## data: z1lm1
## DW = 2.201376609, p-value = 0.751966677
## alternative hypothesis: true autocorrelation is greater than 0
```

Z powyżej wykonanych testów wynika, że mamy do czynienia z minimalną negatywną autokorelacją, jednak nie jest ona istotna statystycznie.

Dla zestawu 2

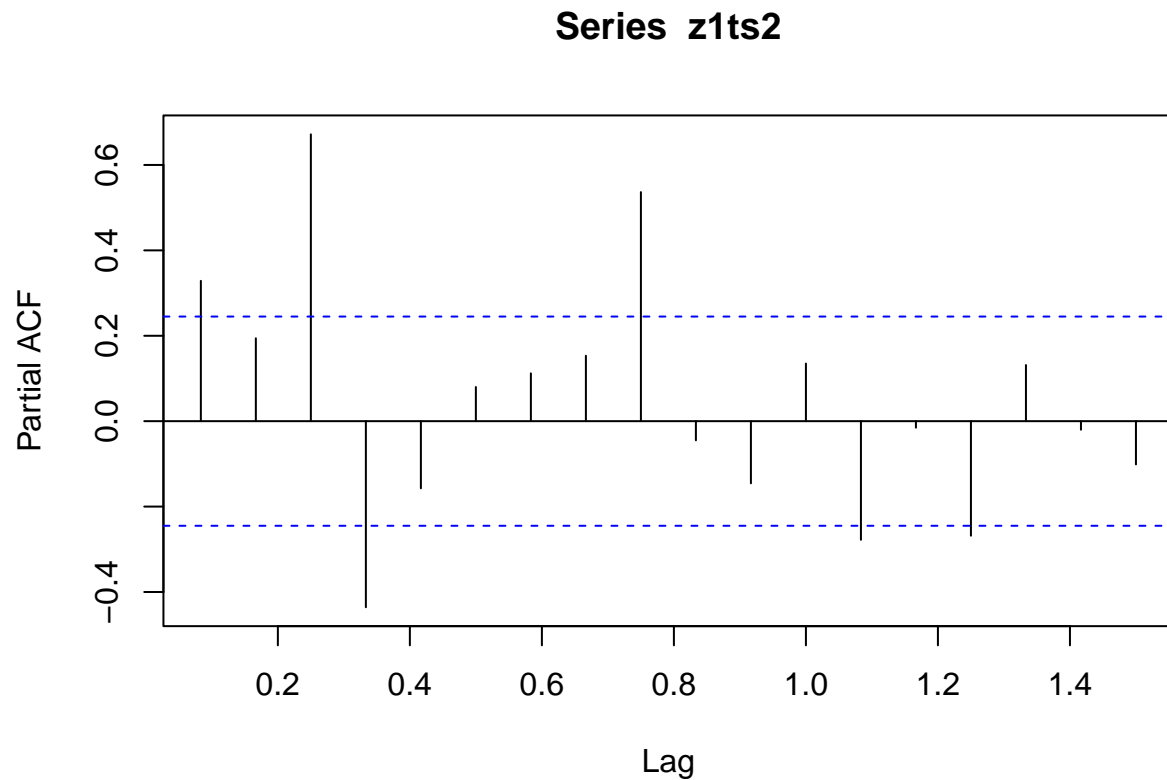
Autokorelacja ilości sprzedarzy w firmie.

```
acf(z1ts2)
```



Autokorelacja cząstkowa ilości sprzedarzy w firmie.

```
pacf(z1ts2)
```



Test Durbina-Watsona

```
df=data.frame(time=1:length(z1ts2),z1ts2)
z1lm2<-lm(z1ts2~time,data = df)
dwtest(z1lm2)
```

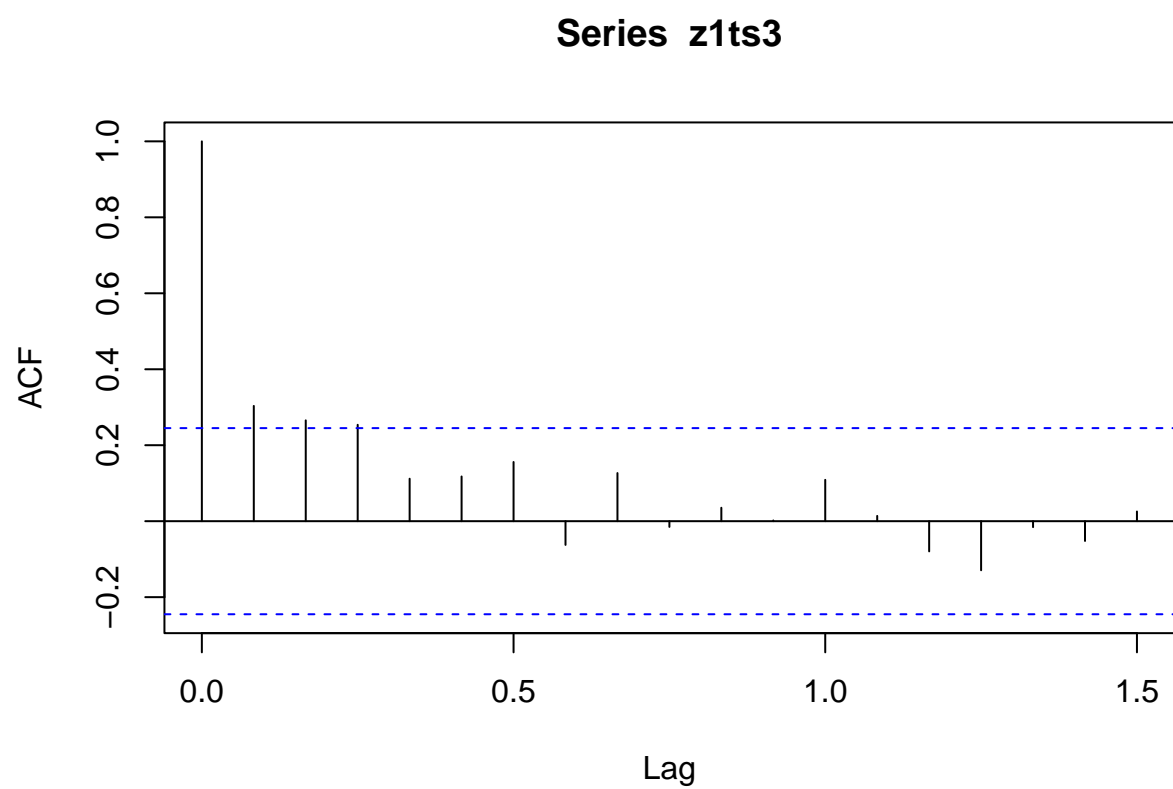
```
##
## Durbin-Watson test
##
## data:  z1lm2
## DW = 2.301678967, p-value = 0.861847786
## alternative hypothesis: true autocorrelation is greater than 0
```

Z powyżej wykonanych testów wynika, że mamy doczynienia z minimalną negatywną autokorelacją, jednak nie jest ona istotna statystycznie.

Dla zestawu 3

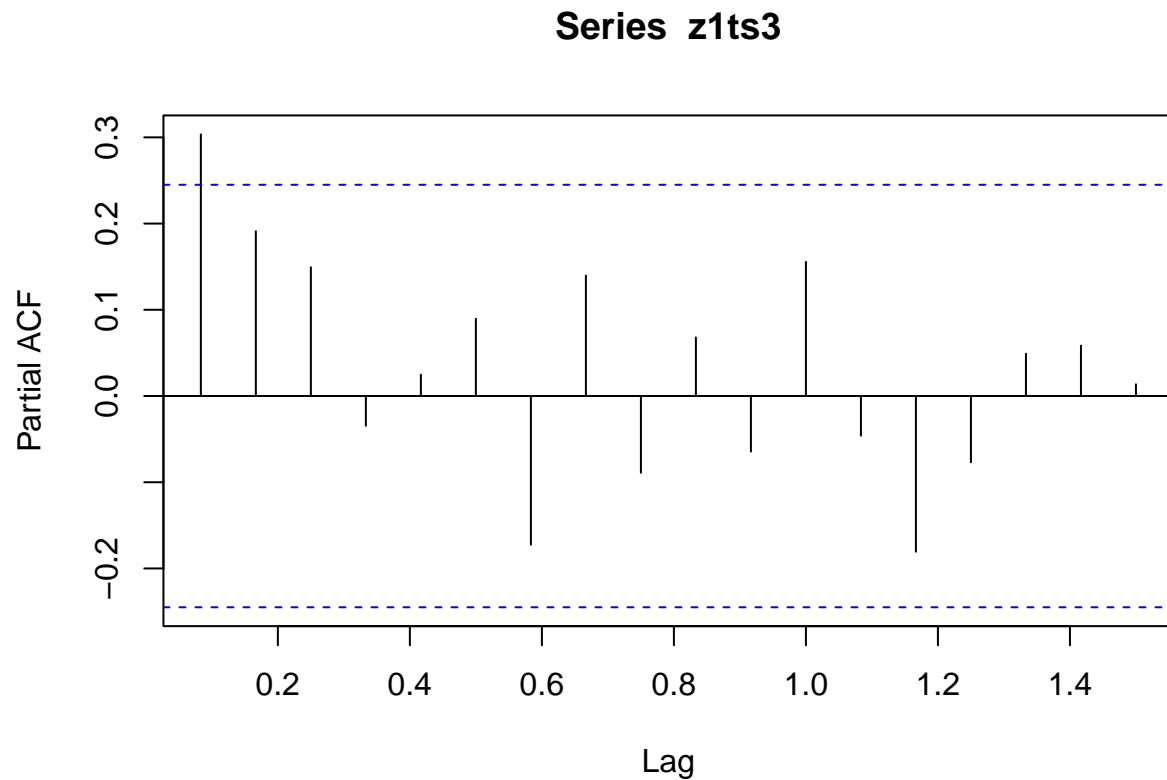
Autokorelacja średniego kosztu produkcji w firmie.

```
acf(z1ts3)
```



Autokorelacja średniego kosztu produkcji w firmie.

```
pacf(z1ts3)
```



Test Durbina-Watsona

```
df=data.frame(time=1:length(z1ts3),z1ts3)
z1lm3<-lm(z1ts3~time,data = df)
dwtest(z1lm3)
```

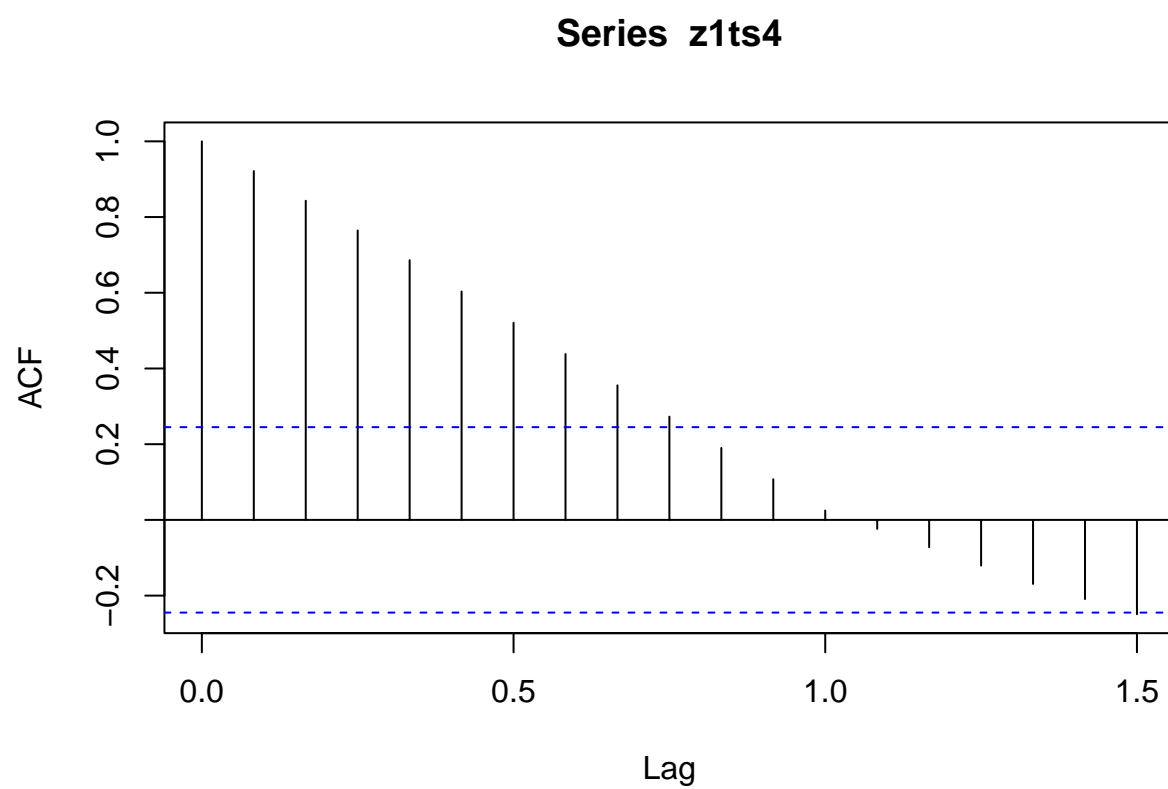
```
##
## Durbin-Watson test
##
## data: z1lm3
## DW = 1.59917854, p-value = 0.0392624752
## alternative hypothesis: true autocorrelation is greater than 0
```

Z powyżej wykonanych testów wynika, że mamy doczynienia z minimalną pozytywną autokorelacją i jest ona istotna statystycznie.

Dla zestawu 4

Autokorelacja informacji o średniej liczbie pracowników w regionie.

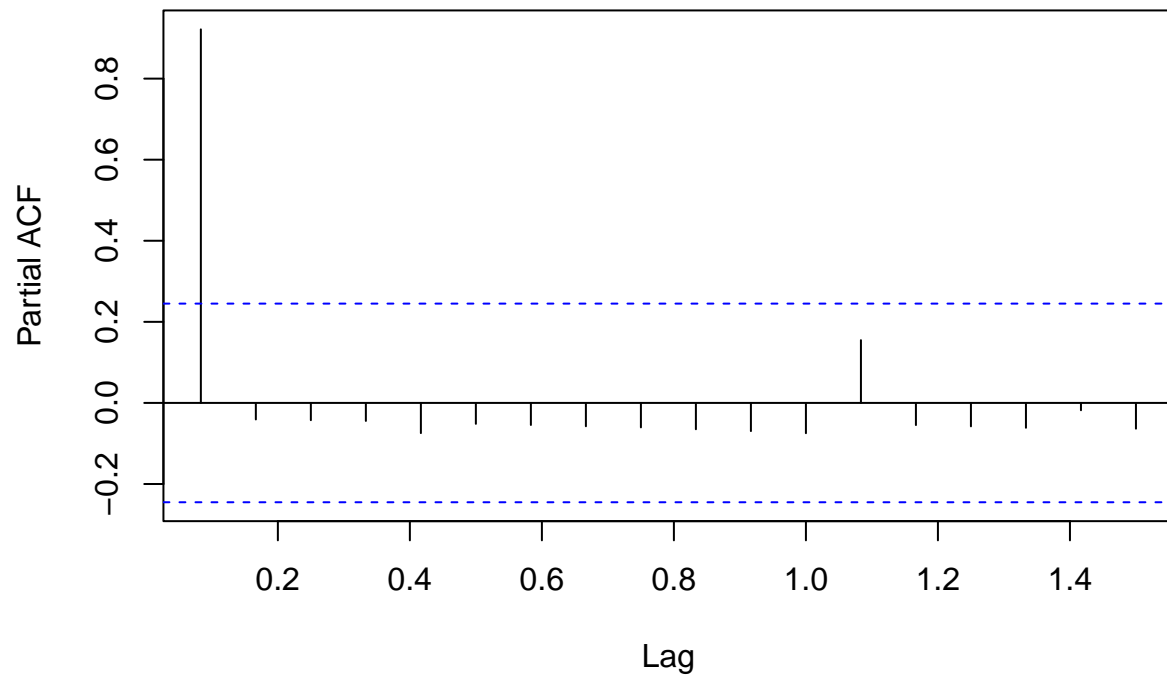
```
acf(z1ts4)
```



Autokorelacja informacji o średniej liczbie pracowników w regionie.

```
pacf(z1ts4)
```


Series z1ts4



Test Durbina-Watsona

```
df=data.frame(time=1:length(z1ts4),z1ts4)
z1lm4<-lm(z1ts4~time,data = df)
dwtest(z1lm4)
```

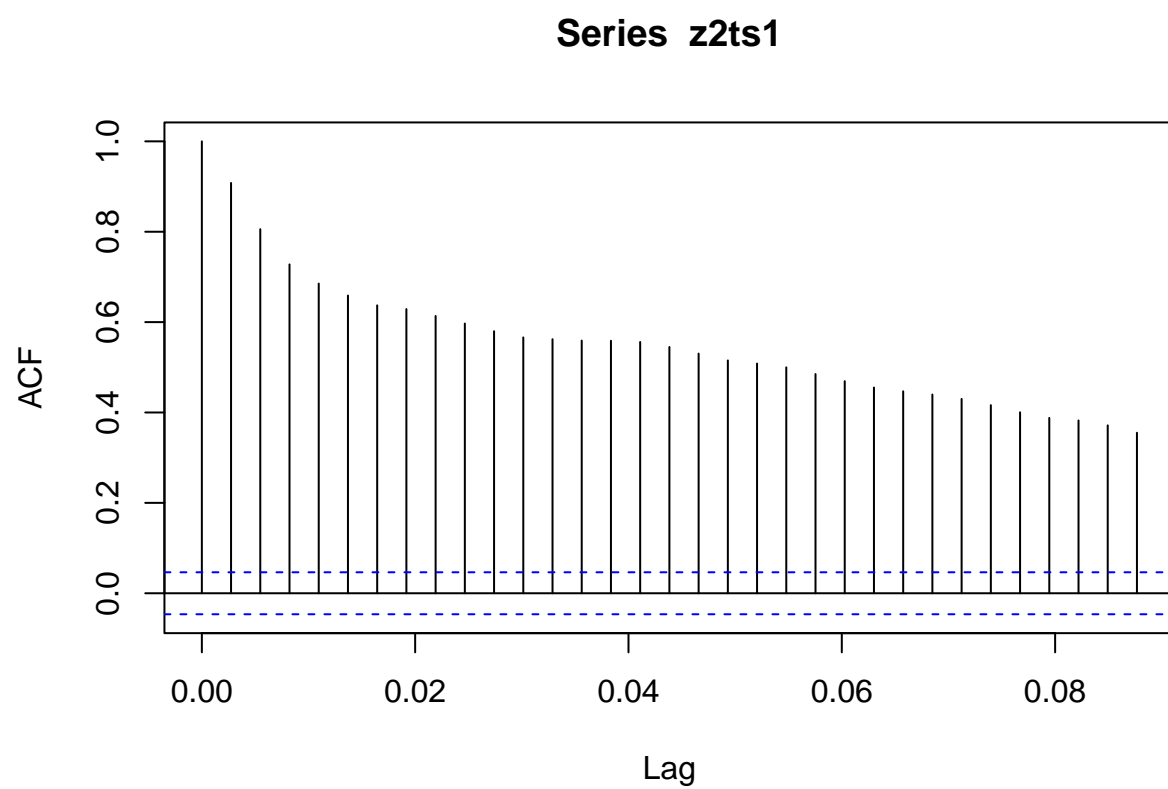
```
##
## Durbin-Watson test
##
## data: z1lm4
## DW = 0.1302507942, p-value < 2.220446e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Z powyżej wykonanych testów wynika, że mamy doczynienia z silną pozytywną autokorelacją, która jest istotna statystycznie

Dla zbioru 2

Autokorelacja średniej dziennej temperatury w Mumbaiu.

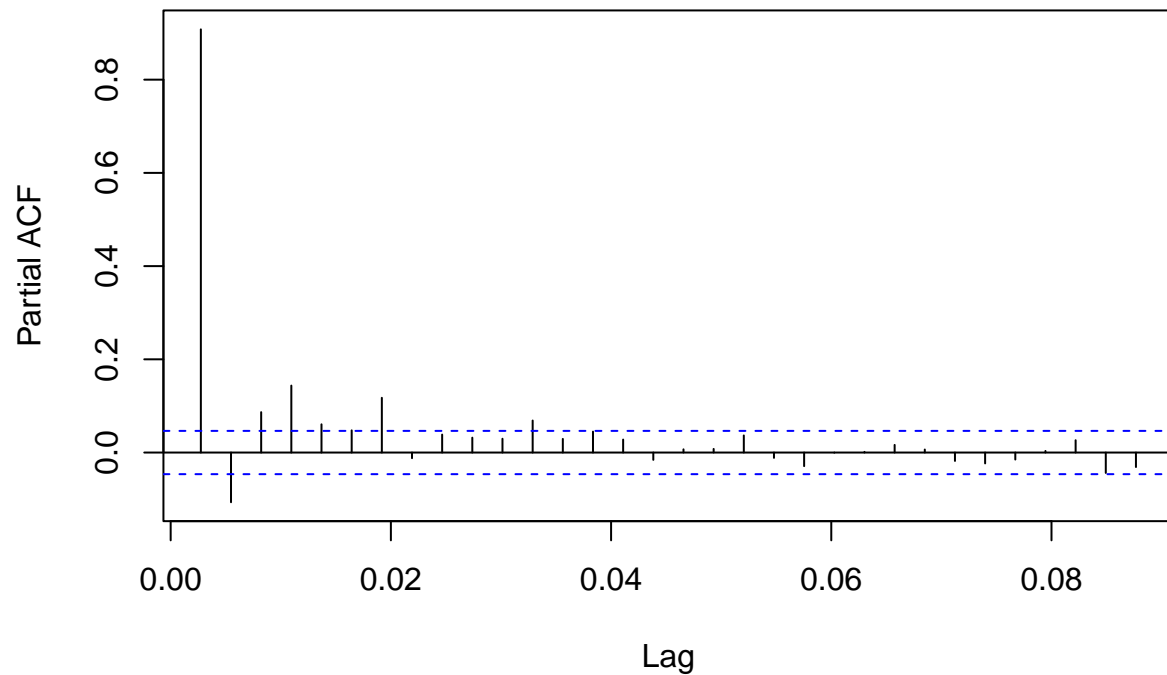
```
acf(z2ts1)
```



Autokorelacja średniej dziennej temperatury w Mumbaiu.

```
pacf(z2ts1)
```

Series z2ts1



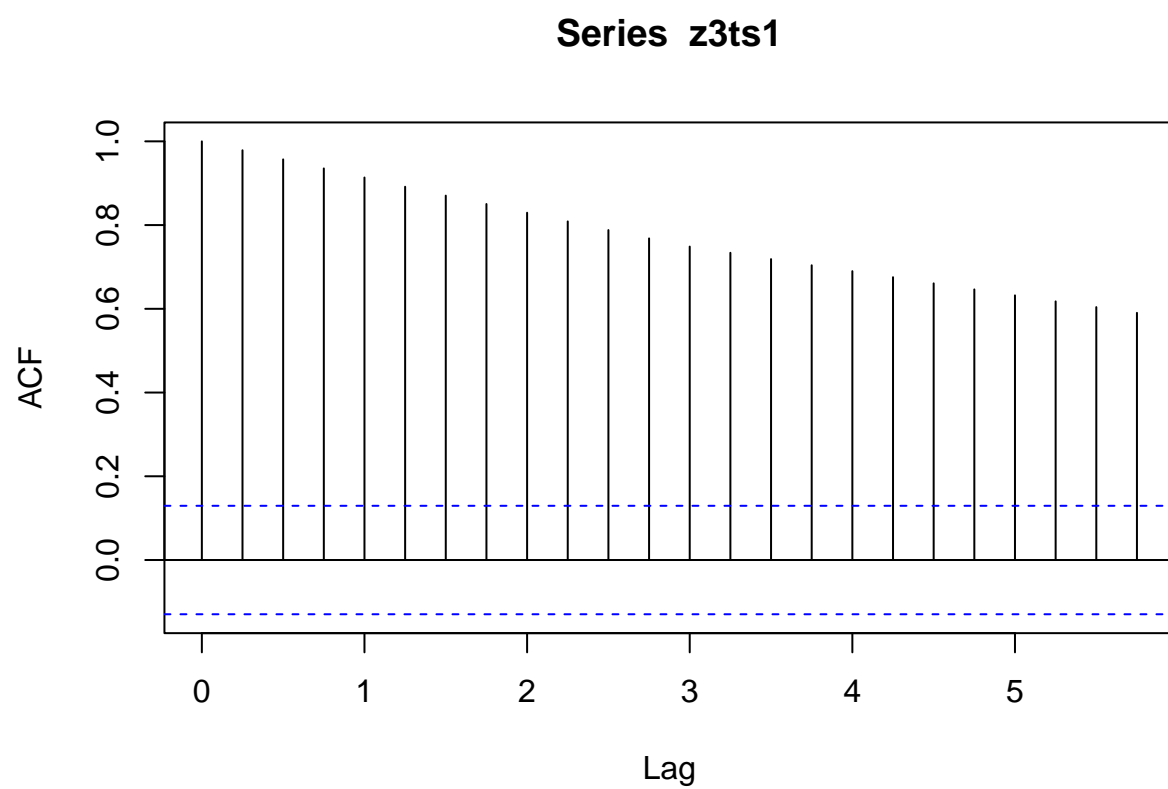
Test Durbina-Watsona

```
df=data.frame(time=1:length(z2ts1),z2ts1)
z2lm1<-lm(z2ts1~time,data = df)
dwtest(z2lm1)
```

```
##
## Durbin-Watson test
##
## data: z2lm1
## DW = 0.1842113922, p-value < 2.220446e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Z powyżej wykonanych testów wynika, że mamy do czynienia z silną pozytywną autokorelacją, która jest istotna statystycznie ## Dla zbioru 3 Autokorelacja kwartalnych danych o długu publicznym USA.

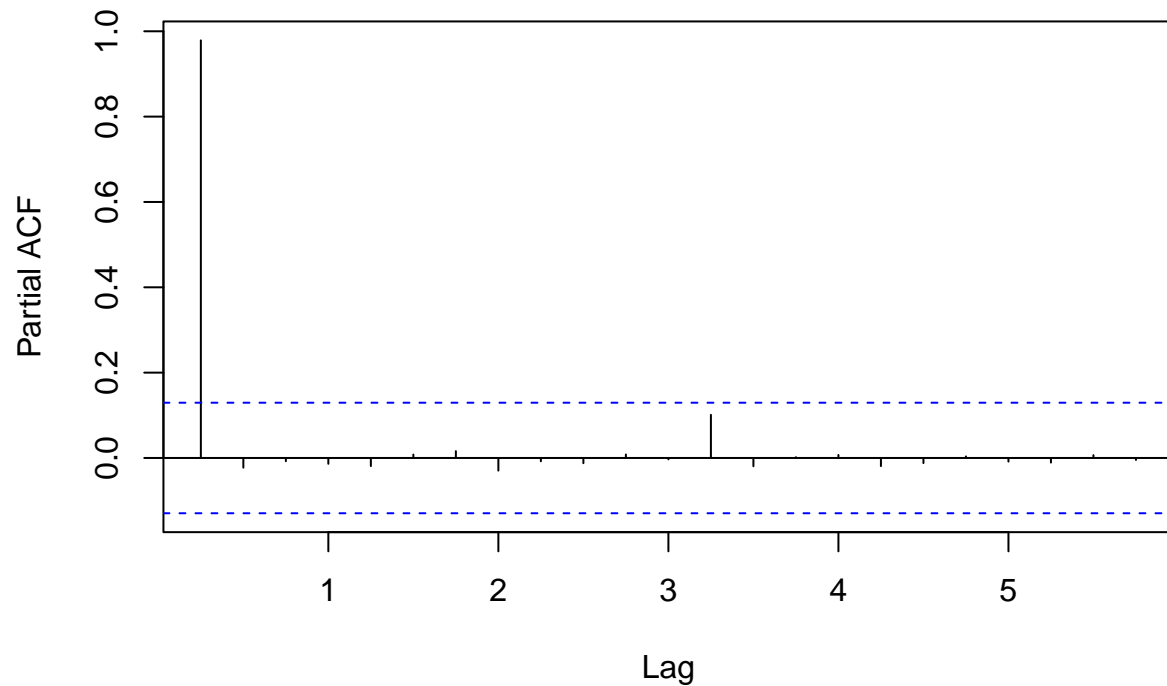
```
acf(z3ts1)
```



Autokorelacja kwartalnych danych o długu publicznym USA.

```
pacf(z3ts1)
```

Series z3ts1



Test Durbina-Watsona

```
df=data.frame(time=1:length(z3ts1),z3ts1)
z3lm1<-lm(z3ts1~time,data = df)
dwtest(z3lm1)
```

```
##
## Durbin-Watson test
##
## data: z3lm1
## DW = 0.005619120769, p-value < 2.220446e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Z powyżej wykonanych testów wynika, że mamy doczynienia z silną pozywną autokorelacją, która jest istotna statystycznie

Test na heteroskedastyczność

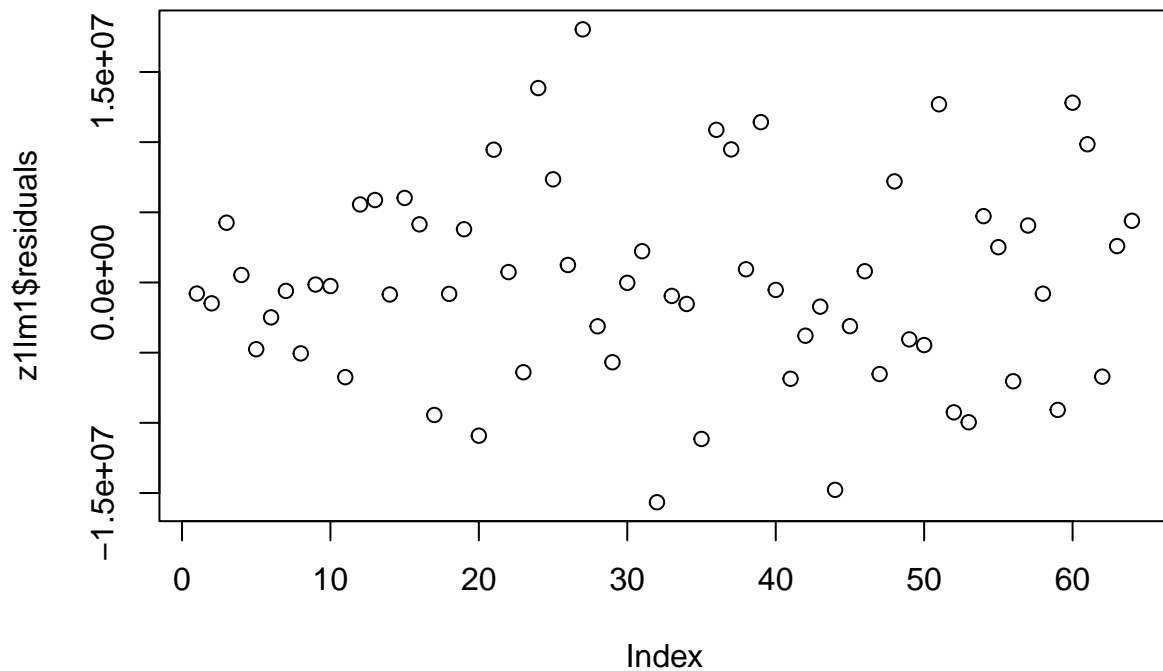
Aby określić heteroskedastyczność szeregu, należy najpierw stworzyć model liniowy z naszych szeregów czasowych, a następnie przeprowadzić test Breuscha-Pagana.

Zbiór 1

Dla zestawu 1

Heteroskedastyczność przychodów przedsiębiorstwa.

```
plot(z1lm1$residuals)
```



```
bptest(z1lm1)
```

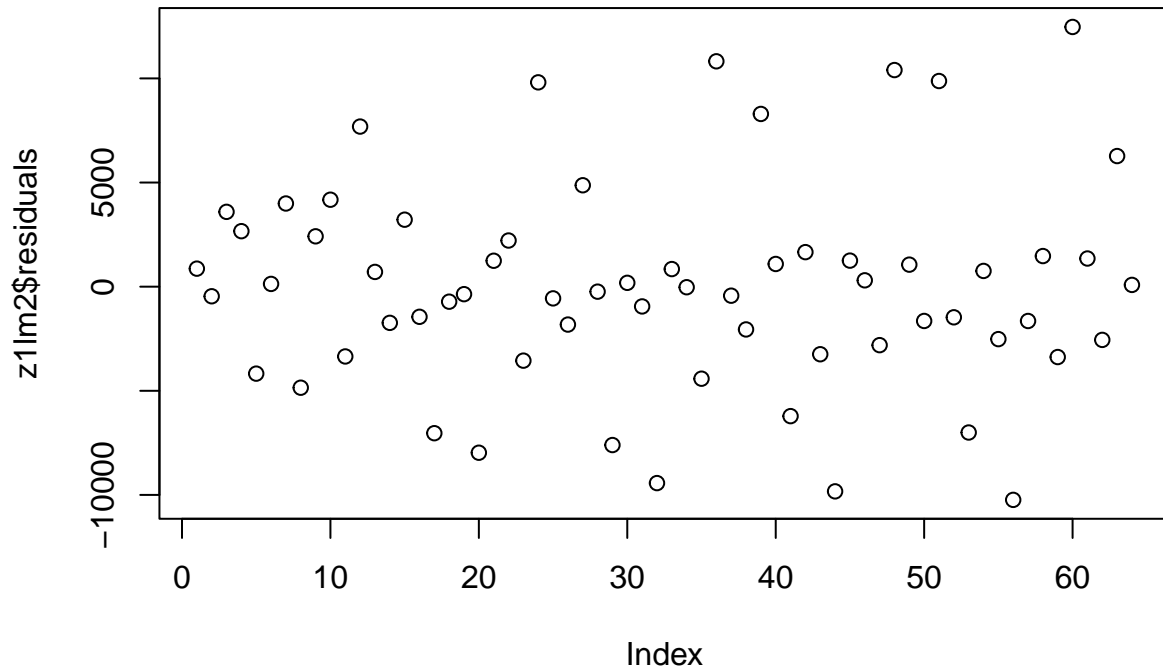
```
##  
## studentized Breusch-Pagan test  
##  
## data: z1lm1  
## BP = 1.953845408, df = 1, p-value = 0.162173073
```

Wykonując powyższy test wynika, że wartość p jest mniejsze niż 0,05, co prowadzi do nieodrzucenia hipotezy zerowej. W konsekwencji, *nie posiadamy wystarczających dowodów na występowanie heteroskedastyczności.*

Dla zestawu 2

Heteroskedastyczność ilości sprzedarzy w firmie.

```
plot(z1lm2$residuals)
```



```
bptest(z1lm2)
```

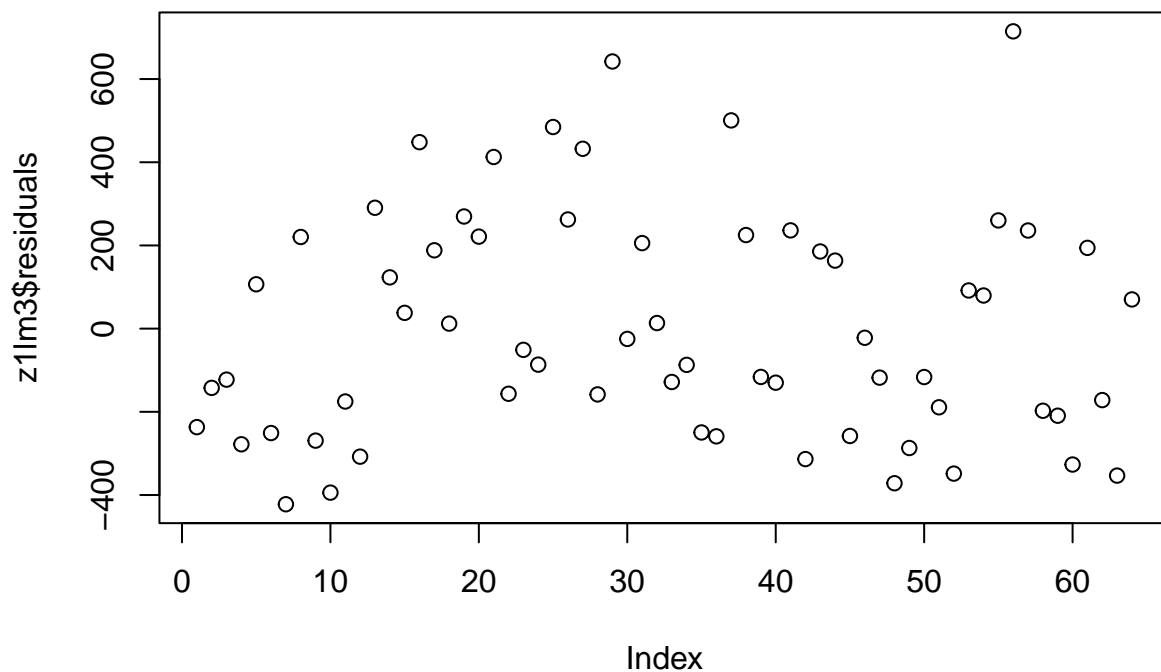
```
##  
## studentized Breusch-Pagan test  
##  
## data: z1lm2  
## BP = 2.573564, df = 1, p-value = 0.10866267
```

Wykonując powyższy test wynika, że wartość p jest większa niż 0,05, co prowadzi do nieodrzućenia hipotezy zerowej. W konsekwencji, *nie posiadamy wystarczających dowodów na występowanie heteroskedastyczności*.

Dla zestawu 3

Heteroskedastyczność średniego kosztu produkcji w firmie.

```
df=data.frame(time=1:length(z1ts3),z1ts3)  
z1lm3<-lm(z1ts3~time,data = df)  
plot(z1lm3$residuals)
```



```
bptest(z1lm3)
```

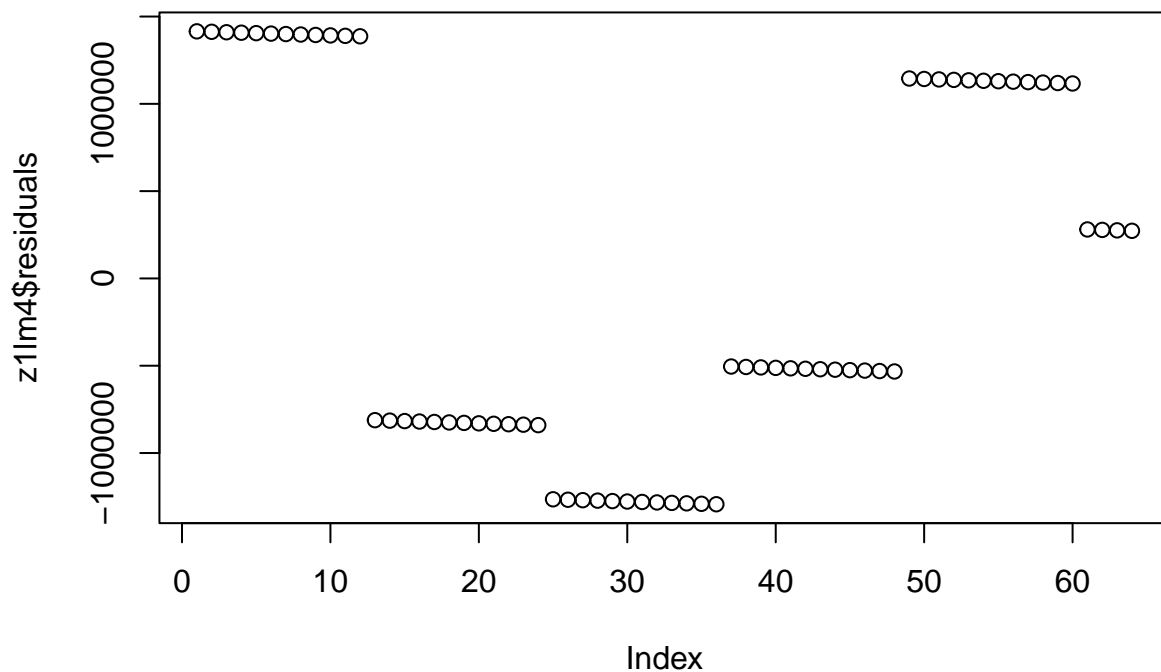
```
##
## studentized Breusch-Pagan test
##
## data: z1lm3
## BP = 0.04862771469, df = 1, p-value = 0.825468592
```

Wykonując powyższy test wynika, że wartość p jest mniejsze niż 0,05, co prowadzi do nieodrzućenia hipotezy zerowej. W konsekwencji, *nie posiadamy wystarczających dowodów na występowanie heteroskedastyczności.*

Dla zestawu 4

Heteroskedastyczność informacji o średniej liczbie pracowników w regionie.

```
df=data.frame(time=1:length(z1ts4),z1ts4)
z1lm4<-lm(z1ts4~time,data = df)
plot(z1lm4$residuals)
```

```
bptest(z1lm4)
```

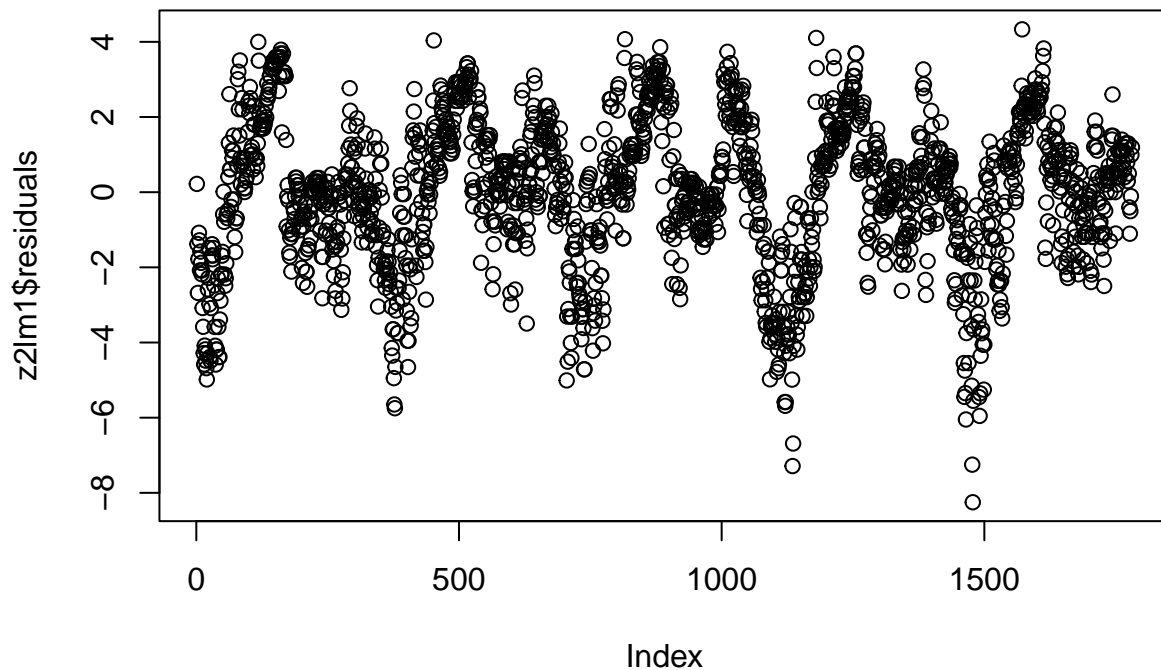
```
##
##  studentized Breusch-Pagan test
##
## data:  z1lm4
## BP = 16.05643948, df = 1, p-value = 6.14821832e-05
```

Wykonując powyższy test wynika, że wartość p jest większa niż 0,05, co prowadzi do odrzucenia hipotezy zerowej. W konsekwencji, *możemy stwierdzić, że występuje heteroskedastyczność.*

Zbiór 2

Heteroskedastyczność średniej dziennej temperatury w Mumbaiu.

```
df=data.frame(time=1:length(z2ts1),z2ts1)
z2lm1<-lm(z2ts1~time,data = df)
plot(z2lm1$residuals)
```



```
bptest(z2lm1)
```

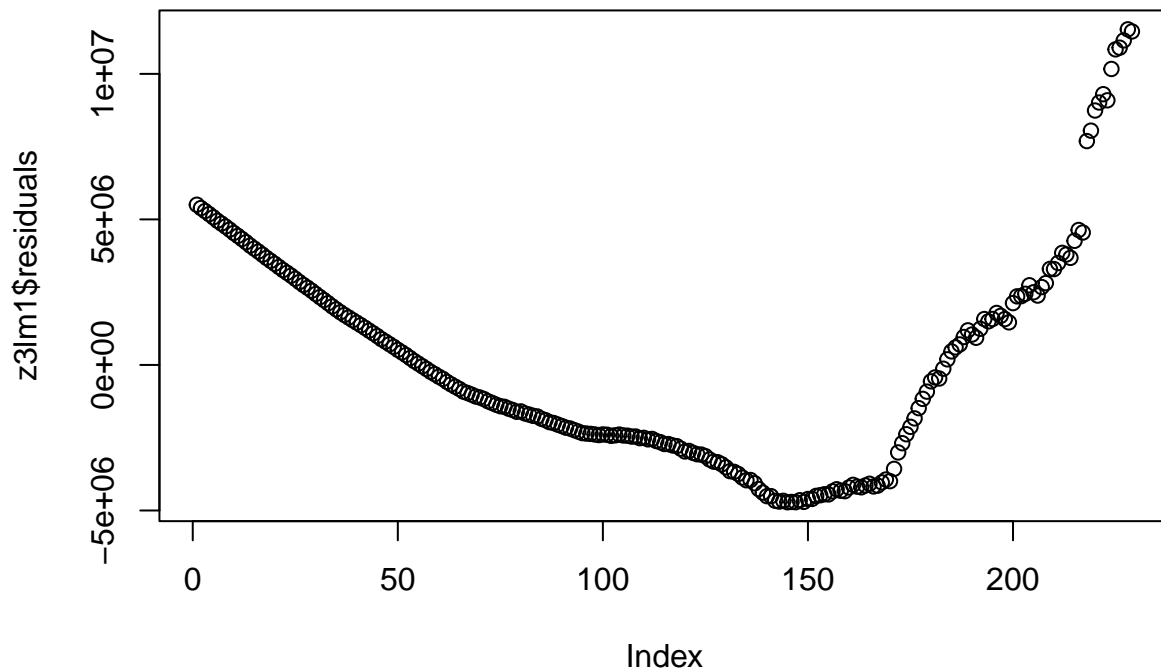
```
##
## studentized Breusch-Pagan test
##
## data:  z2lm1
## BP = 3.859278805, df = 1, p-value = 0.0494715892
```

Wykonując powyższy test wynika, że wartość p jest większa niż 0,05, co prowadzi do odrzucenia hipotezy zerowej. W konsekwencji, *możemy stwierdzić, że występuje heteroskedastyczność.*

Zbiór 3

Heteroskedastyczność kwartalnych danych o długu publicznym USA.

```
df=data.frame(time=1:length(z3ts1),z3ts1)
z3lm1<-lm(z3ts1~time,data = df)
plot(z3lm1$residuals)
```



```
bptest(z3lm1)
```

```
##
## studentized Breusch-Pagan test
##
## data:  z3lm1
## BP = 30.18331407, df = 1, p-value = 3.9307678e-08
```

Wykonując powyższy test wynika, że wartość p jest większa niż 0,05, co prowadzi do odrzucenia hipotezy zerowej. W konsekwencji, *możemy stwierdzić, że występuje heteroschedastyczność.*

Test na stacjonarność szergu

Dla zbioru 1 seria 1

```
urca::ur.kpss(z1ts1) %>% summary() #niestacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
```

```
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 1.4597
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

diff(z1ts1) %>% urca::ur.kpss() %>% summary() # stacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0268
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Nasze dane nie był ## Dla zbioru 1 serii 2

```
urca::ur.kpss(z1ts2) %>% summary() #niestacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 1.3016
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
diff(z1ts2) %>% urca::ur.kpss() %>% summary() # stacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0226
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Dla zbioru 1 serii 3

```
urca::ur.kpss(z1ts3) %>% summary() #niestacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.5867
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
diff(z1ts3) %>% urca::ur.kpss() %>% summary() # stacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0388
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

##Dla zbioru 1 serii 4

```
urca::ur.kpss(z1ts4) %>% summary() #stacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.3626
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

##Dla zbioru 2 serii 1

```
urca::ur.kpss(z2ts1) %>% summary() #stacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 8 lags.
##
## Value of test-statistic is: 0.1644
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
##Dla zbioru 3 serii 1
```

```
urca::ur.kpss(z3ts1) %>% summary() #niestacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 3.8553
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
diff(z3ts1,lag=1) %>% urca::ur.kpss() %>% summary() # niestacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 2.3749
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
residuals(z3lm1) %>% urca::ur.kpss() %>% summary() # niestacjonarny
```

```
##
```

```
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.9918
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
log(z3ts1) %>% urca::ur.kpss() %>% summary()# niestacjonarny
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 4.6172
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

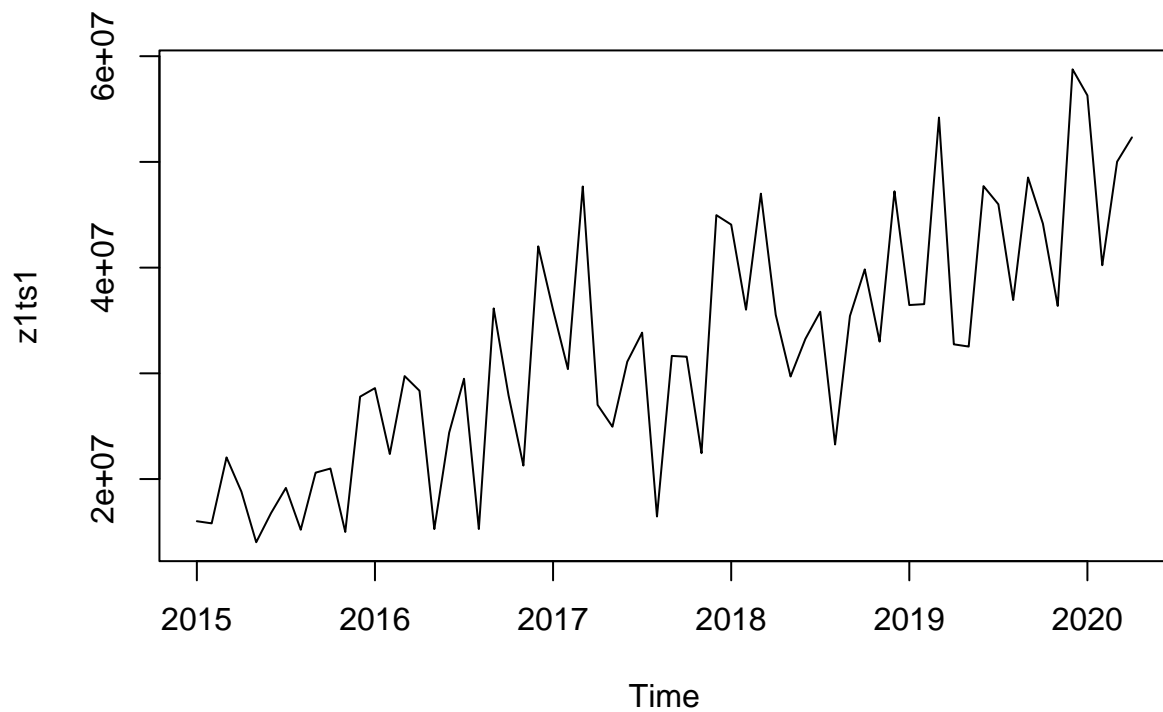
W szeregach z1ts1, z1ts2, z1ts3 mieliśmy doczynienia z szeregami niestacjonarnymi, które udało nam się sprowadzić do postaci stacjonarnej, szereg z1ts4, z2ts1 są stacjonarne, natomiast z3ts1 okazał się być niestacjonarny i nie można go sprowadzić do postaci stacjonarnej w prosty sposób.

Tworzenie modelu

Zbiór 1 seria 1

Bazując na naszych wcześniejszych danych musimy teraz dobrać odpowiednie parametry naszego modelu. Jako iż będziemy korzystać z modelu Arima potrzebujemy wartości parametrów p,d,q.

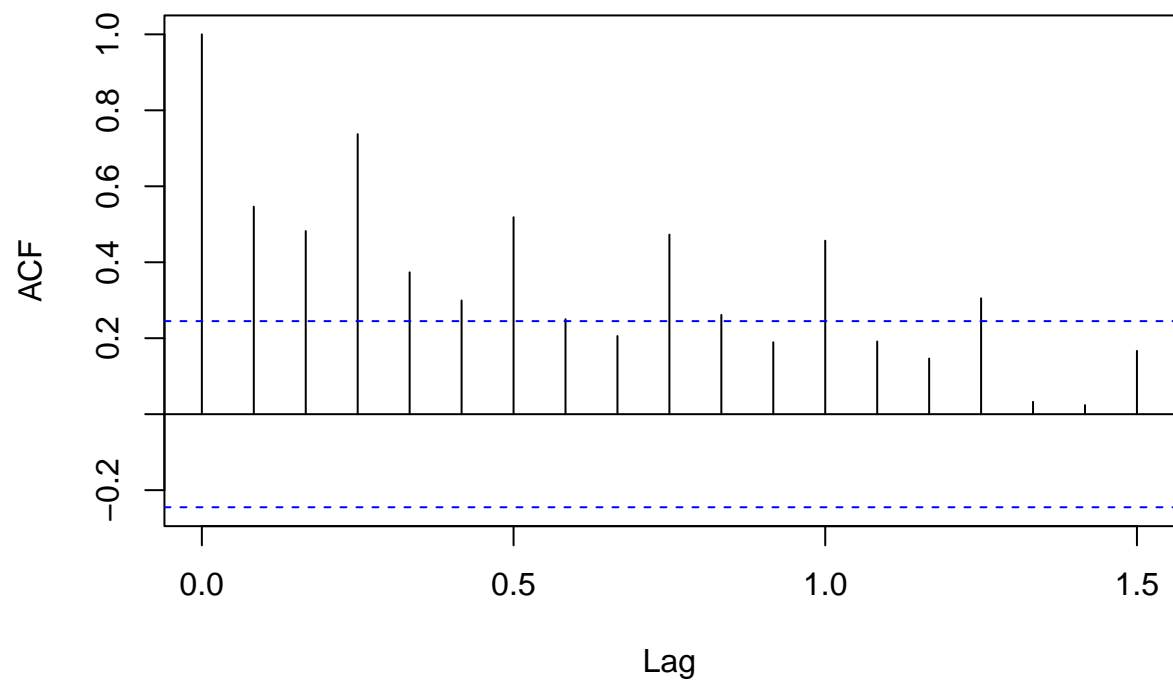
```
plot(z1ts1) #roboczo
```



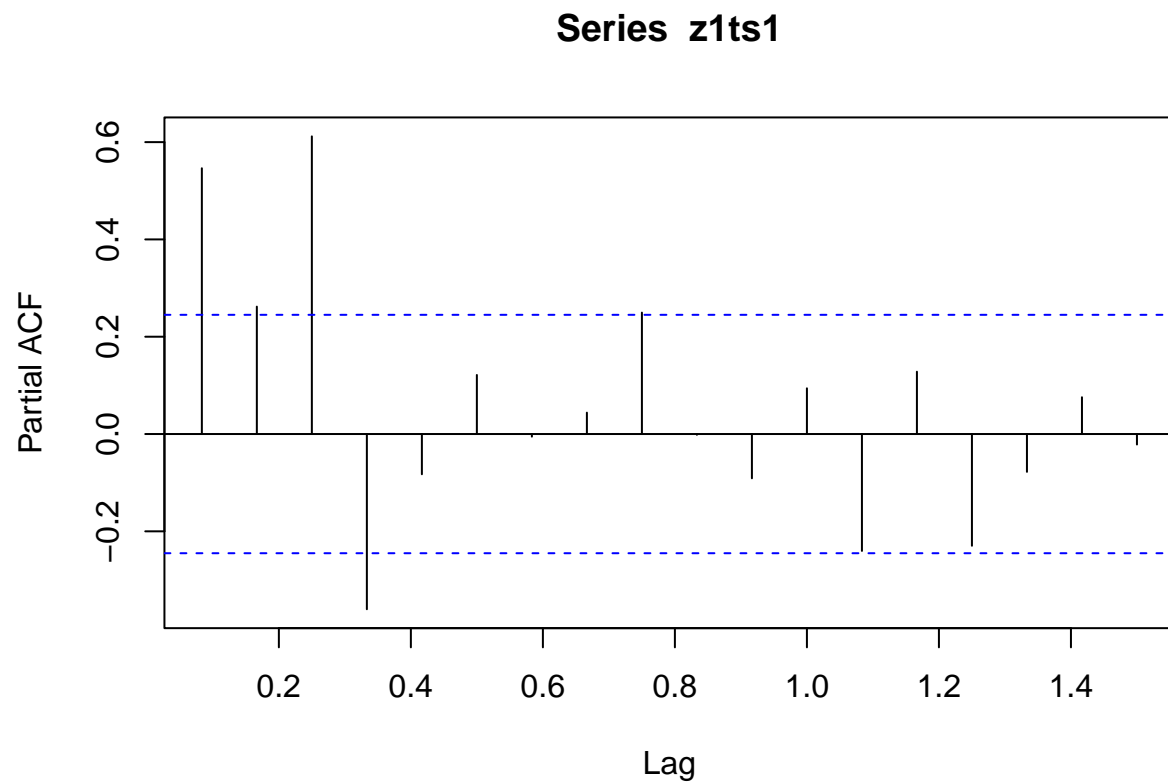
Aby wyznaczyć parametr p patrzymy na nasze korelogramy.

```
acf(z1ts1)
```


Series z1ts1



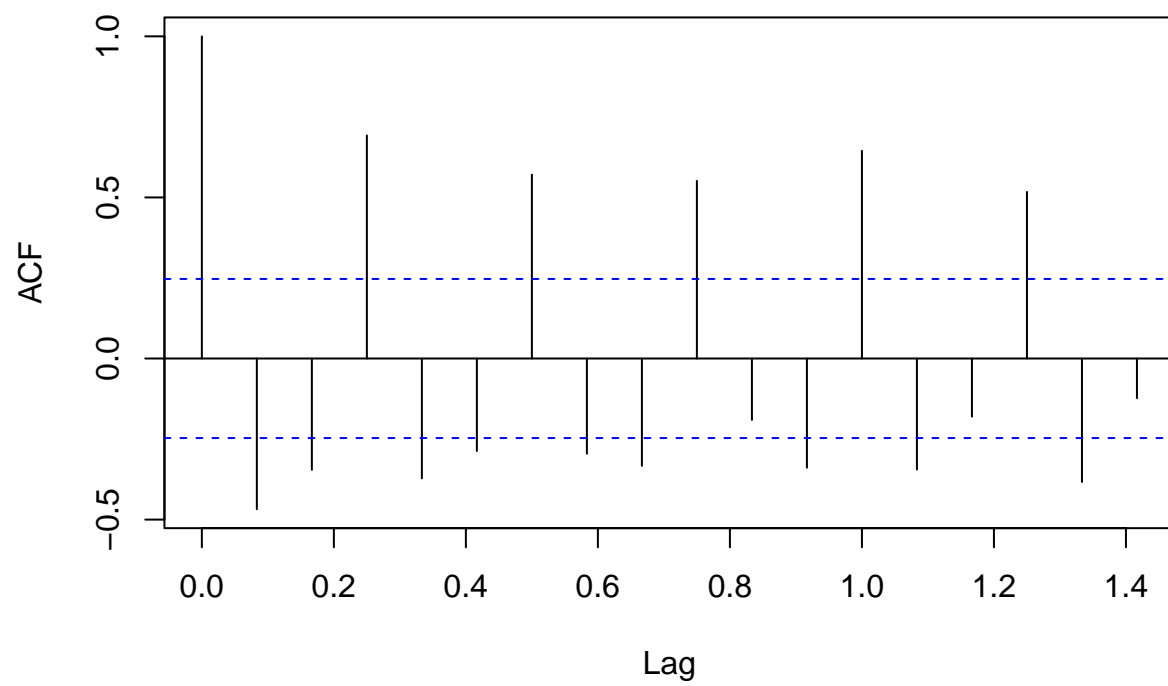
```
pacf(z1ts1)
```



Możemy na nich zauważyć, że nie występuje jednoznaczna autokorelacja. W takim wypadku możemy nasze dane zróżnicować

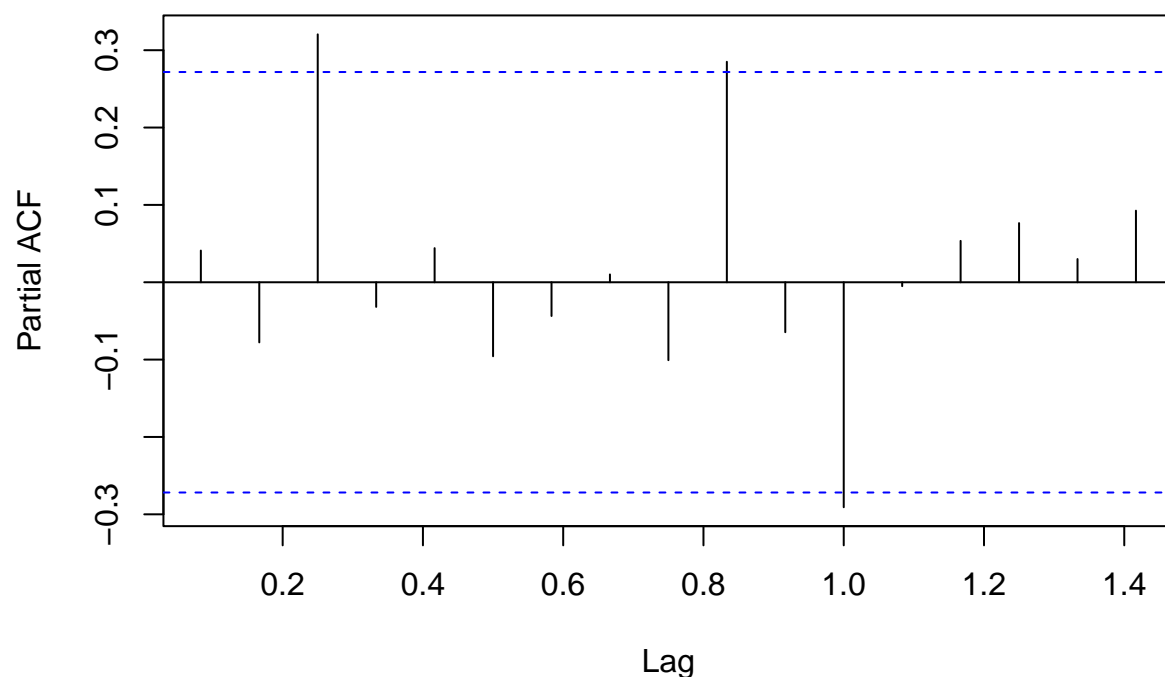
```
acf(diff(z1ts1,lag = 1))
```

Series diff(z1ts1, lag = 1)



```
pacf(diff(z1ts1, lag = 12))
```

Series diff(z1ts1, lag = 12)



Na podstawie powyższych wykresów mając orientację jakie wartości możemy wrzucić do modelu testujemy kilka opcji

```
print("Wersja 1")
```

```
## [1] "Wersja 1"
```

```
Arima(y=z1ts1, order = c(3,1,3),lambda = NULL)
```

```
## Series: z1ts1
## ARIMA(3,1,3)
##
## Coefficients:
##          ar1          ar2          ar3          ma1          ma2
##    -0.760086155  -0.766437955  0.232583729  0.233656795  0.35649738
## s.e.   0.262887081   0.260511953  0.260991861  0.229629835  0.21278346
##          ma3
##    -0.693226700
## s.e.   0.229245461
##
## sigma^2 = 33530603079204: log likelihood = -1071.24
## AIC=2156.49  AICc=2158.52  BIC=2171.49
```

```
print("Wersja 2")
```

```
## [1] "Wersja 2"
```

```
Arima(y=z1ts1,lambda = NULL,seasonal = c(3,1,3))
```

```
## Series: z1ts1
## ARIMA(0,0,0)(3,1,3)[12]
##
## Coefficients:
##          sar1          sar2          sar3          sma1          sma2
##    0.492440620  0.964345776 -0.469791583 -0.834397448 -0.775037701
## s.e.  1.586720318  0.699113912  1.461010870  3.107421562  1.914644422
##          sma3
##    0.784376344
## s.e.  2.542298555
##
## sigma^2 = 35684037515948: log likelihood = -897.43
## AIC=1808.86  AICc=1811.41  BIC=1822.52
```

```
print("Wersja 3")
```

```
## [1] "Wersja 3"
```

```
Arima(y=z1ts1, order = c(0,1,0),lambda = NULL)
```

```
## Series: z1ts1
## ARIMA(0,1,0)
##
## sigma^2 = 1.12400577e+14: log likelihood = -1108.52
## AIC=2219.03  AICc=2219.1  BIC=2221.17
```

```
print("Wersja 4")
```

```
## [1] "Wersja 4"
```

```
test <- Arima(y=z1ts1,lambda = NULL,seasonal = c(1,1,1))
print("Wersja 5")
```

```
## [1] "Wersja 5"
```

```
Arima(y=z1ts1, order = c(0,1,0),lambda = "auto")
```

```
## Series: z1ts1
## ARIMA(0,1,0)
## Box Cox transformation: lambda= 0.706549181705
##
## sigma^2 = 4379015456: log likelihood = -788.7
## AIC=1579.39  AICc=1579.46  BIC=1581.54
```

```
print("Wersja 6")
```

```
## [1] "Wersja 6"
```

```
z1tst1_best_model <- Arima(y=z1ts1,lambda = "auto",seasonal = c(0,1,0))  
z1tst1_best_model
```

```
## Series: z1ts1  
## ARIMA(0,0,0)(0,1,0)[12]  
## Box Cox transformation: lambda= 0.706549181705  
##  
## sigma^2 = 2904276762: log likelihood = -640.31  
## AIC=1282.62 AICc=1282.7 BIC=1284.57
```

Z spośród stworzonych modeli wybraliśmy najlepszy, teraz spróbujemy stworzyć model AutoArima i spróbujemy go porównać z obecnie najlepszym

```
auto.arima(z1ts1,d=1,max.p = 5,max.q =5,max.d = 5,seasonal = TRUE)
```

```
## Series: z1ts1  
## ARIMA(2,1,0)(1,1,0)[12]  
##  
## Coefficients:  
##          ar1          ar2          sar1  
##      -0.670729490  -0.533708478  -0.540121180  
## s.e.    0.130560826   0.134873128   0.135814968  
##  
## sigma^2 = 28227103851676: log likelihood = -863.08  
## AIC=1734.16 AICc=1735.03 BIC=1741.88
```

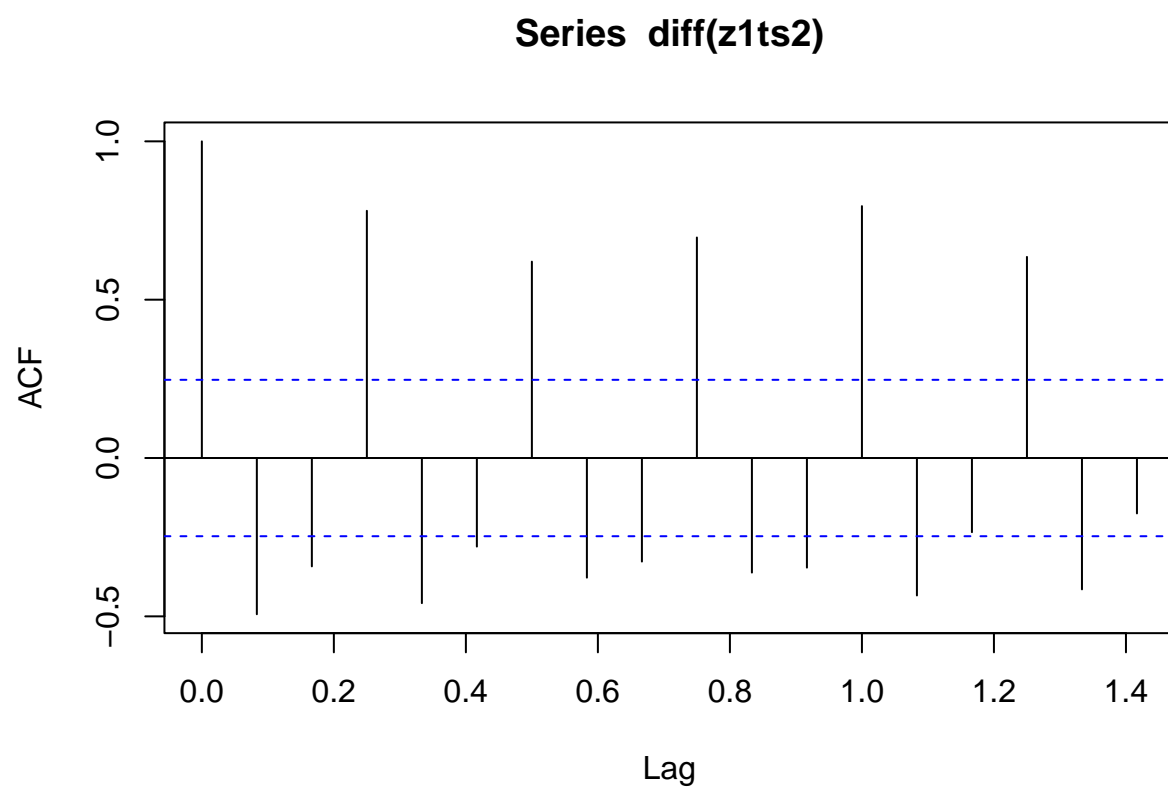
Podsumowując nasz wcześniejszy model jest najlepszy.

Teraz tworząc następne modele postępujemy analogicznie jak powyżej

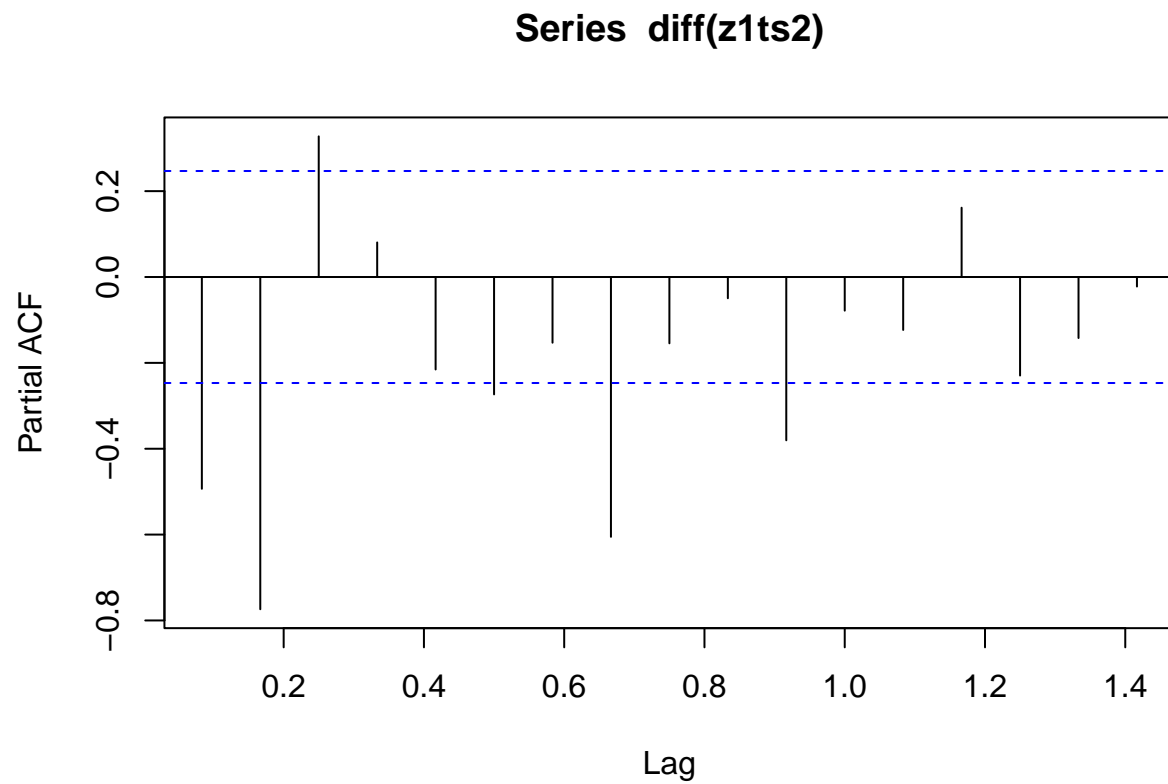
Zbiór 1 seria 2

Szacujemy nasze parametry p i q

```
acf(diff(z1ts2)) # 0 lub 1
```



```
pacf(diff(z1ts2)) # bardziej 0
```

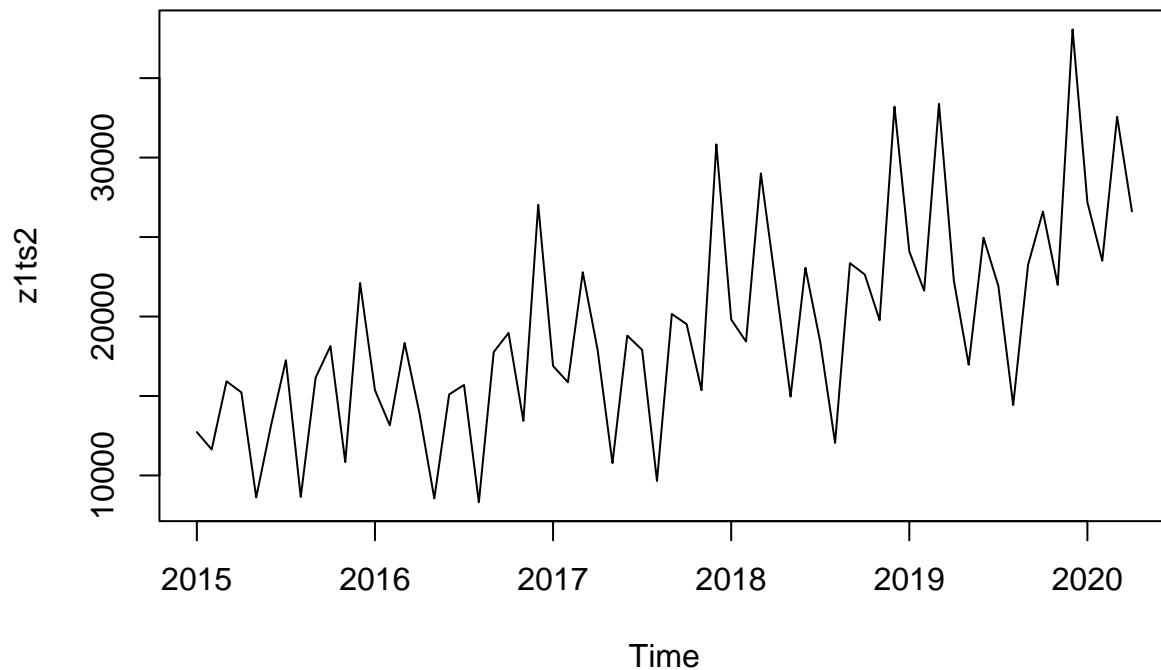


Tworzymy nasze modele i ja porównujemy

```
print("Wersja 1")
```

```
## [1] "Wersja 1"
```

```
plot(z1ts2)
```

```
Arima(y=z1ts2, order = c(0,1,0),lambda = NULL)
```

```
## Series: z1ts2
## ARIMA(0,1,0)
##
## sigma^2 = 56800927.1: log likelihood = -651.83
## AIC=1305.66 AICc=1305.72 BIC=1307.8
```

```
print("Wersja 2")
```

```
## [1] "Wersja 2"
```

```
Arima(y=z1ts2,lambda = NULL,seasonal = c(0,1,1))
```

```
## Series: z1ts2
## ARIMA(0,0,0)(0,1,1)[12]
##
## Coefficients:
##          sma1
##      0.999760336
## s.e.  0.432905161
##
## sigma^2 = 3857387.83: log likelihood = -477.6
## AIC=959.2 AICc=959.44 BIC=963.1
```

```
print("Wersja 3")
```

```
## [1] "Wersja 3"
```

```
Arima(y=z1ts2, seasonal = c(0,1,1),lambda = 'auto')
```

```
## Series: z1ts2
## ARIMA(0,0,0)(0,1,1)[12]
## Box Cox transformation: lambda= 0.24348758416
##
## Coefficients:
##             sma1
##      0.999988124
## s.e.  0.411690982
##
## sigma^2 = 1.26346924: log likelihood = -89.37
## AIC=182.75   AICc=182.99   BIC=186.65
```

```
print("Wersja 4")
```

```
## [1] "Wersja 4"
```

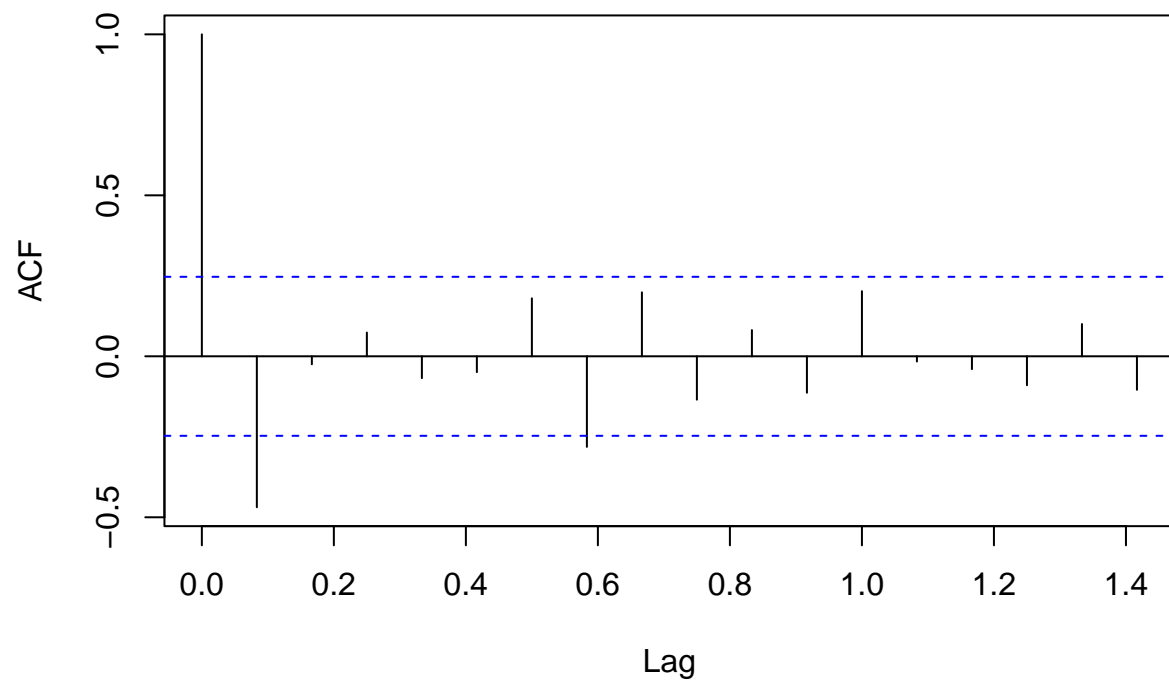
```
z1ts2_best_model<-Arima(y=z1ts2, seasonal = c(0,1,1),lambda = 'auto')
```

Zbiór 1 seria 3

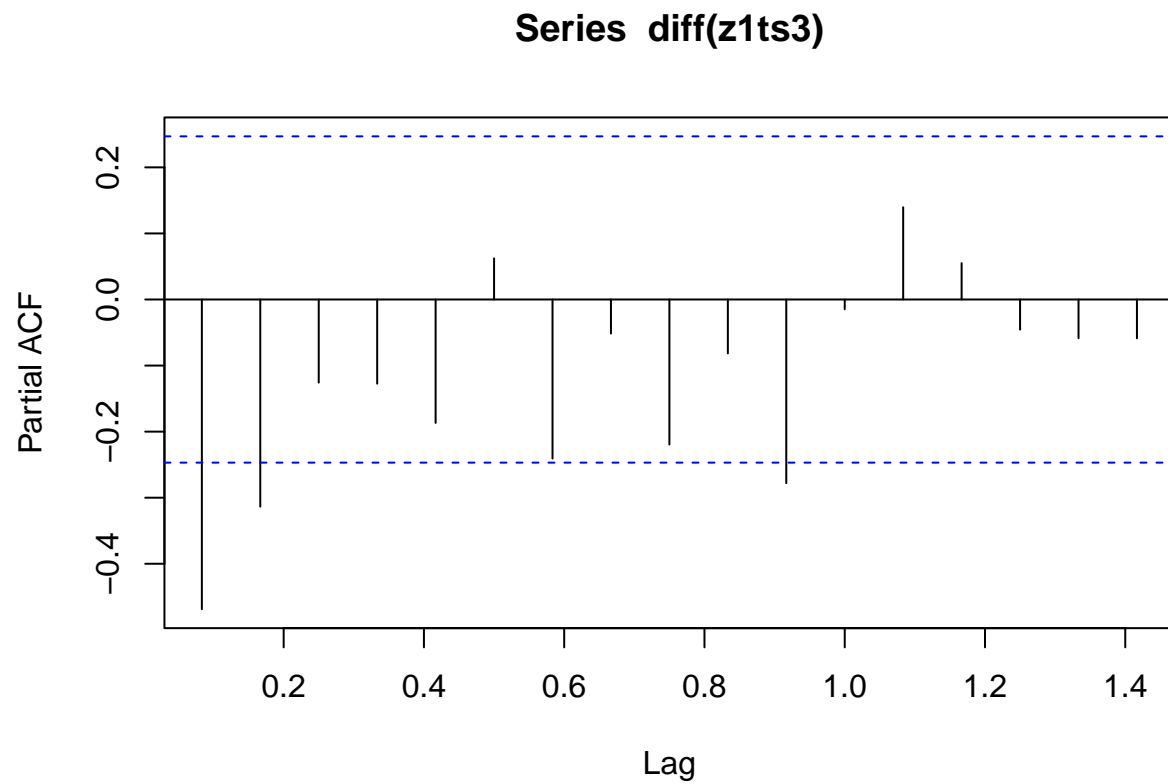
Szacujemy nasze parametry p i q

```
acf(diff(z1ts3)) # 0 lub 1
```

Series diff(z1ts3)



```
pacf(diff(z1ts3)) # 0 lub 1
```

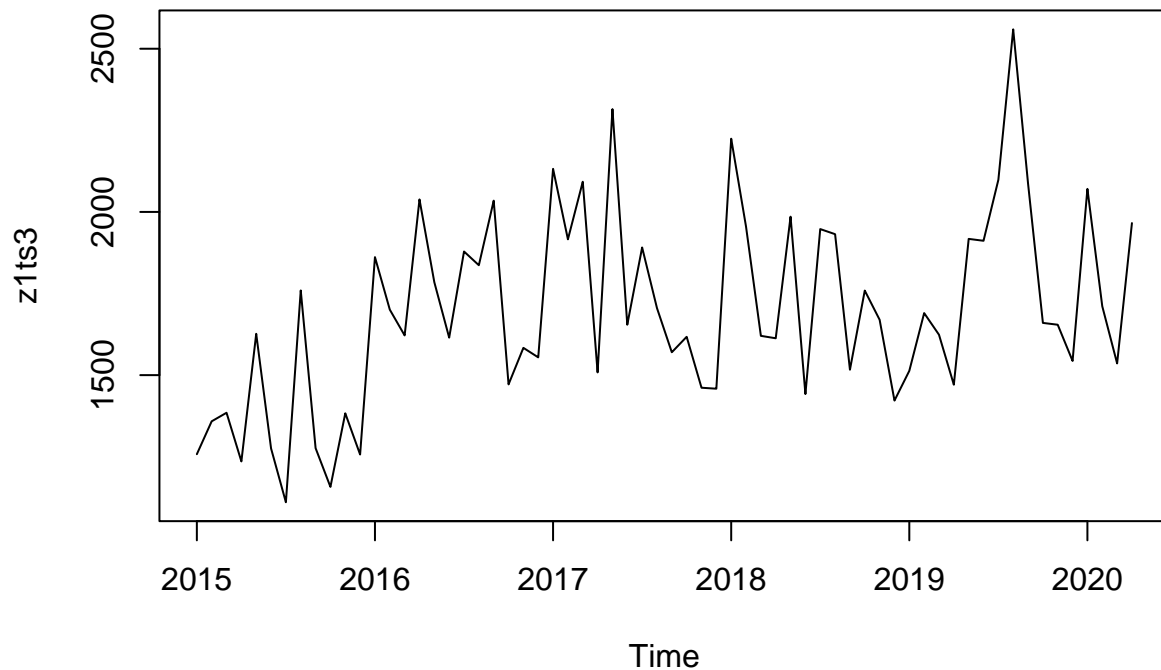


Tworzymy nasze modele i ja porównujemy

```
print("Wersja 1")
```

```
## [1] "Wersja 1"
```

```
plot(z1ts3)
```



```
Arima(y=z1ts3, order = c(0,1,1),lambda = NULL)
```

```
## Series: z1ts3
## ARIMA(0,1,1)
##
## Coefficients:
##             ma1
##      -0.777576967
## s.e.   0.094864280
##
## sigma^2 = 77531.403: log likelihood = -443.99
## AIC=891.99  AICc=892.19  BIC=896.27
```

```
print("Wersja 2")
```

```
## [1] "Wersja 2"
```

```
Arima(y=z1ts3,lambda = NULL,seasonal = c(0,1,0))
```

```
## Series: z1ts3
## ARIMA(0,0,0)(0,1,0)[12]
##
## sigma^2 = 126463.647: log likelihood = -379.23
## AIC=760.45  AICc=760.53  BIC=762.4
```

```
print("Wersja 3")
```

```
## [1] "Wersja 3"
```

```
Arima(y=z1ts3, order = c(0,1,1),lambda = 'auto')
```

```
## Series: z1ts3
## ARIMA(0,1,1)
## Box Cox transformation: lambda= 1.13401850576
##
## Coefficients:
##             ma1
##      -0.779932384
## s.e.   0.095288471
##
## sigma^2 = 575728.656: log likelihood = -507.15
## AIC=1018.31  AICc=1018.51  BIC=1022.6
```

```
print("Wersja 4")
```

```
## [1] "Wersja 4"
```

```
auto.arima(z1ts3)
```

```
## Series: z1ts3
## ARIMA(0,1,1)(1,0,0)[12]
##
## Coefficients:
##             ma1             sar1
##      -0.768837992  0.328685735
## s.e.   0.087465007  0.140030282
##
## sigma^2 = 71332.2245: log likelihood = -441.51
## AIC=889.02  AICc=889.43  BIC=895.45
```

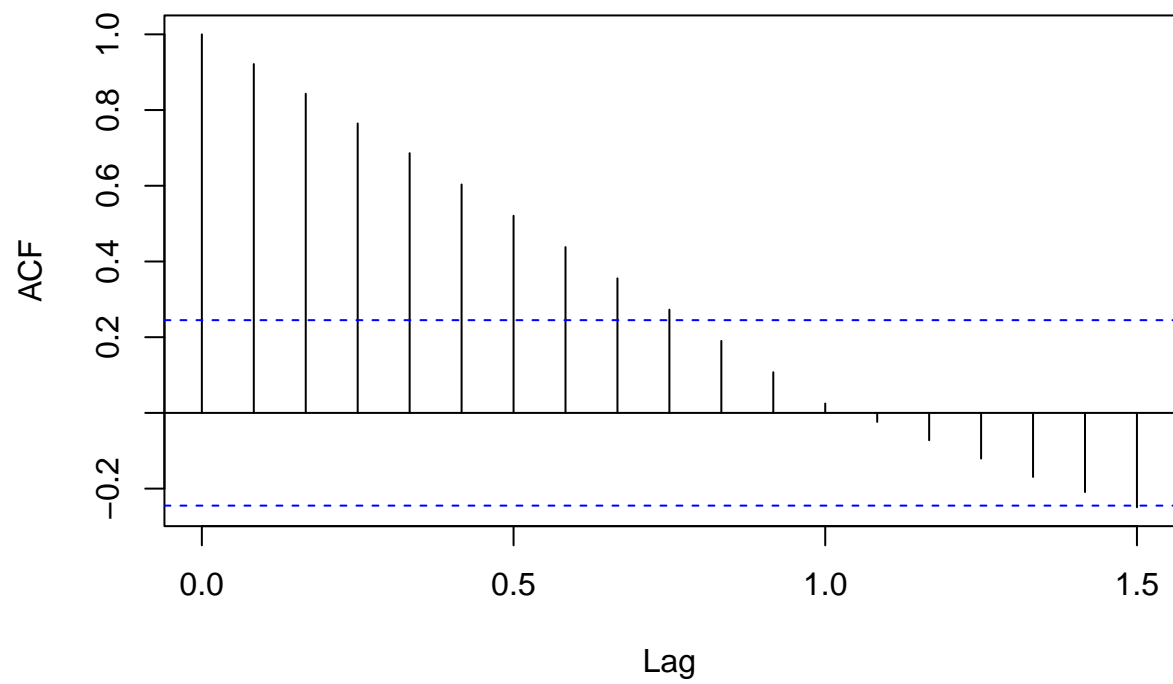
```
z1ts3_best_model<-auto.arima(z1ts3)
```

Zbiór 1 seria 4

Szacujemy nasze parametry p i q

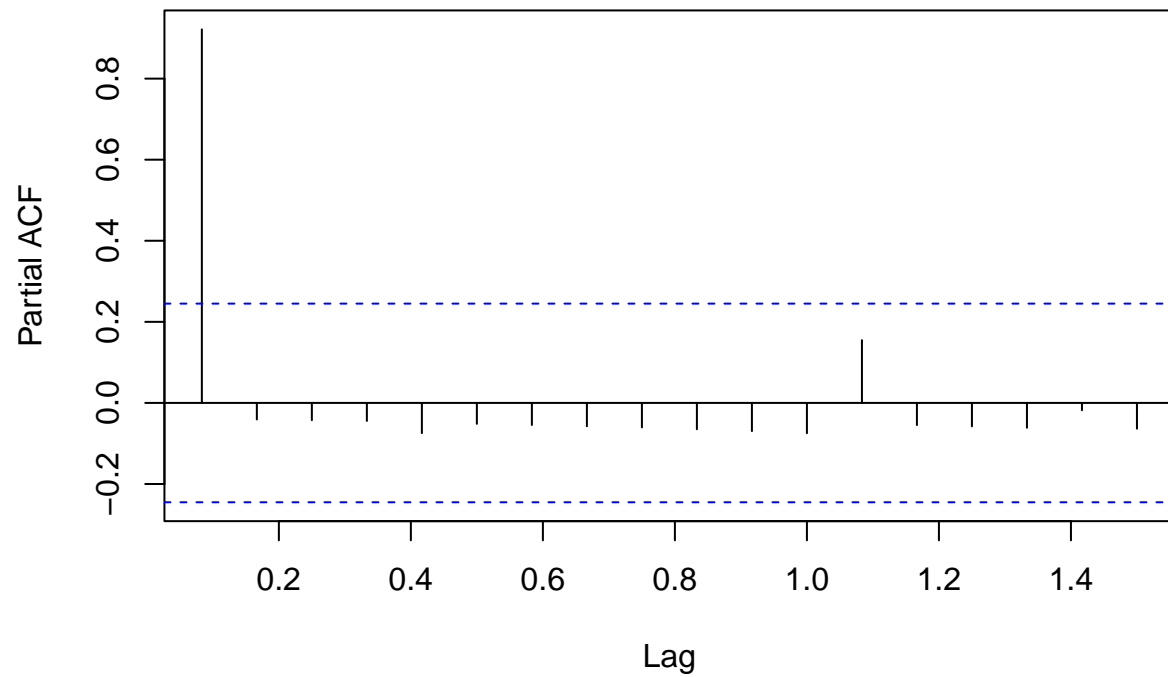
```
acf(z1ts4) # 0 lub 2
```

Series z1ts4



```
pacf(z1ts4) # 1
```

Series z1ts4

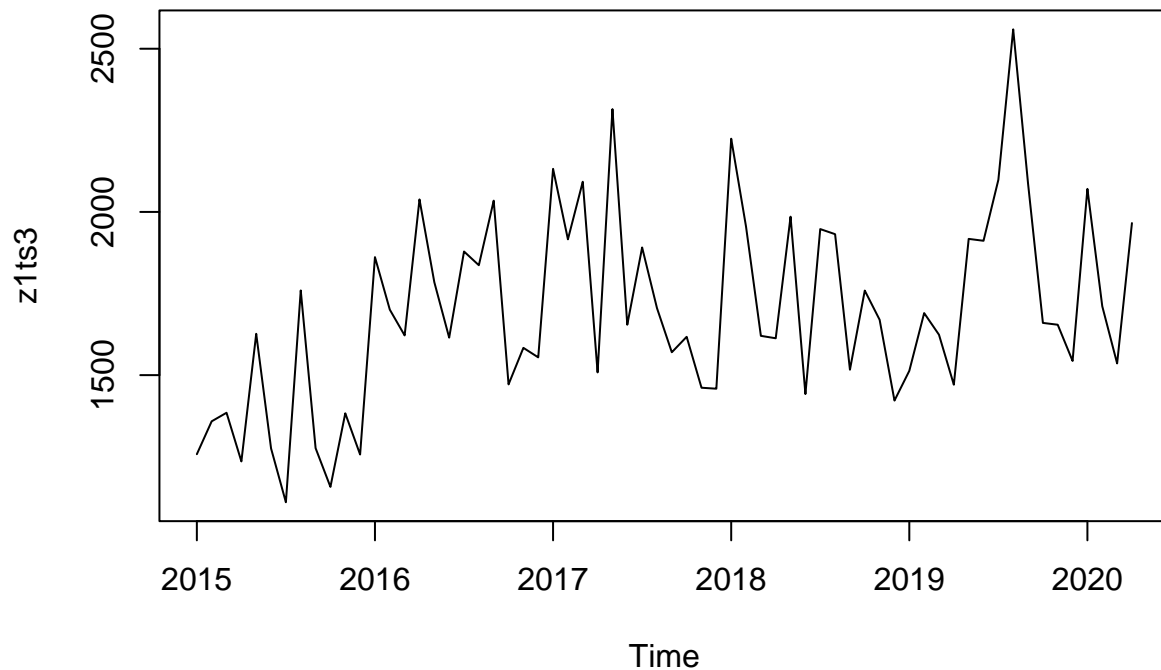


Tworzymy nasze modele i ja porównujemy

```
print("Wersja 1")
```

```
## [1] "Wersja 1"
```

```
plot(z1ts3)
```

```
Arima(y=z1ts4, order = c(0,0,2),lambda = NULL)
```

```
## Series: z1ts4
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1          ma2          mean
##      1.037827722  0.578004619 28709430.041092
## s.e.  0.109150404  0.084209359   169362.874685
##
## sigma^2 = 288855317994: log likelihood = -934.42
## AIC=1876.84   AICc=1877.52   BIC=1885.48
```

```
print("Wersja 2")
```

```
## [1] "Wersja 2"
```

```
Arima(y=z1ts4,lambda = NULL,seasonal = c(0,0,2))
```

```
## Series: z1ts4
## ARIMA(0,0,0)(0,0,2)[12] with non-zero mean
##
## Coefficients:
##          sma1          sma2          mean
```

```
##      -0.359978291  -0.639946112  28608190.9401371
## s.e.   0.193425311   0.163558626    71193.8253465
##
## sigma^2 = 590318753892:  log likelihood = -965.57
## AIC=1939.13   AICc=1939.81   BIC=1947.77
```

```
print("Wersja 3")
```

```
## [1] "Wersja 3"
```

```
auto.arima(z1ts4)
```

```
## Series: z1ts4
## ARIMA(0,1,0)
##
## sigma^2 = 145201015558:  log likelihood = -898.98
## AIC=1799.97   AICc=1800.03   BIC=1802.11
```

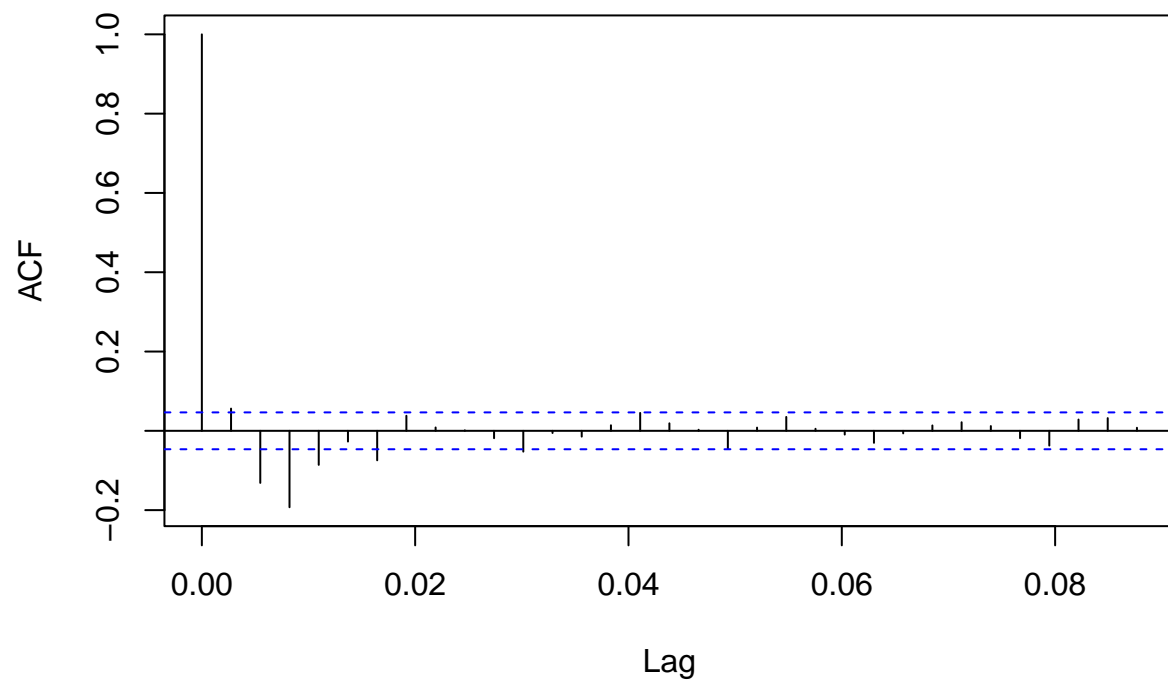
```
z1ts4_best_model<-auto.arima(z1ts4,stationary = TRUE,seasonal = FALSE)
z1ts4_best_model2<-Arima(y=z1ts4,lambda = NULL,seasonal = c(0,0,2))
```

Zbiór 2

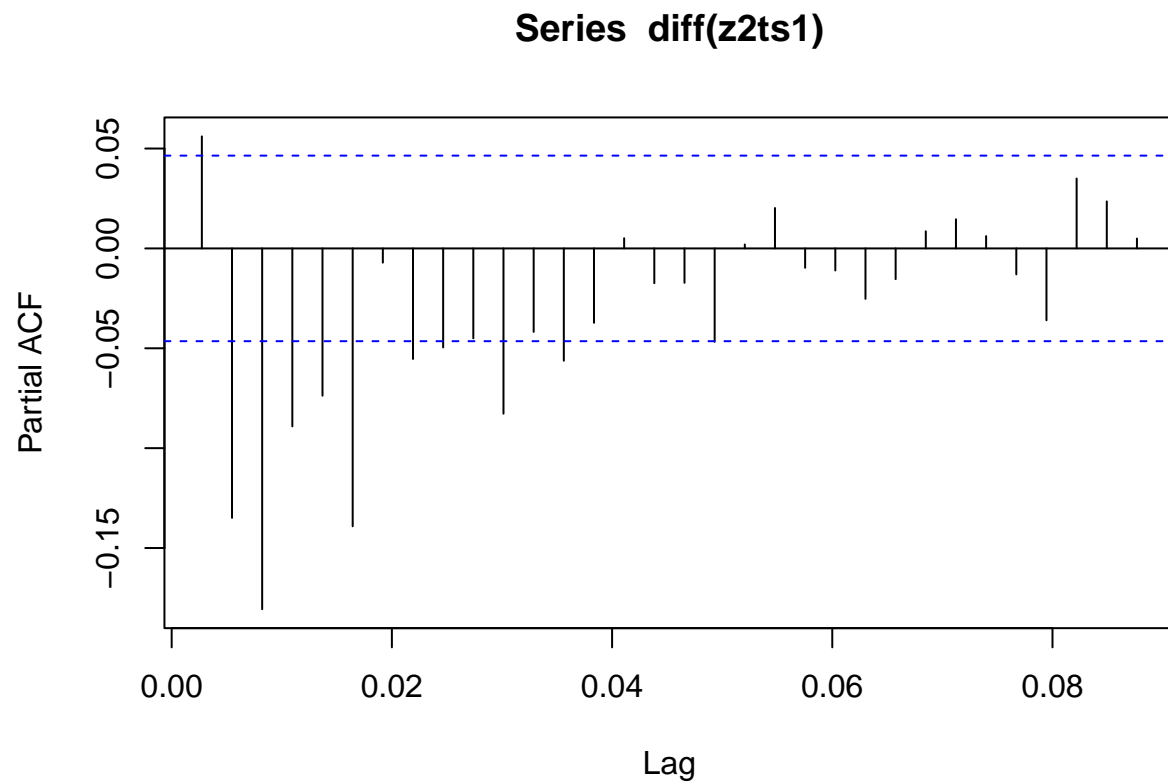
Szacujemy nasze parametry p i q

```
acf(diff(z2ts1)) # 0
```

Series diff(z2ts1)



```
pacf(diff(z2ts1)) # 0
```

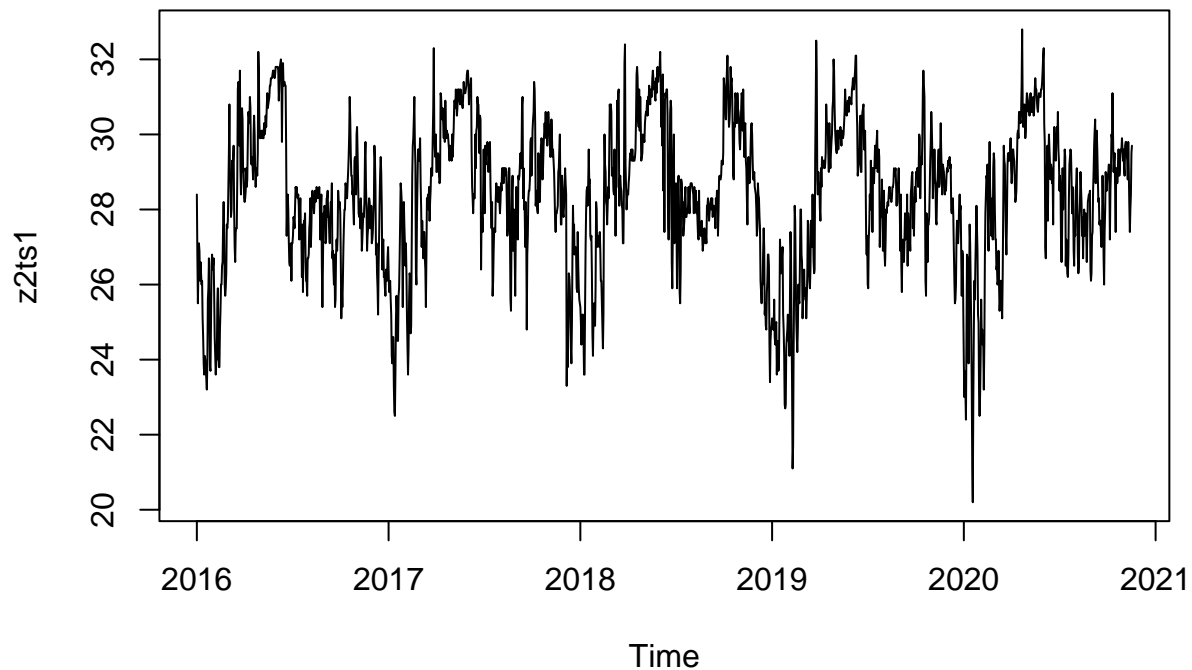


Tworzymy nasze modele i ja porównujemy

```
print("Wersja 1")
```

```
## [1] "Wersja 1"
```

```
plot(z2ts1)
```



```
Arima(y=z2ts1, order = c(6,1,6),lambda = NULL)
```

```
## Series: z2ts1
## ARIMA(6,1,6)
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ar5
##    0.120485816 -0.231143748 -0.406337647 -0.266964283  0.440145114
## s.e. 0.109156973  0.099838253  0.105955338  0.108388462  0.116964692
##          ar6          ma1          ma2          ma3          ma4
##   -0.127762817 -0.132583294  0.048054045  0.193017772  0.141828720
## s.e.  0.088920318  0.109173219  0.101080552  0.099282060  0.108474112
##          ma5          ma6
##   -0.598010747 -0.121423821
## s.e.  0.119260338  0.100128094
##
## sigma^2 = 0.636741064: log likelihood = -2118.25
## AIC=4262.5   AICc=4262.7   BIC=4333.79
```

```
print("Wersja 2")
```

```
## [1] "Wersja 2"
```

```
Arima(y=z2ts1,lambda = NULL,seasonal = c(0,0,0))
```

```
## Series: z2ts1
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##             mean
##      28.342560359
## s.e.   0.046431010
##
## sigma^2 = 3.84170445: log likelihood = -3725.17
## AIC=7454.34   AICc=7454.34   BIC=7465.31
```

```
print("Wersja 3")
```

```
## [1] "Wersja 3"
```

```
z2ts1_best_model<-auto.arima(z2ts1)
z2ts1_best_model
```

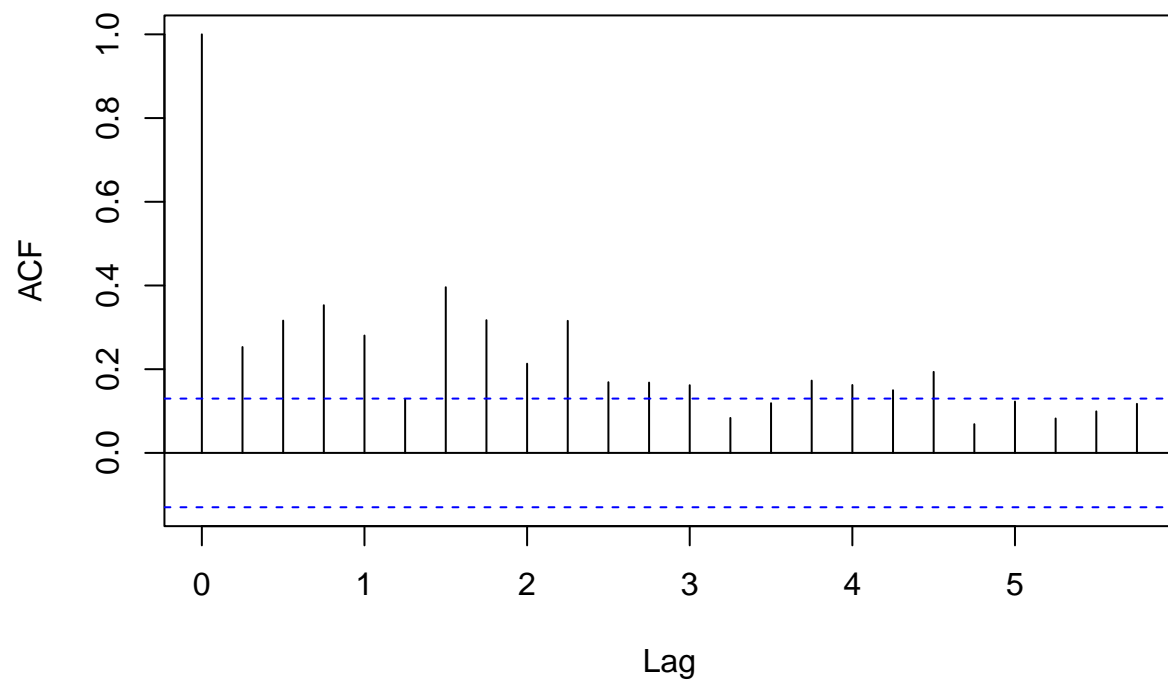
```
## Series: z2ts1
## ARIMA(2,0,0)(0,1,0)[365]
##
## Coefficients:
##             ar1             ar2
##      0.900594661  -0.170838920
## s.e.   0.026176808   0.026186483
##
## sigma^2 = 1.21436968: log likelihood = -2146.09
## AIC=4298.17   AICc=4298.19   BIC=4313.94
```

Zbiór 3

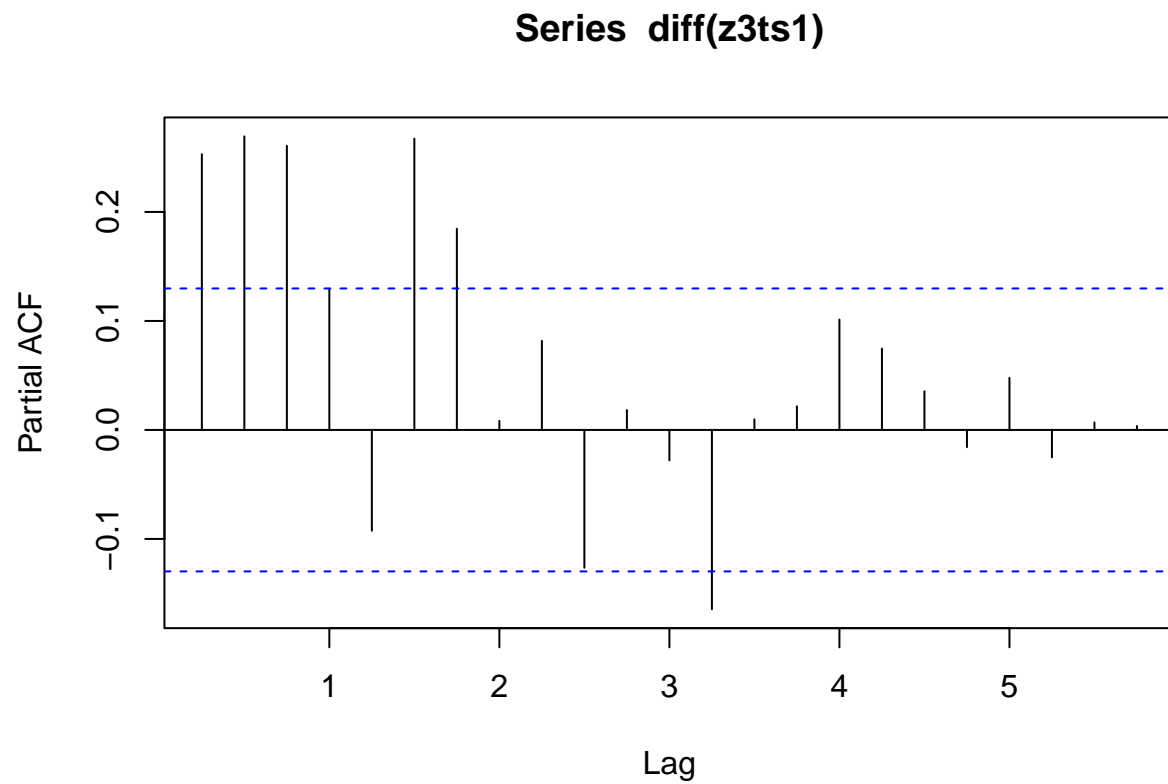
Szacujemy nasze parametry p i q

```
acf(diff(z3ts1)) # 1 lub 2
```

Series diff(z3ts1)



```
pacf(diff(z3ts1)) # 1 lub 2
```

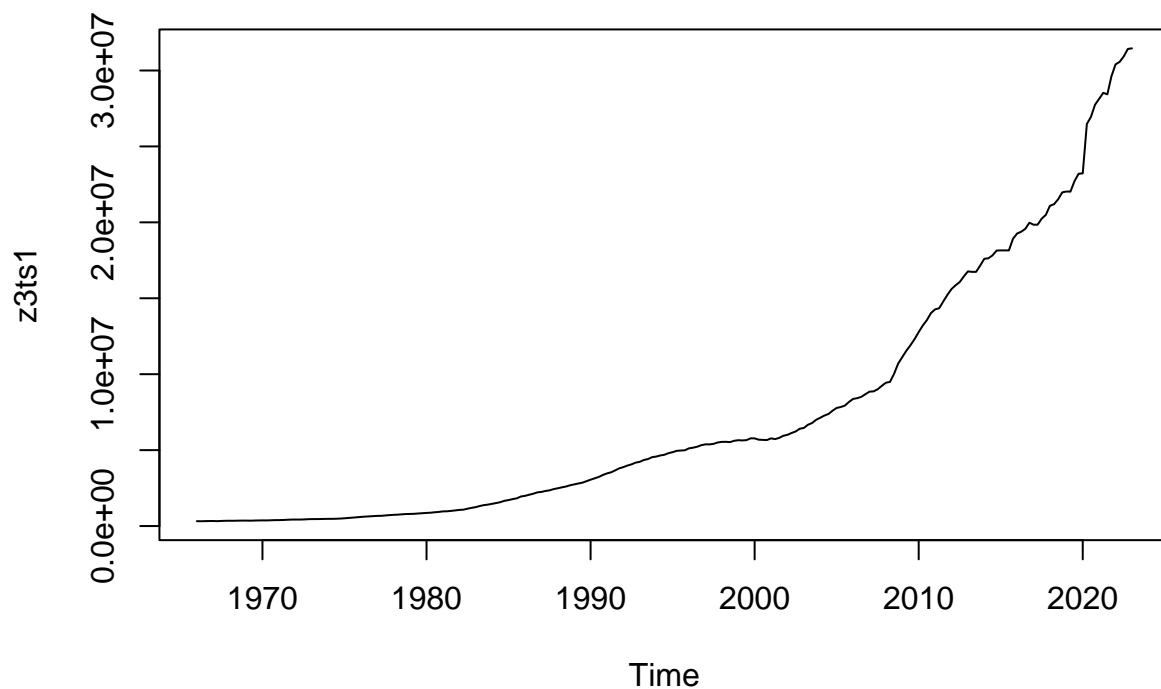


Tworzymy nasze modele i ja porównujemy

```
print("Wersja 1")
```

```
## [1] "Wersja 1"
```

```
plot(z3ts1)
```

```
Arima(y=z3ts1, order = c(4,3,4),lambda = NULL) #6244
```

```
## Series: z3ts1
## ARIMA(4,3,4)
##
## Coefficients:
##          ar1          ar2          ar3          ar4          ma1
##      -1.160428115  -0.780654263  0.232890134  0.254686881  -0.739468121
## s.e.   0.073899460   0.123203005  0.124464921  0.075065070   0.041925458
##          ma2          ma3          ma4
##      -0.450535148  -0.775545653  0.970194980
## s.e.   0.025517402   0.044235100  0.046661361
##
## sigma^2 = 51302594961: log likelihood = -3112.89
## AIC=6243.77  AICc=6244.6  BIC=6274.55
```

```
print("Wersja 2")
```

```
## [1] "Wersja 2"
```

```
Arima(y=z3ts1,lambda = NULL,seasonal = c(1,3,1))
```

```
## Series: z3ts1
## ARIMA(0,0,0)(1,3,1)[4]
```

```
##
## Coefficients:
##          sar1          sma1
##      -0.466047823  -0.999999500
## s.e.    0.059847941   0.026548251
##
## sigma^2 = 334599622700: log likelihood = -3196.1
## AIC=6398.21   AICc=6398.32   BIC=6408.35
```

```
print("Wersja 3")
```

```
## [1] "Wersja 3"
```

```
auto.arima(z3ts1) #6277
```

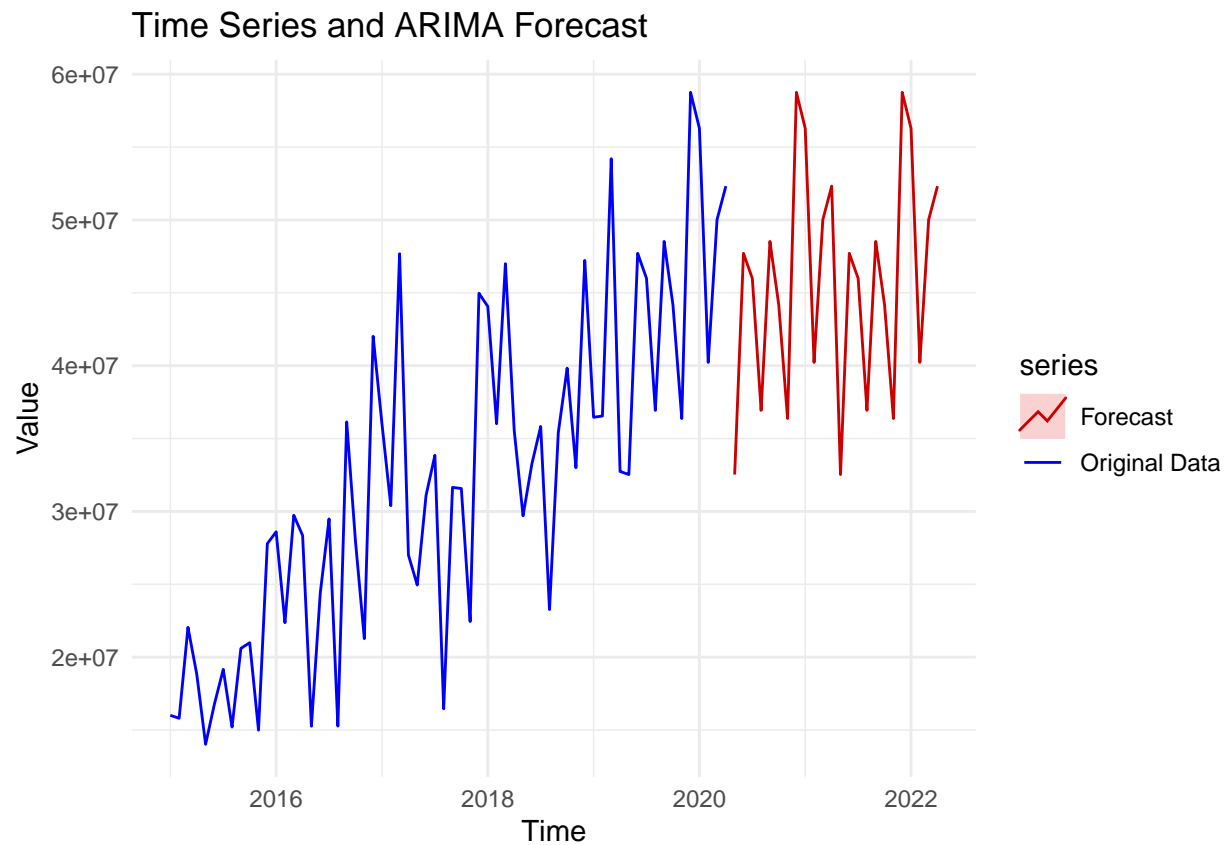
```
## Series: z3ts1
## ARIMA(0,2,1)
##
## Coefficients:
##          ma1
##      -0.905258256
## s.e.    0.031122056
##
## sigma^2 = 58756323130: log likelihood = -3136.88
## AIC=6277.75   AICc=6277.8   BIC=6284.6
```

```
z3ts1_best_model<-auto.arima(z3ts1)
```

Predykcja danych

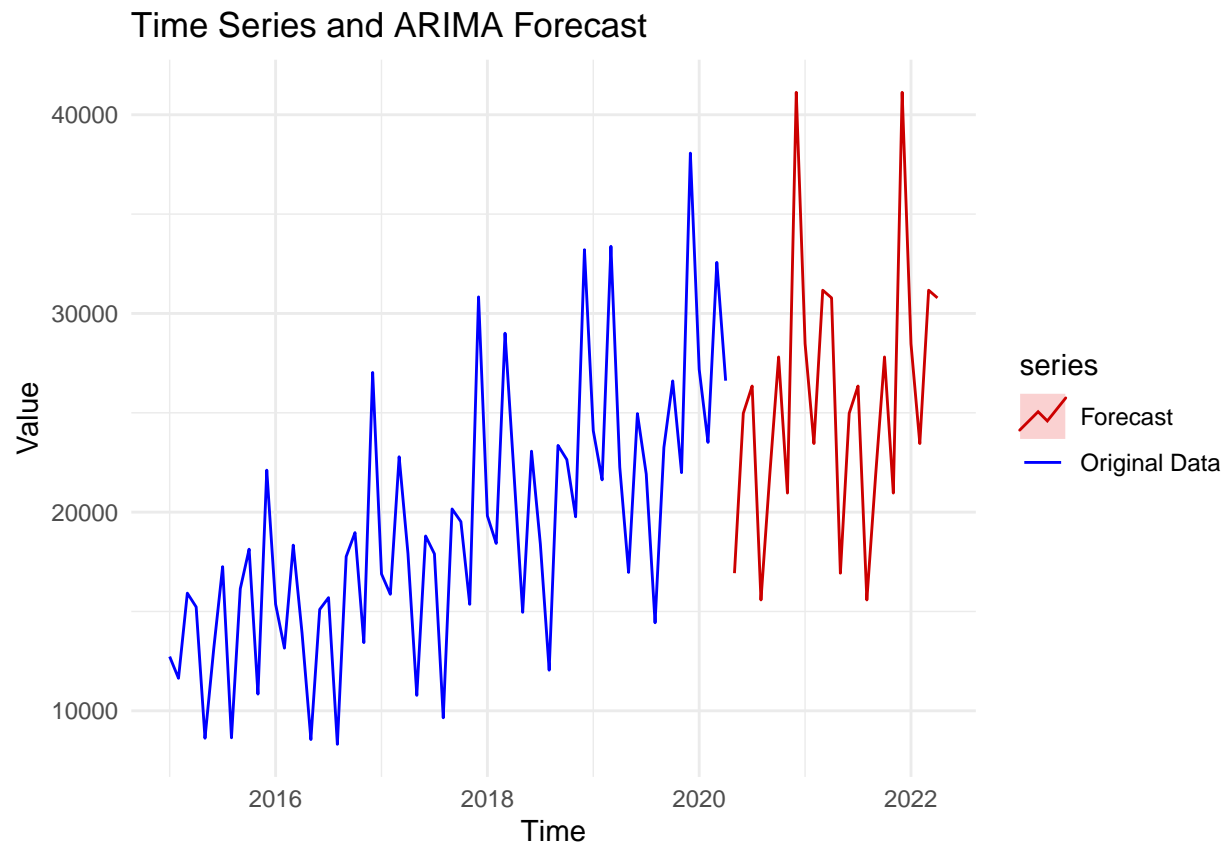
Teraz mając nasze modele możemy dokonać predykcji ## Zbiór 1 Dla naszego zbioru spróbujemy dokonać predykcji na następny rok ### Predykcja przychodów przedsiębiorstwa

```
forecast_valuesz11 <-forecast(z1tst1_best_model,h=24)
autoplot(z1ts1, series="Original Data") +
  autolayer(forecast_valuesz11, series="Forecast", PI=FALSE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="red"))
```



Predykcja ilości sprzedaży produktów

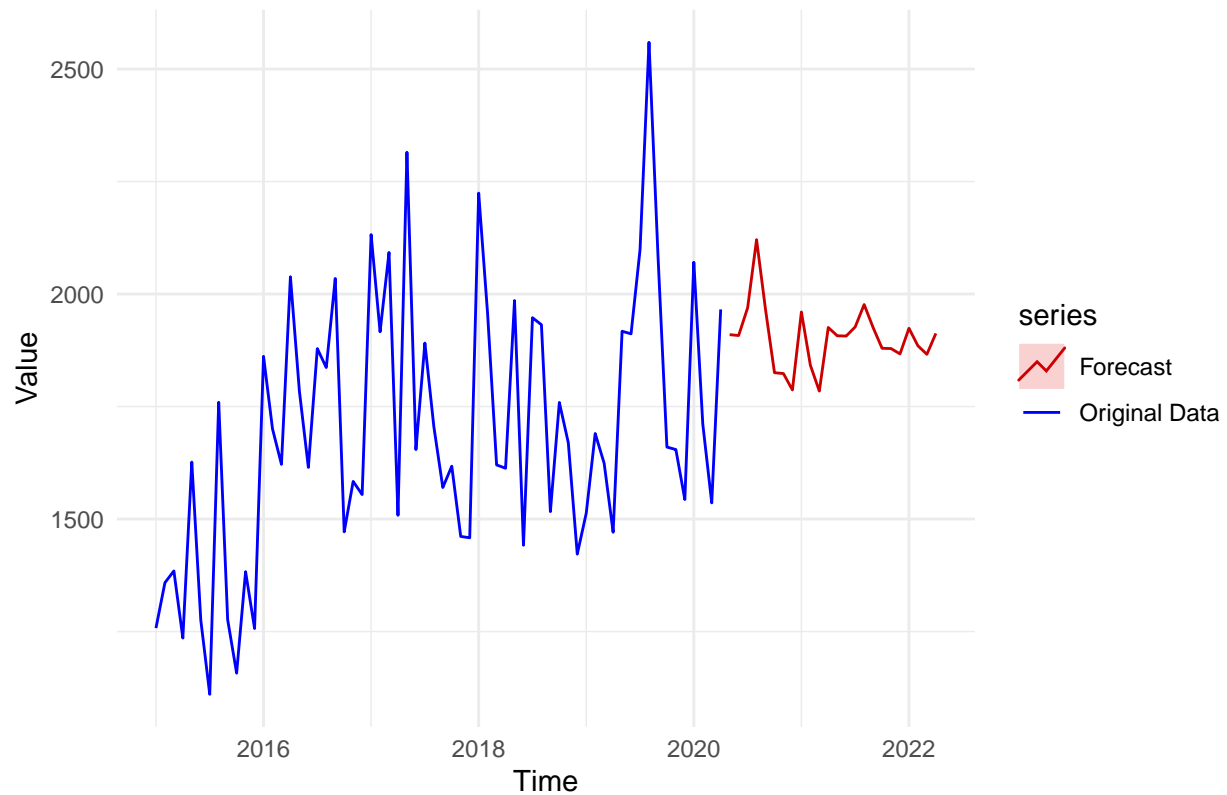
```
forecast_valuesz12 <-forecast(z1ts2_best_model,h=24)
autoplot(z1ts2, series="Original Data") +
  autolayer(forecast_valuesz12, series="Forecast", PI=FALSE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="red"))
```



Predykcja średniego kosztu produkcji

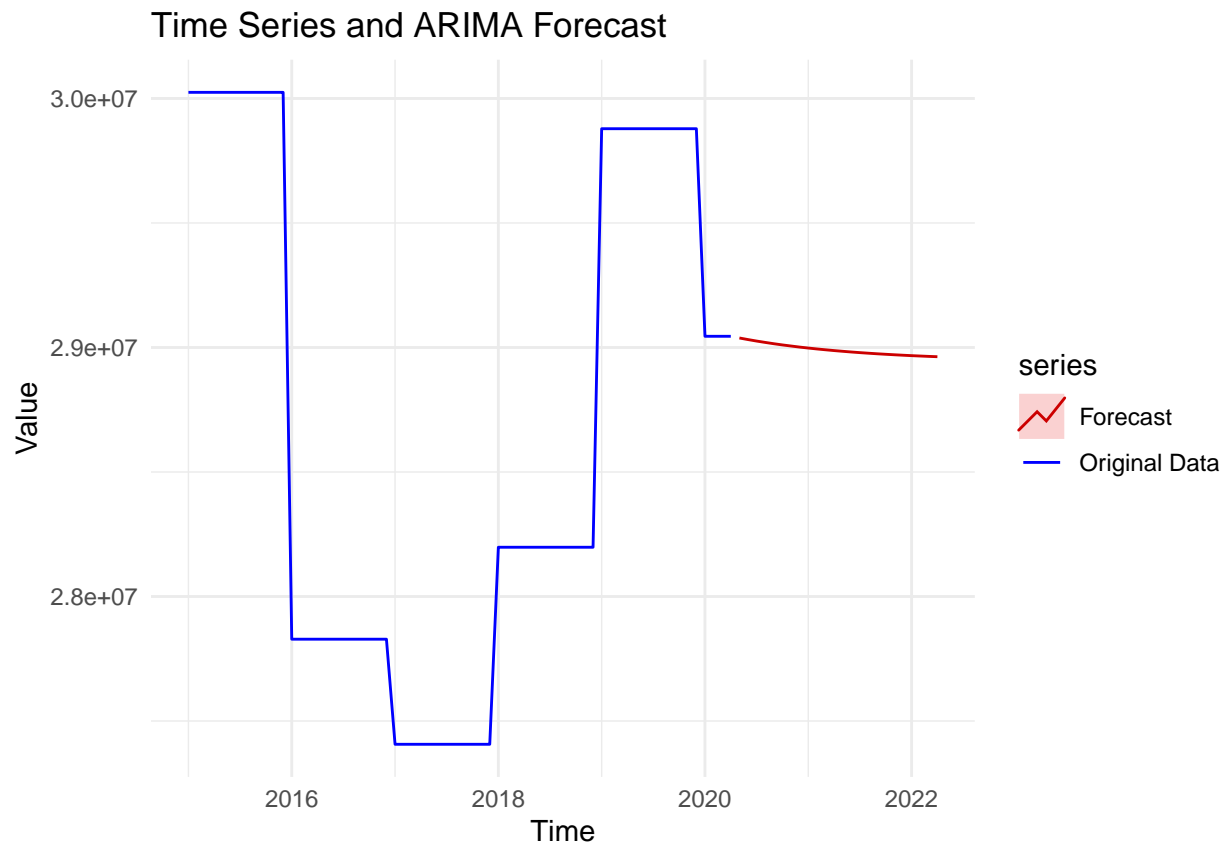
```
forecast_valuesz13 <-forecast(z1ts3_best_model,h=24)
autoplot(z1ts3, series="Original Data") +
  autolayer(forecast_valuesz13, series="Forecast", PI=FALSE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="red"))
```

Time Series and ARIMA Forecast



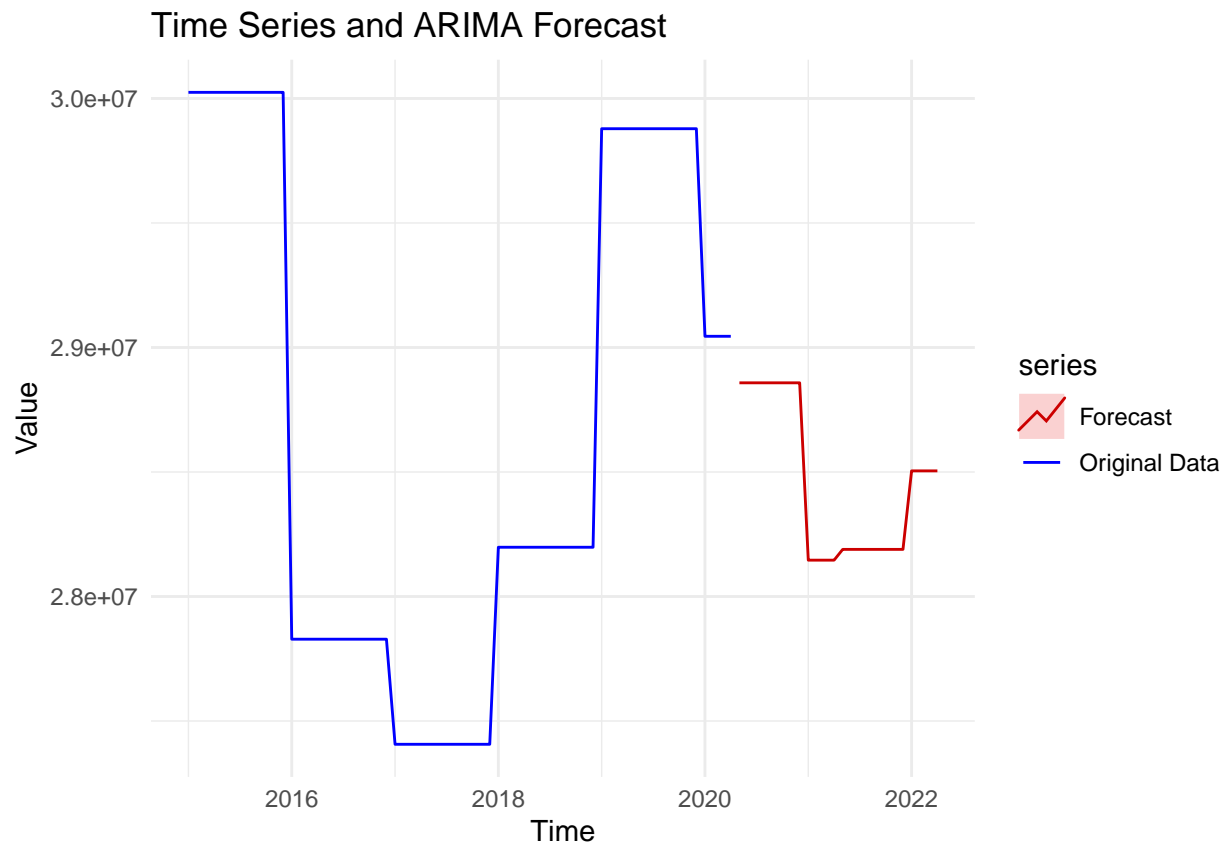
Predykcja rocznych płac regionów w wersji modelu autoarima

```
forecast_valuesz14 <-forecast(z1ts4_best_model,h=24)
autoplot(z1ts4, series="Original Data") +
  autolayer(forecast_valuesz14, series="Forecast", PI=FALSE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="red"))
```



Predykcja rocznych płac regionów w wersji modelu arima - nieznaczna różnica AICc

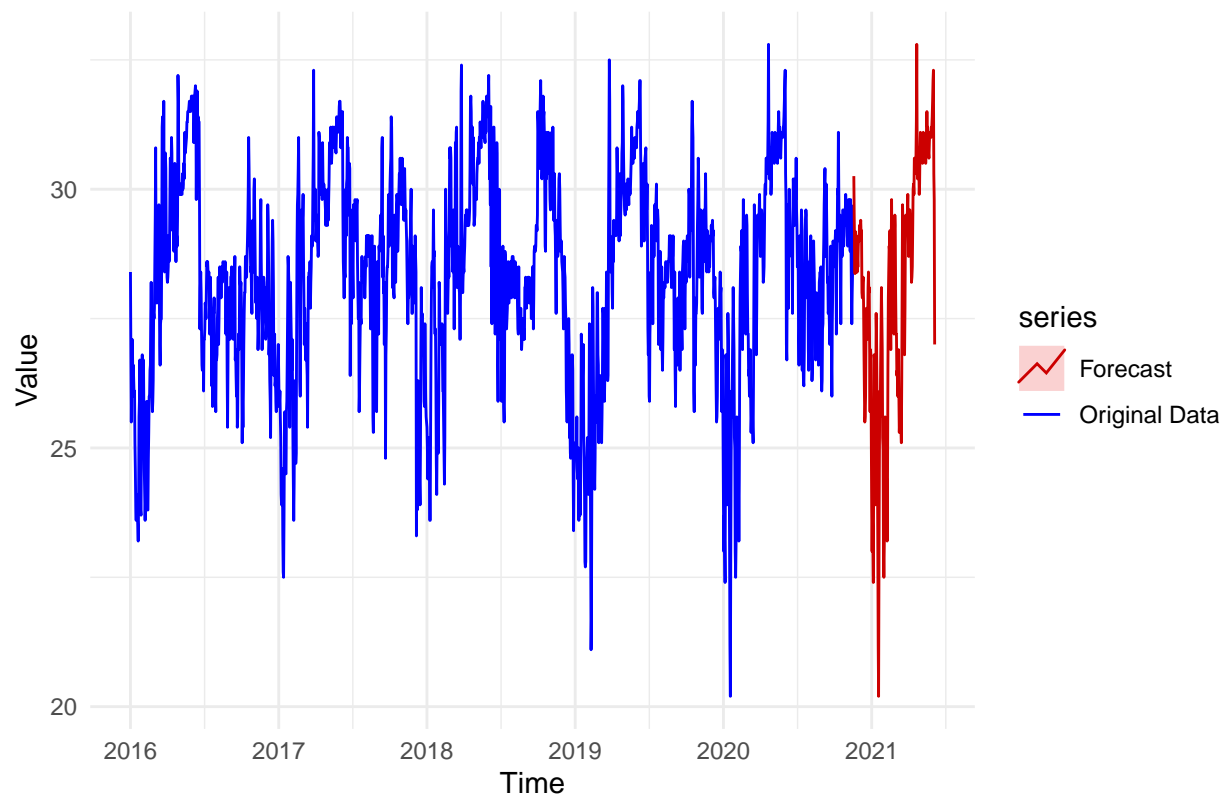
```
forecast_valuesz142 <-forecast(z1ts4_best_model2,h=24)
autoplot(z1ts4, series="Original Data") +
  autolayer(forecast_valuesz142, series="Forecast", PI=FALSE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="red"))
```



Zbiór 2 ### Predykcja dziennych temperatur

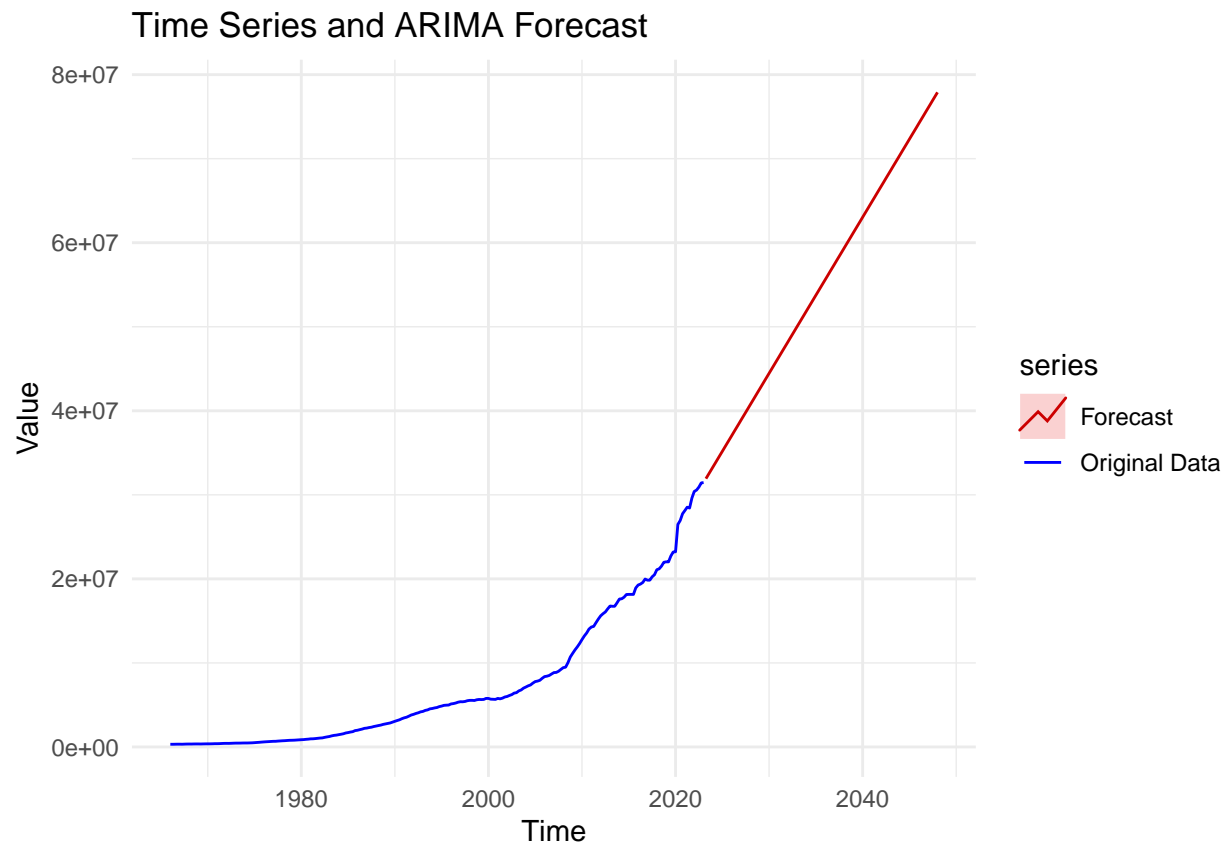
```
forecast_valuesz21 <-forecast(z2ts1_best_model,h=200)
autoplot(z2ts1, series="Original Data") +
  autolayer(forecast_valuesz21, series="Forecast", PI=FALSE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="red"))
```

Time Series and ARIMA Forecast



Zbiór 3 ### Predykcja kwartalna długu

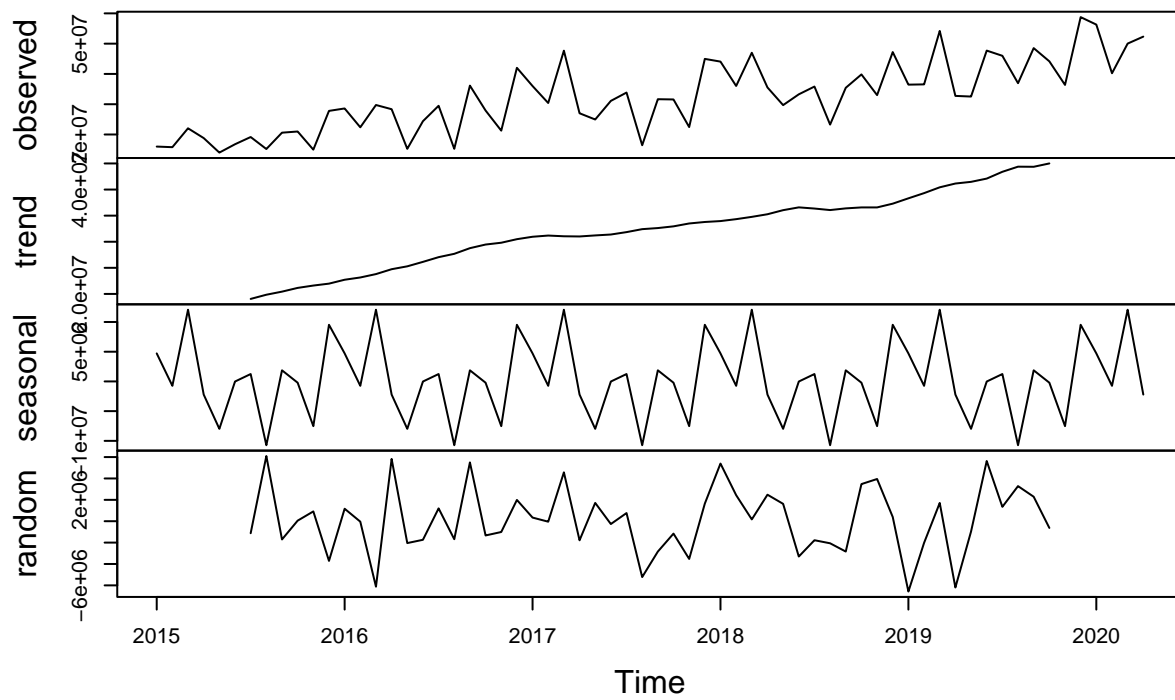
```
forecast_valuesz31 <-forecast(z3ts1_best_model,h=100)
autoplot(z3ts1, series="Original Data") +
  autolayer(forecast_valuesz31, series="Forecast", PI=FALSE) +
  ggtitle("Time Series and ARIMA Forecast") +
  xlab("Time") +
  ylab("Value") +
  theme_minimal() +
  scale_colour_manual(values=c("Original Data"="blue", "Forecast"="red"))
```

Dekompozycje szeregu PODODAWAĆ OPISY ## Szereg z1ts1

```
decomposedz11 <- decompose(z1ts1)
plot(decomposedz11)
```

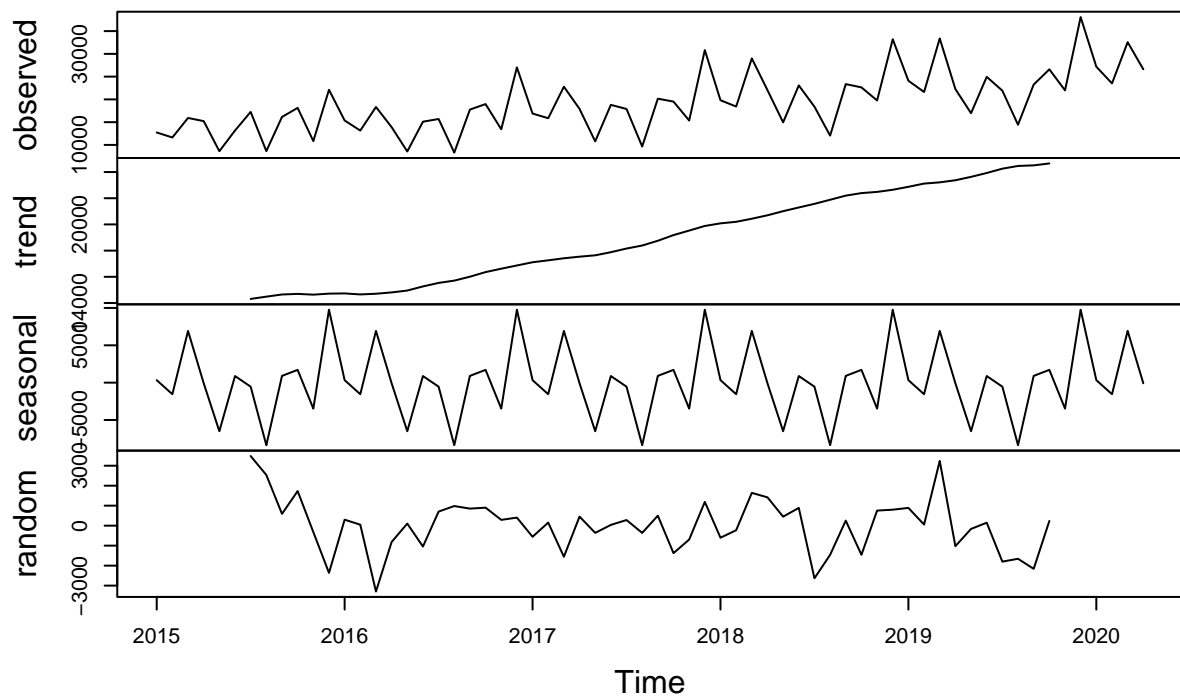
Decomposition of additive time series



Szereg z1ts2

```
decomposedz12 <- decompose(z1ts2)
plot(decomposedz12)
```

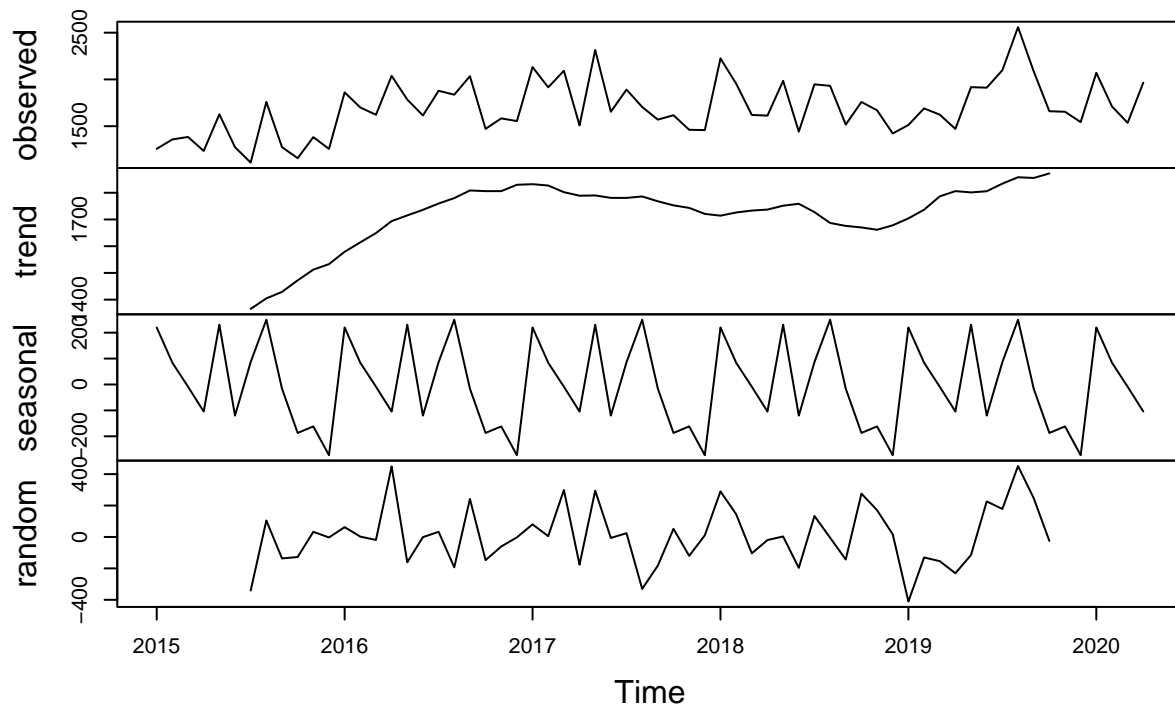
Decomposition of additive time series



```
# Szereg z1ts3
```

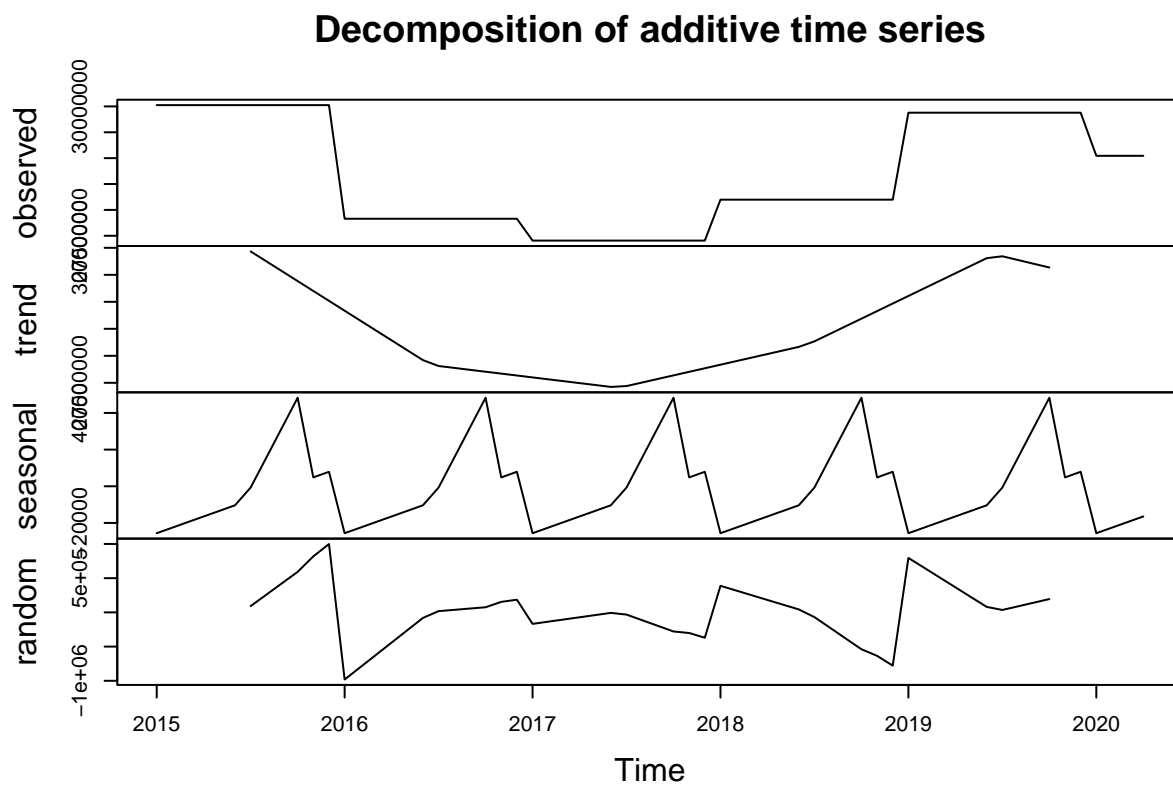
```
decomposedz13 <- decompose(z1ts3)  
plot(decomposedz13)
```

Decomposition of additive time series



```
# Szereg z1ts4
```

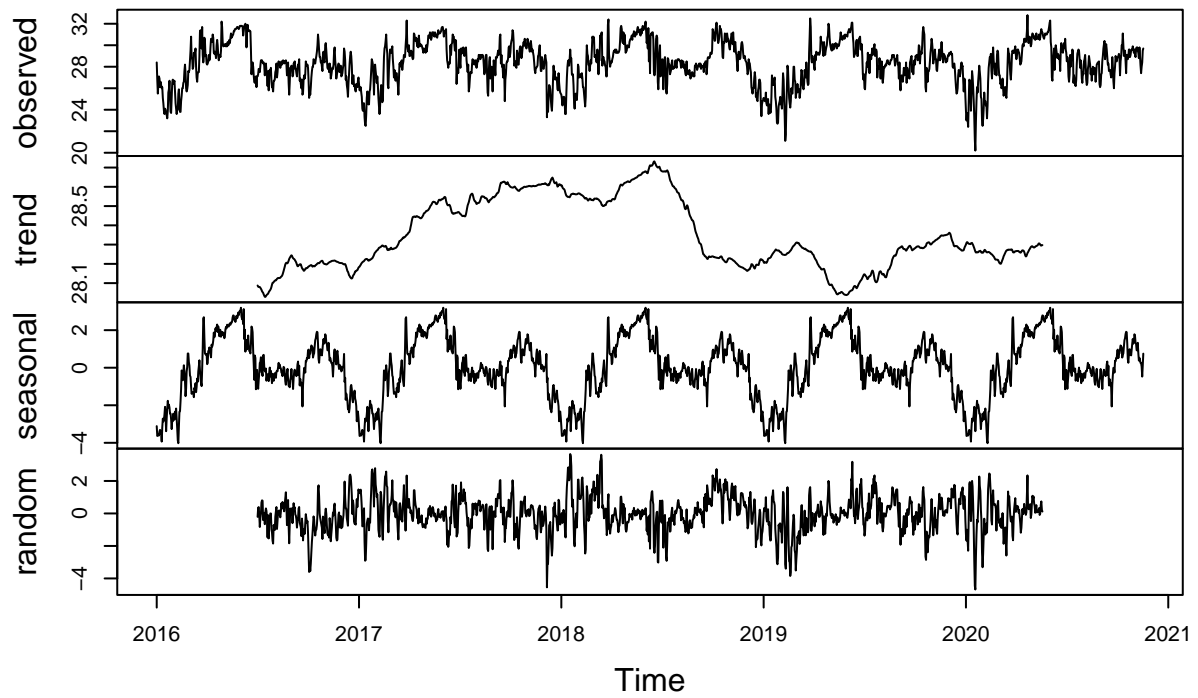
```
decomposedz14 <- decompose(z1ts4)  
plot(decomposedz14)
```



```
# Szereg z2ts1
```

```
decomposedz21 <- decompose(z2ts1)
plot(decomposedz21)
```

Decomposition of additive time series



```
# Szereg z3ts1
```

```
decomposedz31 <- decompose(z3ts1)  
plot(decomposedz31)
```

Decomposition of additive time series

