Analiza danych - Projekt v.0.1

Kamil Kukiełka, Michał Zakielarz, Klaudia Kopeć

2024-04-29

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(dplyr)
```

```
##
## Dołączanie pakietu: 'dplyr'
```

```
## Następujące obiekty zostały zakryte z 'package:stats':
##
##     filter, lag
```

```
## Następujące obiekty zostały zakryte z 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(urca)
library(ggplot2)
library(lmtest)
```

```
## Ładowanie wymaganego pakietu: zoo
```

```
##
## Dołączanie pakietu: 'zoo'
```

```
## Następujące obiekty zostały zakryte z 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(skedastic)
```

Zaczytanie danych i ich przygotowanie

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

Zestaw 1

Nasz pierwszy zestaw dotyczy różnych wskaźników pewnego przedsiębiorstwa, które zmienieniają 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. ## Zbiór 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
```

Zbiór 2

Zawiera ilość sprzedarzy 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
```

Zbiór 3

Zawiera średni koszt produkcji

```
##        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
```

Zbiór 4

Zawiera informację o średniej liczbie pracowników w regionie (rocznie)

```
##        Period Average_annual_payroll_of_regiion
## 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 śrendią dzienną temperaturę w Mumbaiu. Nasz zbiór zawiera więcej danych takich jak wilgoć, prędkość czy kierunek wiartu, 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
```

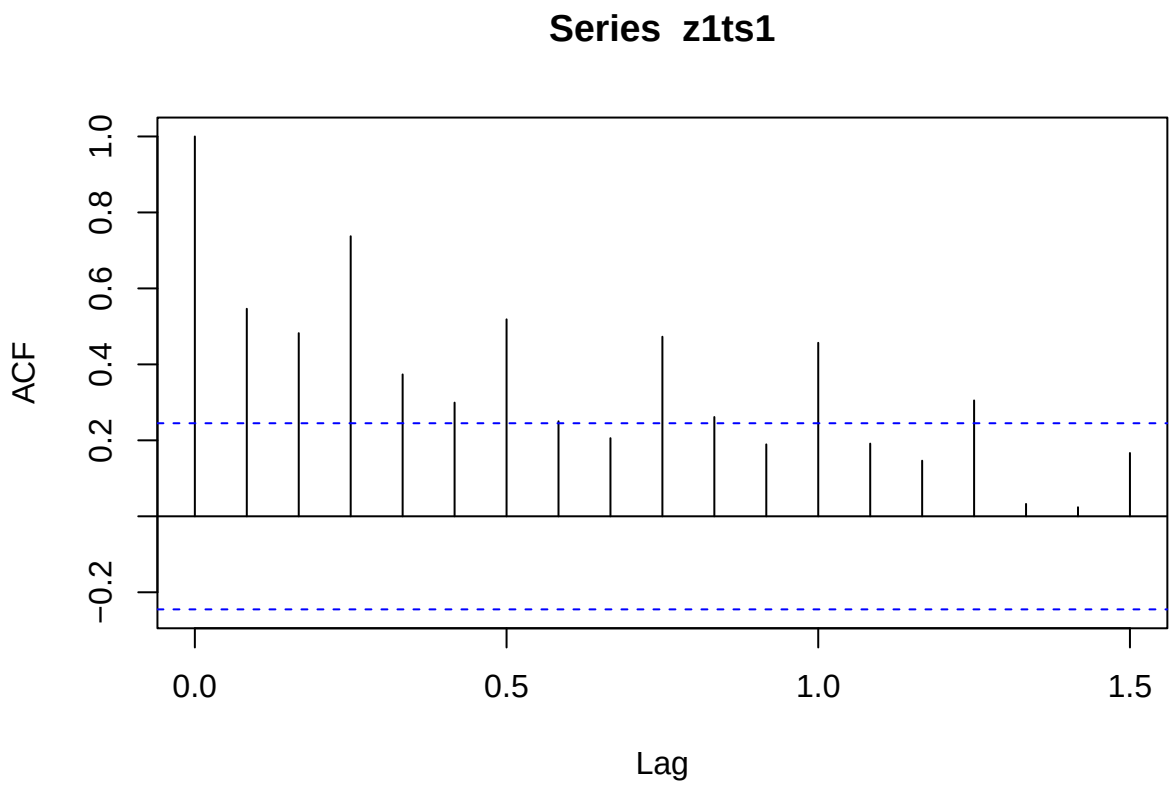
Zamiana na szereg czasowy

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_regiion,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)
```

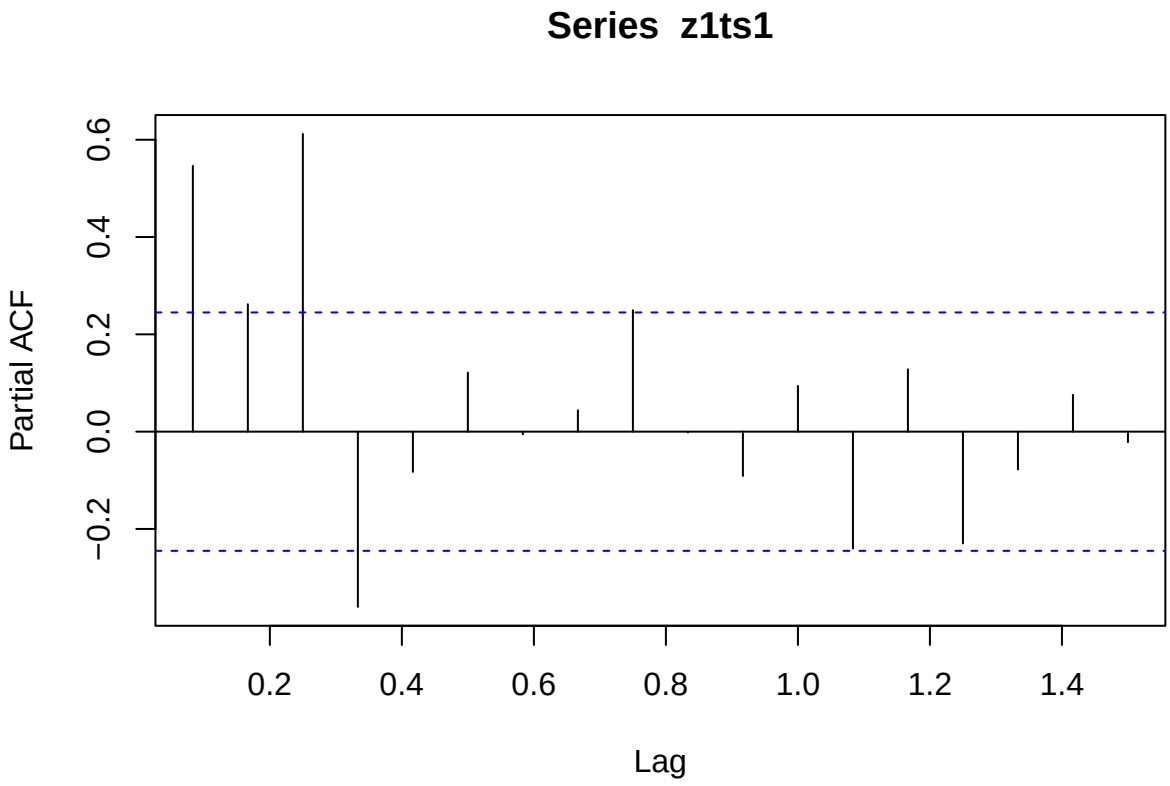
#Przeprowadzenie testów ## Autokorelacja ### Dla zbioru 1 Autokorelacja przychodów przedsiębiorstwa

```
acf(z1ts1)
```



Autokorelacja cząstkowa? przychodów przedsiębiorstwa

```r
pacf(z1ts1)
```



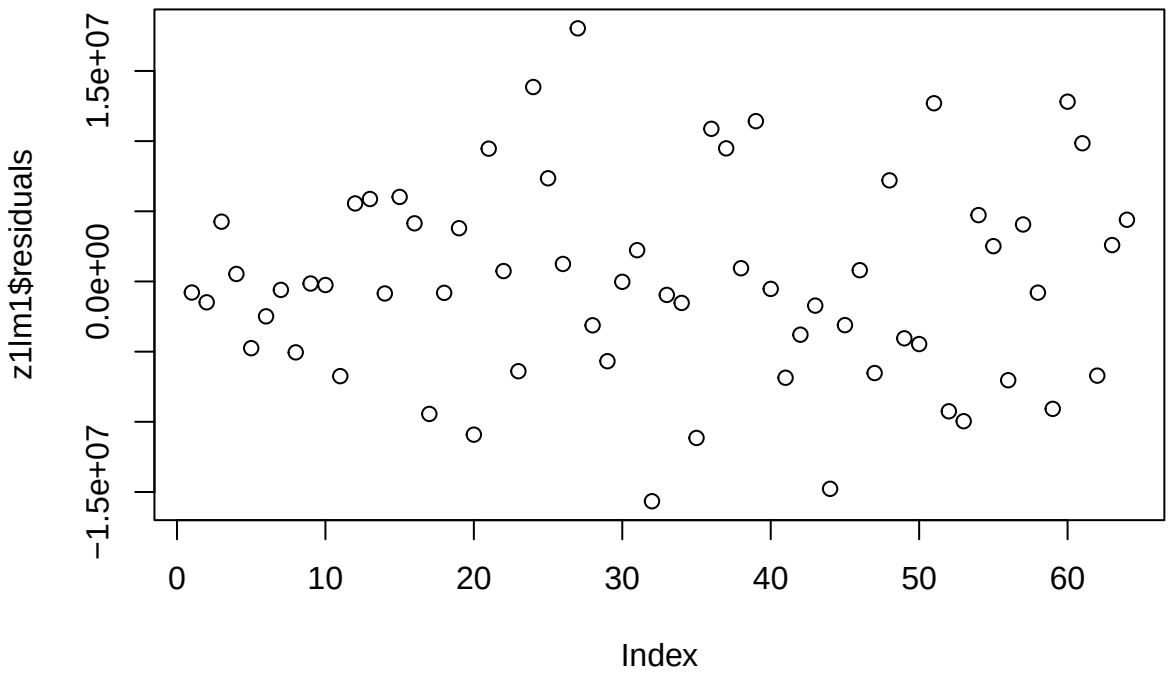
i tak dla pozostałych ## Dla zbioru 2

Test na heteroskedastyczność

Aby określić heteroskedastyczność szeregu należy najpier stowrzyć model liniowy z naszych szeregów czasowych, a następnie przeprowadzić test Breuscha-Pagana ## Dla zbioru 1

heteroskedastyczność przychodów przedsiębiorstwa

```r
df=data.frame(time=1:length(z1ts1),z1ts1)
z1lm1<-lm(z1ts1~time,data = df)
plot(z1lm1$residuals)
```




```
bptest(z1lm1)
```

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

Z tego wynika ....

#Test na stacjonarność szergu

Dla zbioru 1

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

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 3 lags.
##
```

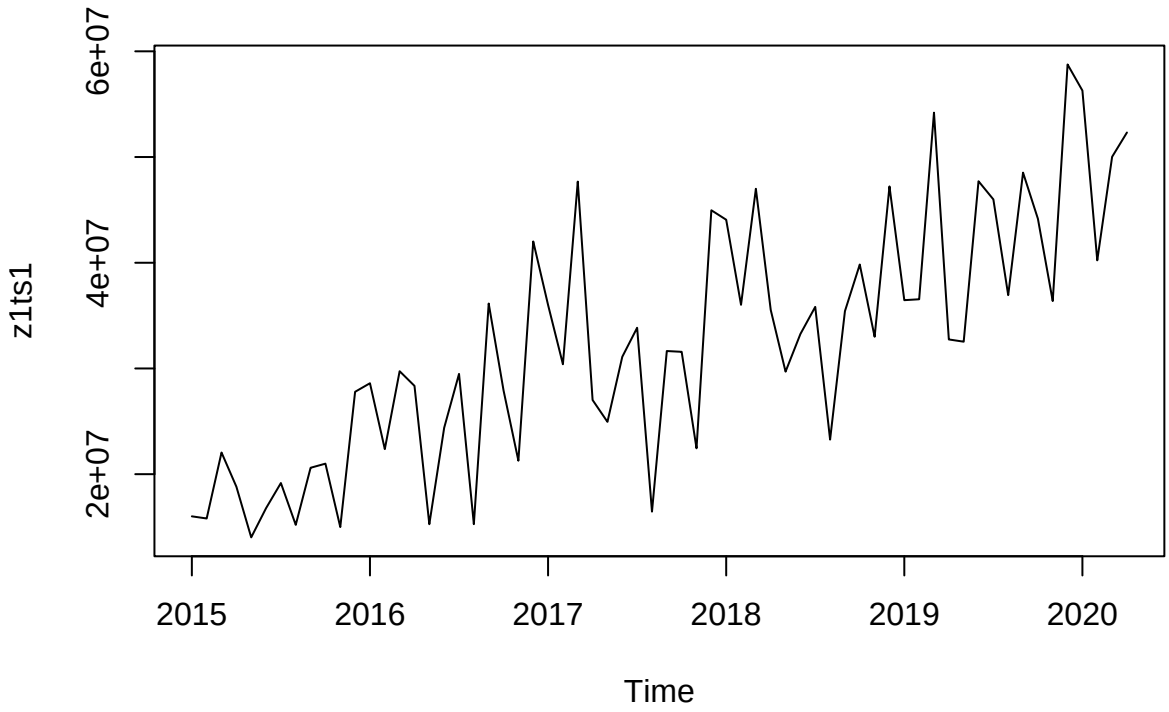
```
## Value of test-statistic is: 1.4597
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
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
```

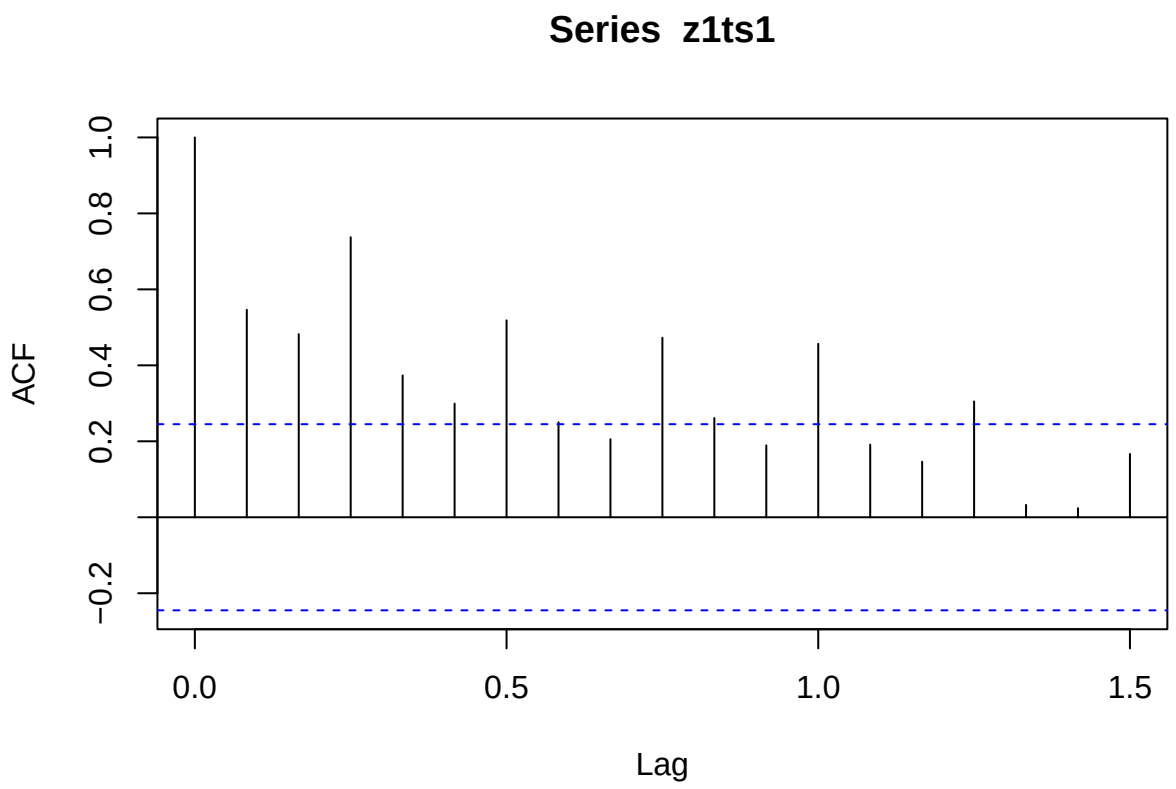
#Tworzenie modelu Bazując na naszych wcześniejszych danych musimy teraz dobrać odpowiednie parametry naszego modelu. Jako iż będziemy korzystać z modelu Arima potrzebujemy wartośći parametrów p,d,q.

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

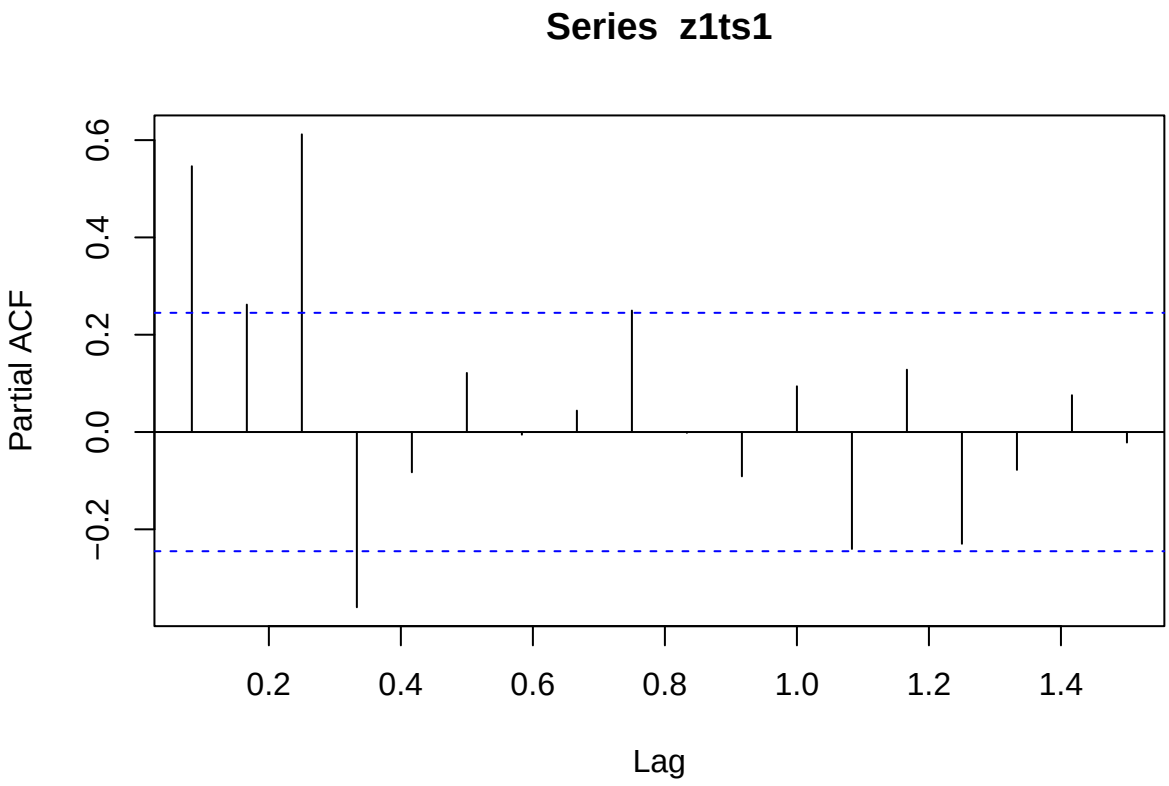


Aby wyznaczyć parametr p patrzymy na nasze korelogramy.

```
acf(z1ts1)
```

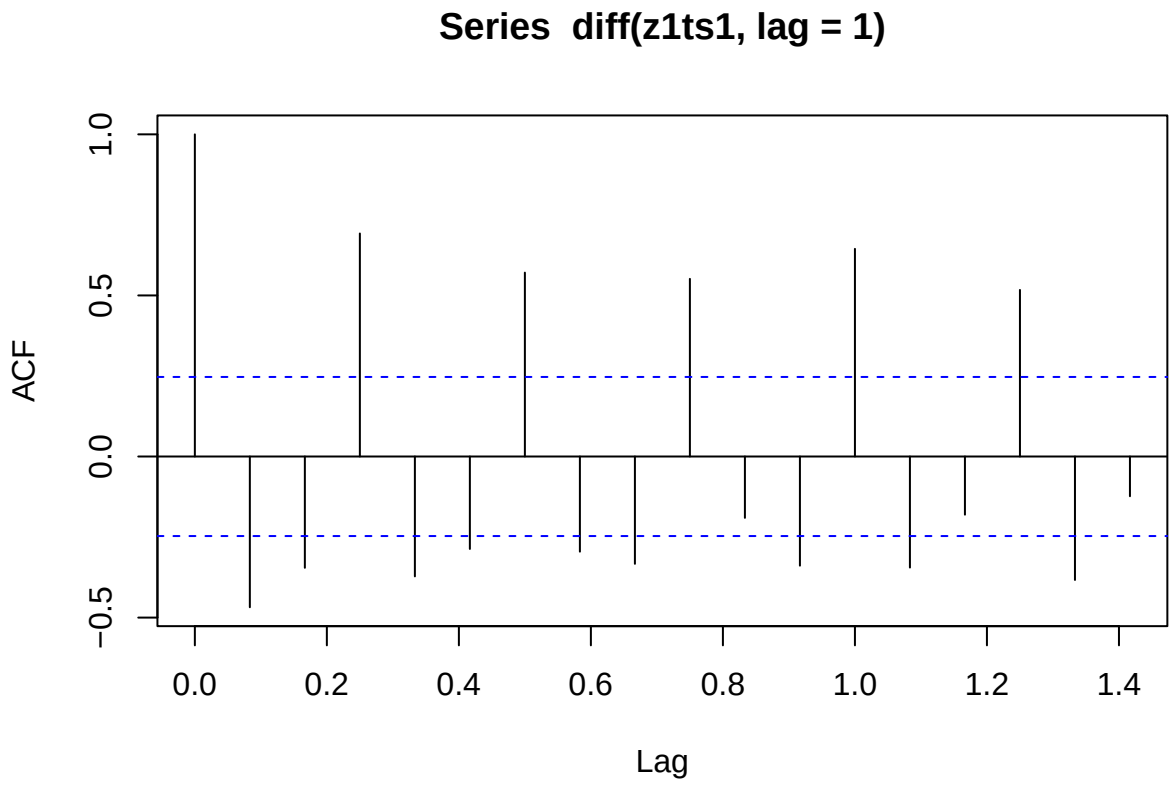



```
pacf(z1ts1)
```

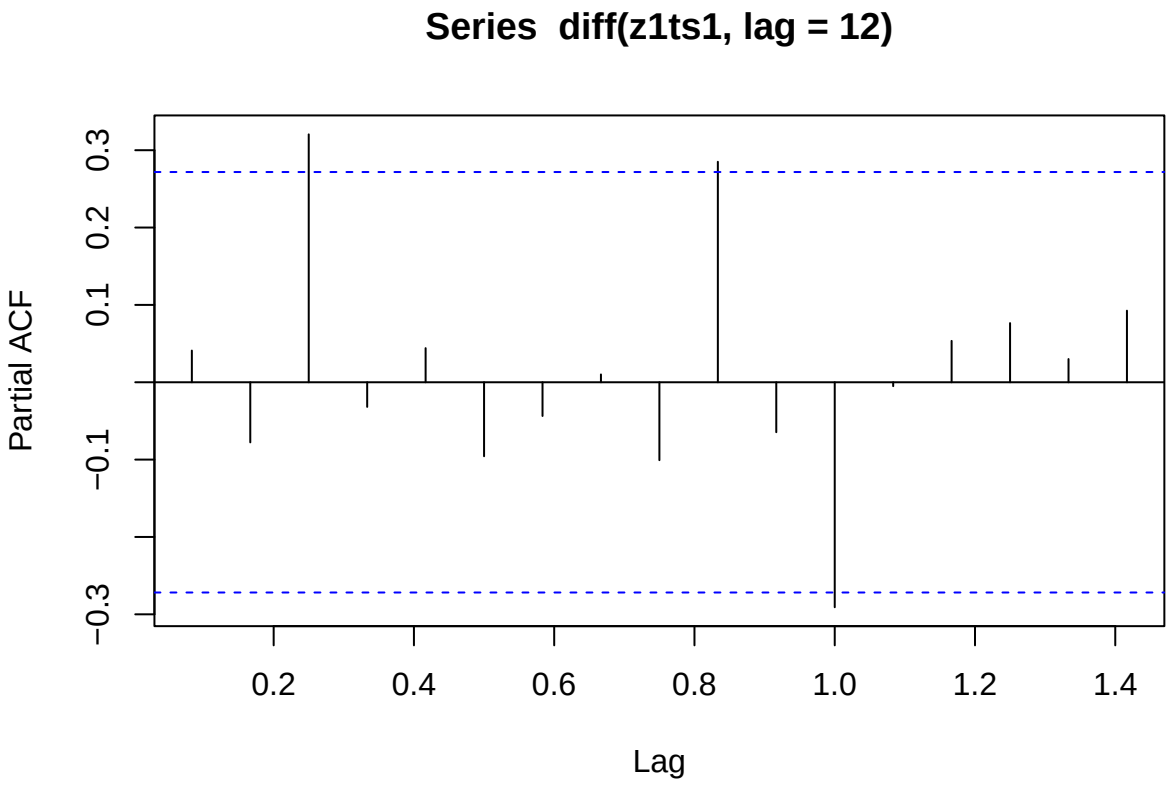


Możemy na nich zauważyć, że nie występuje jednoznaczna autokorelacja. W takim wypadku możemy nasze dane zróżnicować

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




```
pacf(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
```

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

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

```r
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
```

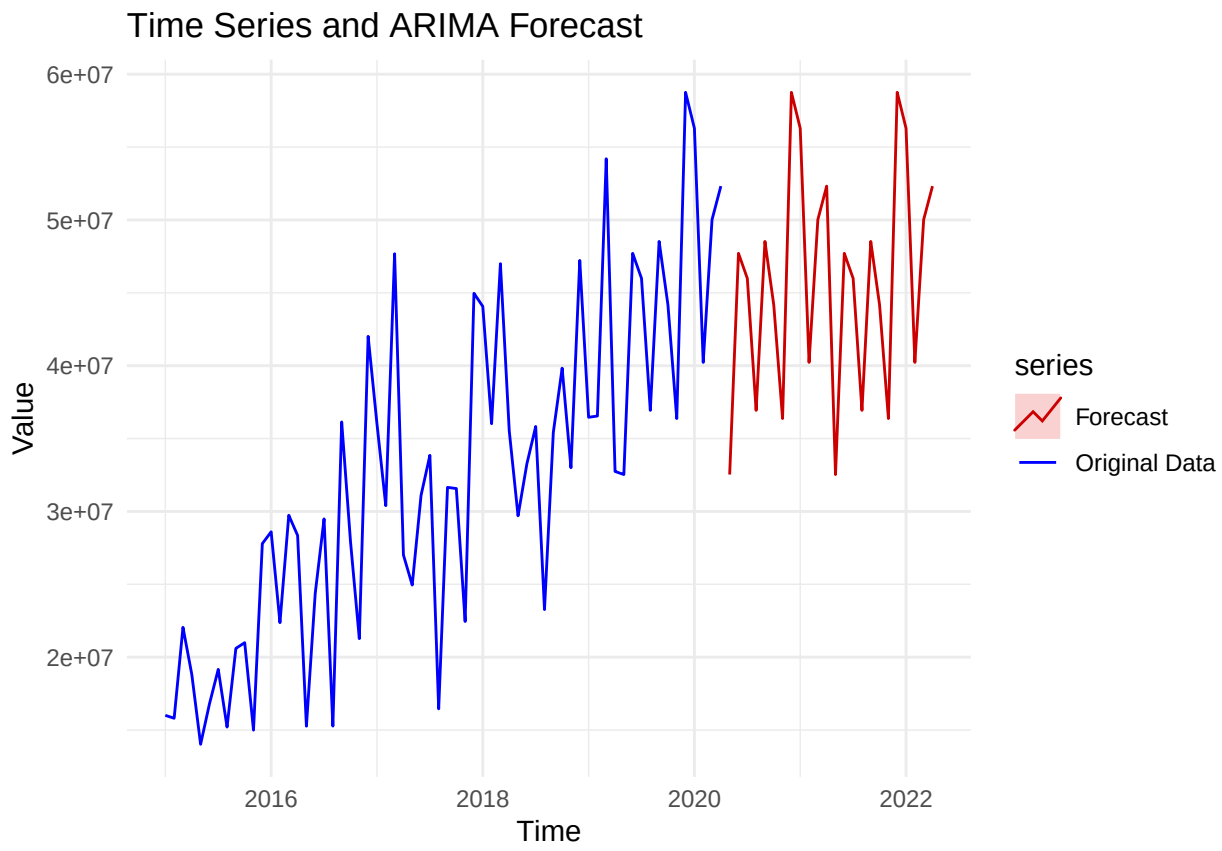
Z pośród stoworznych modeli wybraliśmy najlepszy, teraz spróbujemy stowrzyć model AutoArima i spróbujemy go porównać z obecnie najlepszym

```r
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
```

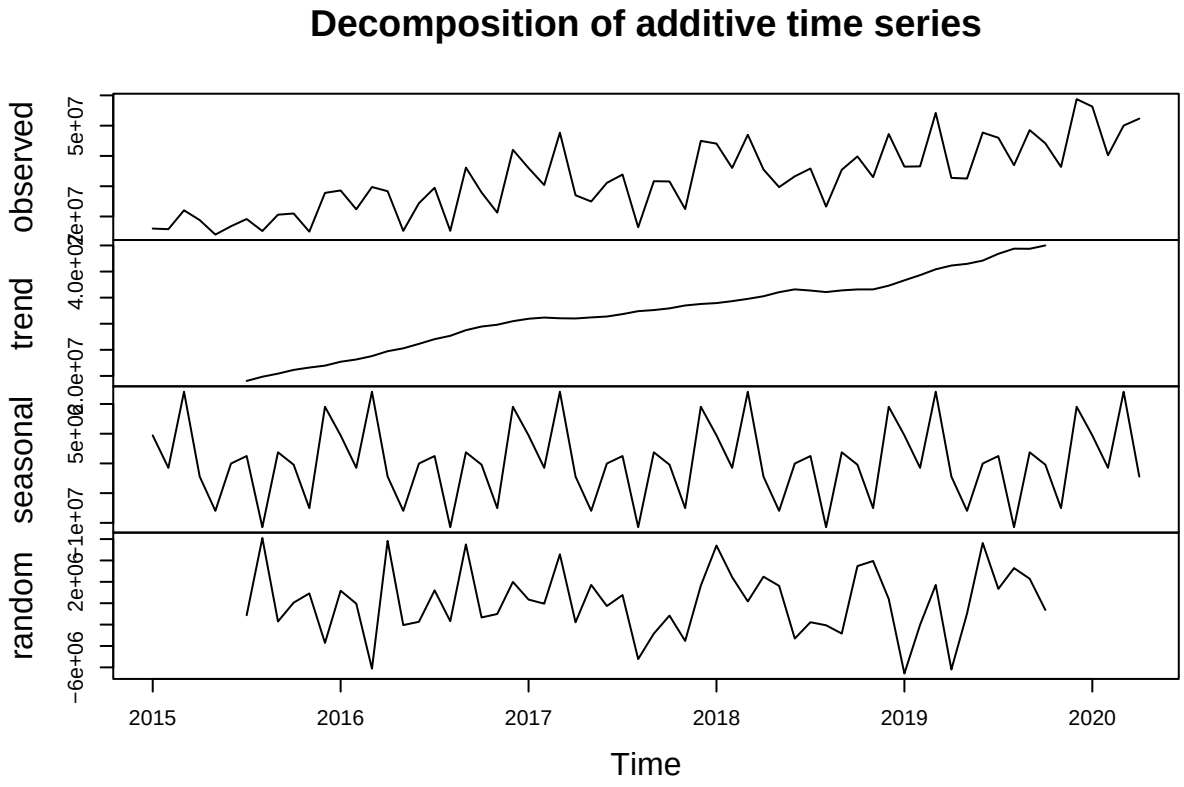
Podsumowując nasz wcześniejszy model jest lepszy :) # Predykcja danych Teraz mając nasze modele możemy dokonać predykcji ## Zbiór 1 Dla naszego zbioru spróbujemy dokonać predykcji na następny rok ###Predykcja przychodów przedsiębiorstwa

```r
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"))
```



#Dekomozycja szeregu

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



Na podstawie powyższych obserwacji możemy stwierdzić, że szereg ma widoczny trend, jest sezonowy oraz występują odchylenia losowe.