Sprawozdanie z projektu

Szeregi czasowe

Kamil Baran 166631

Inżynieria i analiza danych

Spis treści

1. Użyte dane

W swoim projekcie użyłem 2 rodzajów danych pobranych ze strony https://fred.stlouisfed.org/.

Pierwszy rodzaj miał posiadać wyraźny trend a drugi wyraźną sezonowość.

Tematy, które wybrałem:

- Średnie tygodniowe zarobki pracowników produkcyjnych i nie nadzorczych

Dane te reprezentują średnie zarobki pracowników w latach od 1939 do 2022 roku. Podczas tworzenia szeregu ograniczyłem dane do roku 1968.

- Nowe prywatne jednostki mieszkalne autoryzowane przez pozwolenie na budowę dla Kalifornie

Dane te pokazują ilość jednostek mieszkalnych od 1998 roku do 2022r. w Kalifornii

2. Wczytanie danych do środowiska R-studio

```
#wczytanie danych
zarobki <- read.csv("C:/Users/Kamil/Desktop/projekt szeregi/zarobki.csv")

colnames(zarobki) <- c("Data", "Ilosc_dolarow_na_tydzien")

domy <- read.csv("C:/Users/Kamil/Desktop/projekt szeregi/domy.csv")

colnames(domy) <- c("Data", "Ilosc_domow")
```

2.1 Widok danych w R-studio

	Data	Ilosc_dolarow_na_tydzien
1	1968-01-01	113.52
2	1968-02-01	115.06
3	1968-03-01	115.87
4	1968-04-01	114.86
5	1968-05-01	117.38
6	1968-06-01	117.79
7	1968-07-01	117.91
8	1968-08-01	117.62
9	1968-09-01	119.43
10	1968-10-01	120.54

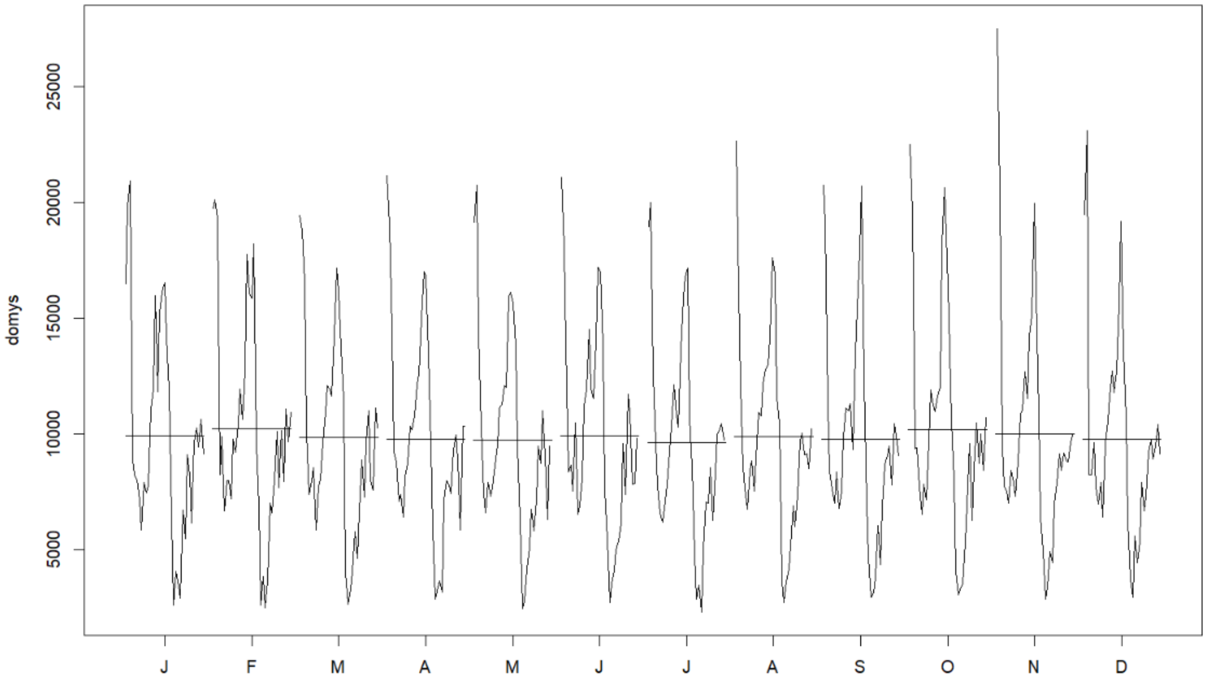
	Data	Ilosc_domow
1	1988-01-01	16466.483
2	1988-02-01	19704.379
3	1988-03-01	19420.946
4	1988-04-01	21148.678
5	1988-05-01	19129.808
6	1988-06-01	21077.471
7	1988-07-01	18940.715
8	1988-08-01	22646.439
9	1988-09-01	20723.744
10	1988-10-01	22482.934

3. Stworzenie szeregów czasowych

```
#wybieramy 2 kolumne z kazdej z wczytanych danych
zarobki2 <- zarobki[,2]
domy2 <- domy[,2]

#stworzenie szeregow czasowych
(domys <- ts(domy2, start = c(1988,1), frequency = 12))
(zarobkis <- ts(zarobki2, start = c(1968,1), frequency = 12))
```

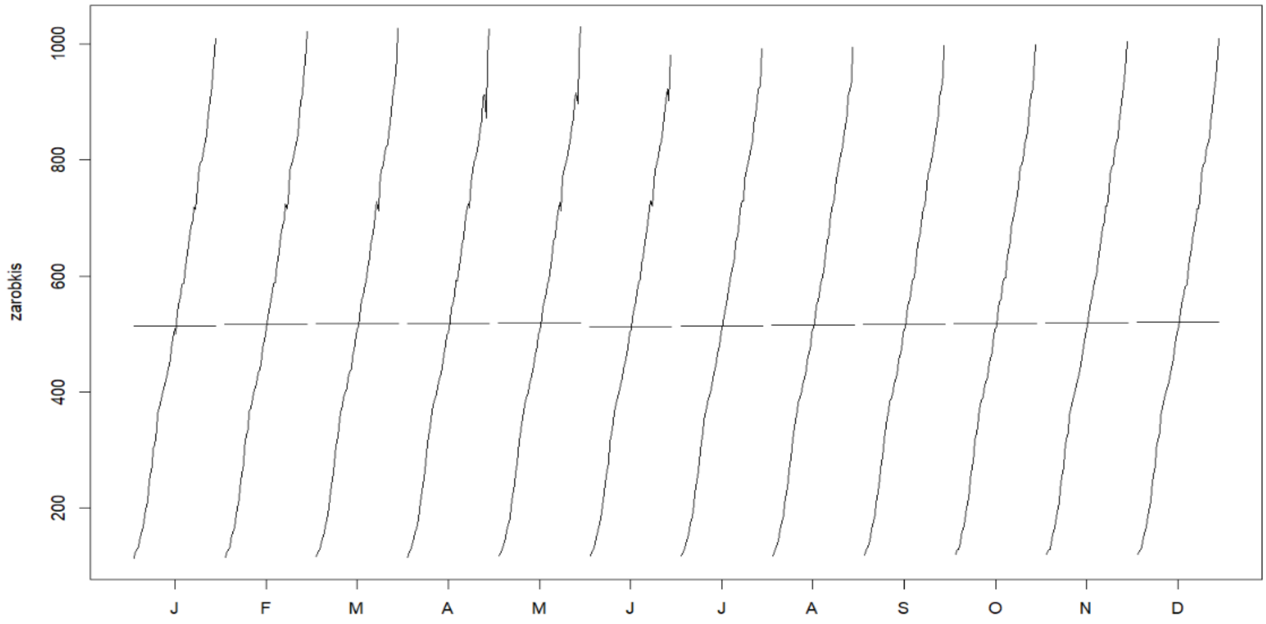
4. Omówienie głównych cech analizowanych szeregów

- Dane I – jednostki mieszkalne



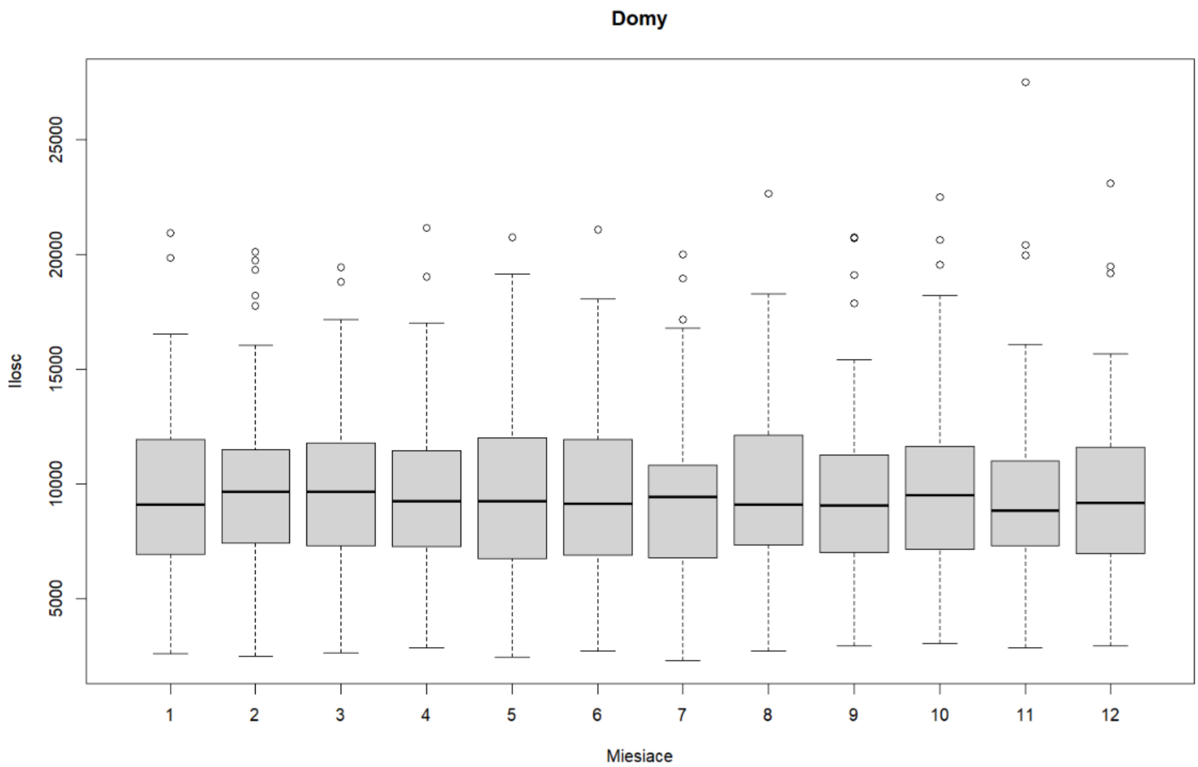
Rysunek 1 Monthplot - Dane 1

- Dane II- średnie zarobki



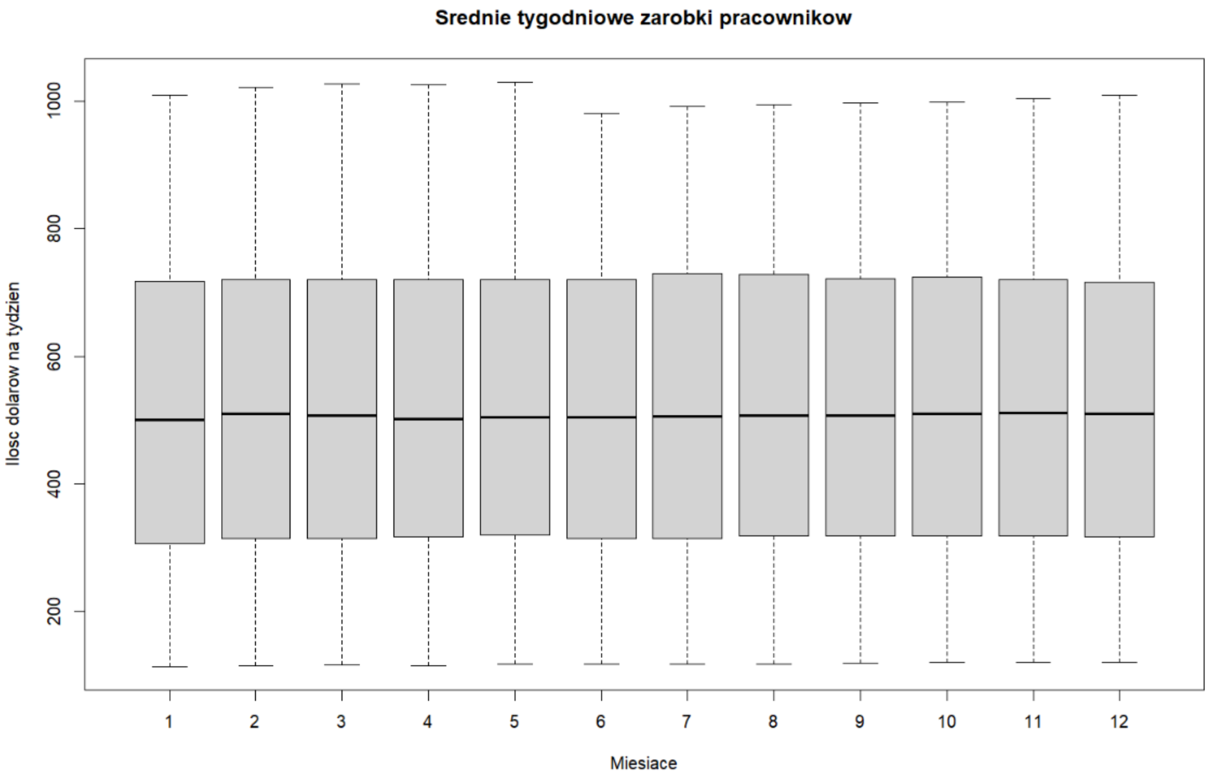
Rysunek 2- Monthplot - dane 2

- Dane I



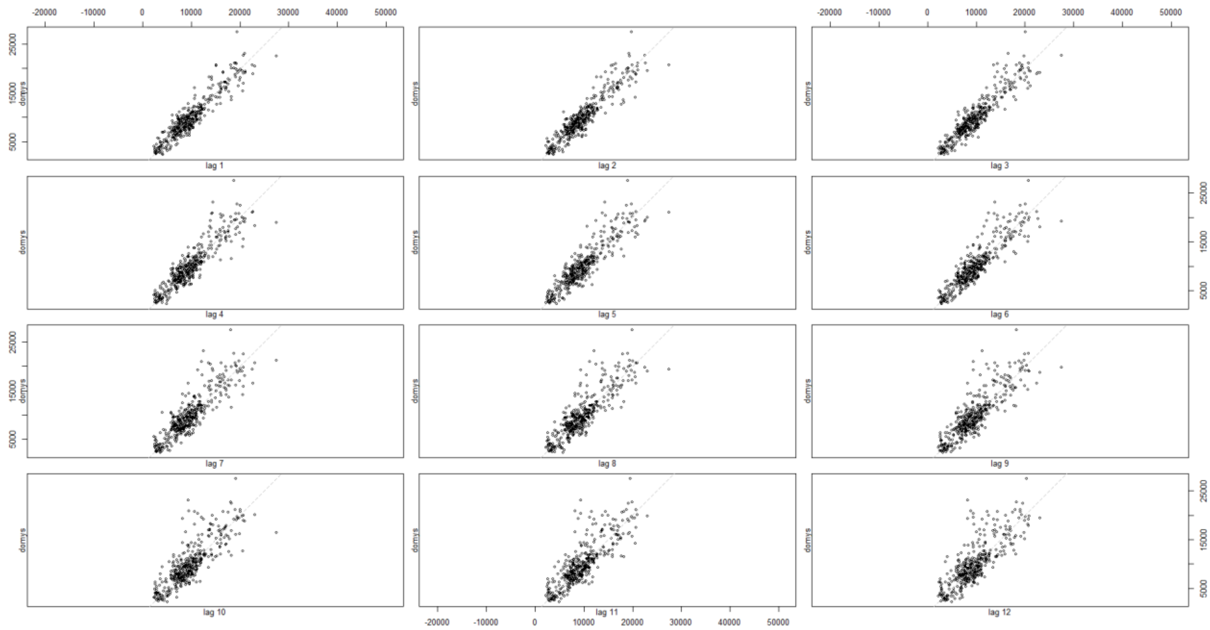
Rysunek 3- Boxplot- dane 1

- Dane II



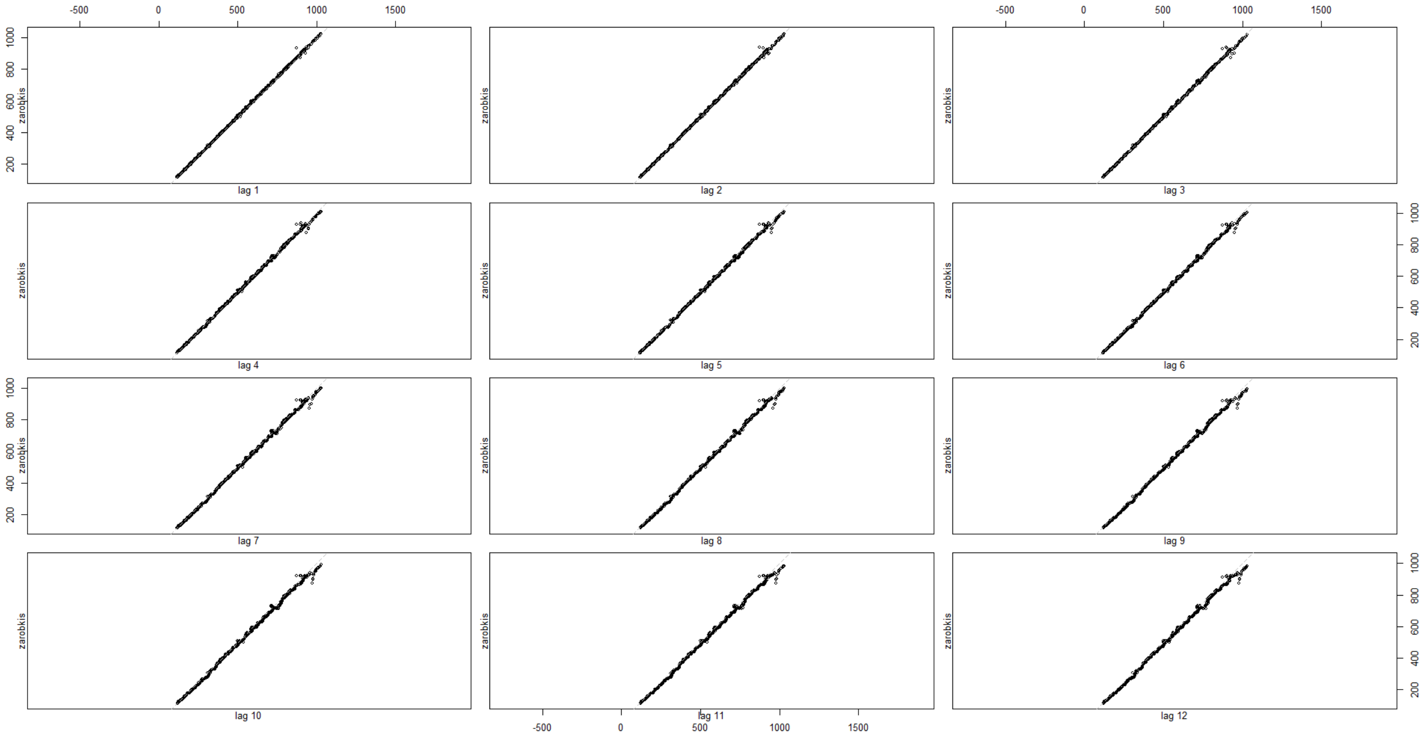
Rysunek 4- Boxplot dane2

- Dane I



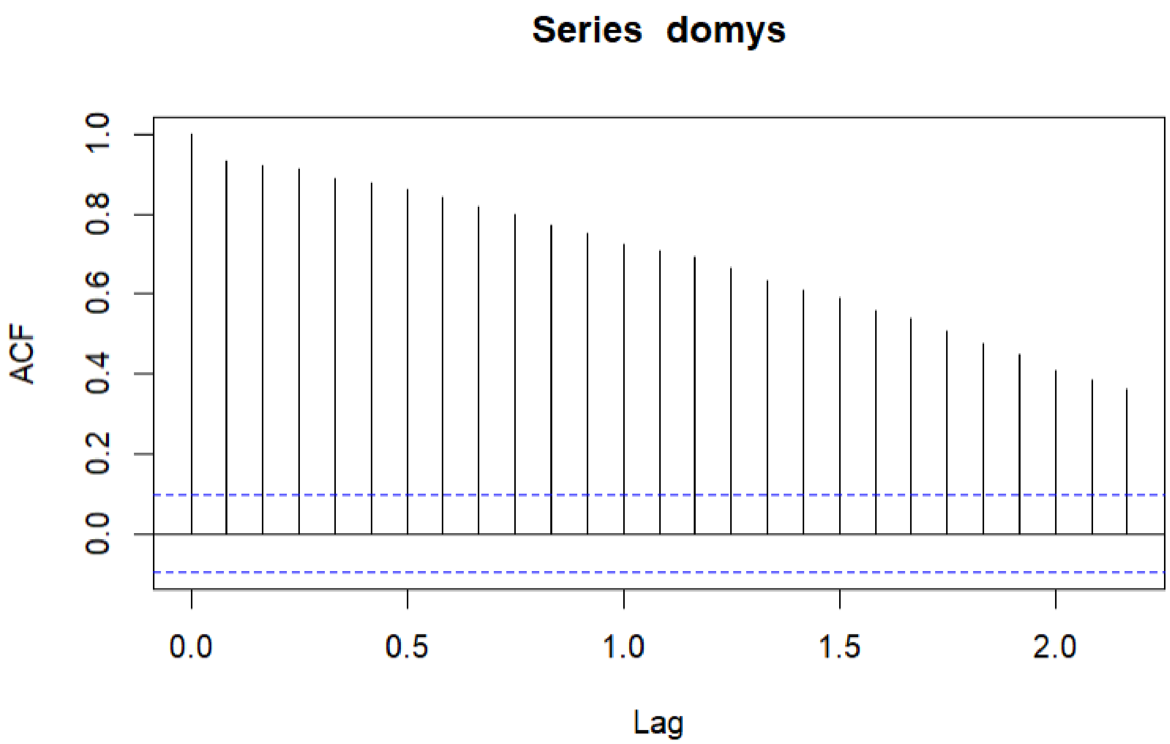
Rysunek 5 Lag plot-dane I

- Dane II

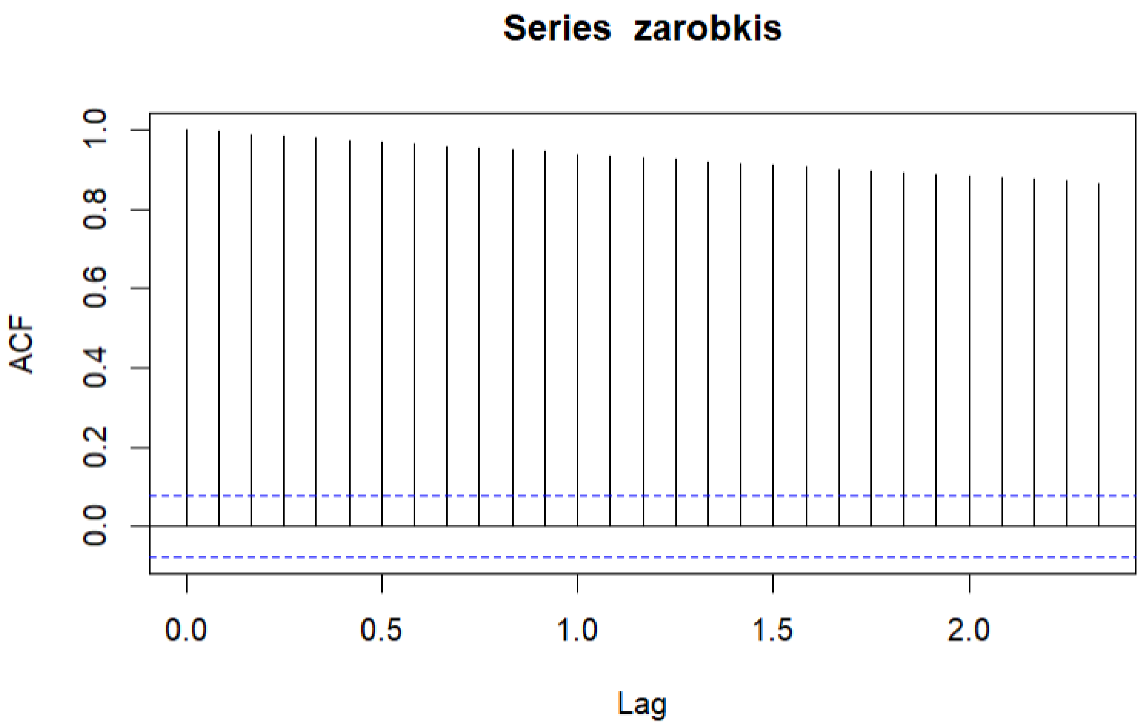


Rysunek 6 Lag plot - dane 2

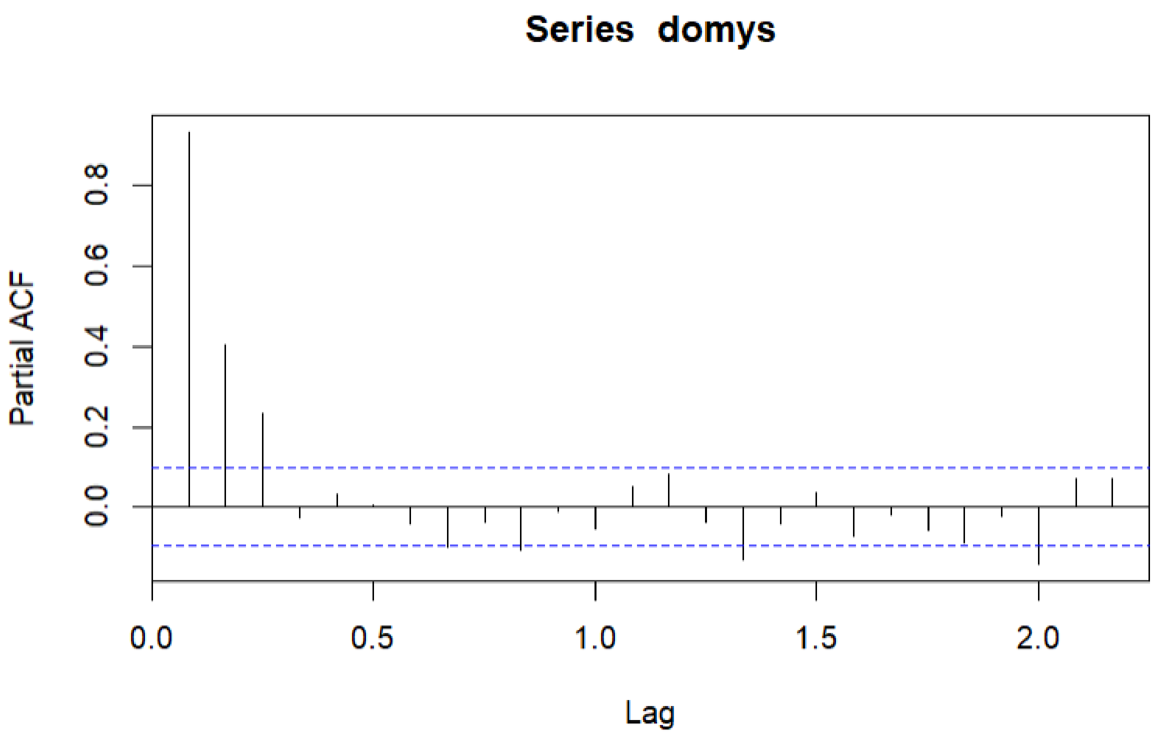
- Dane I



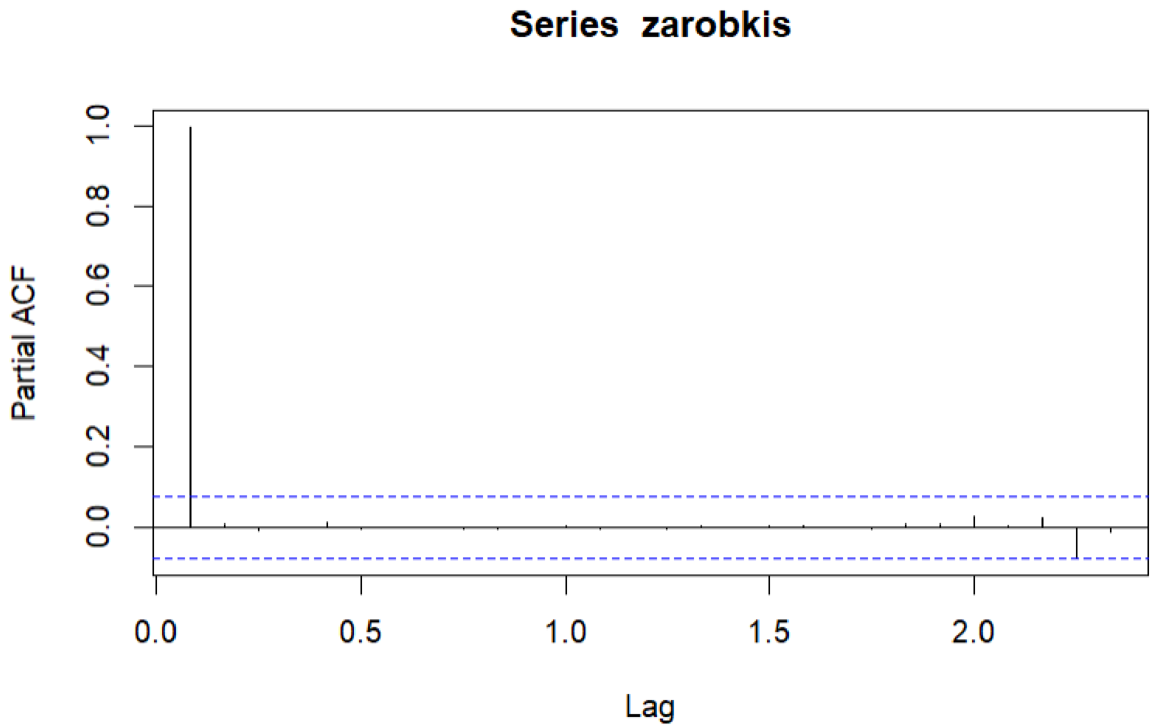
- Dane II



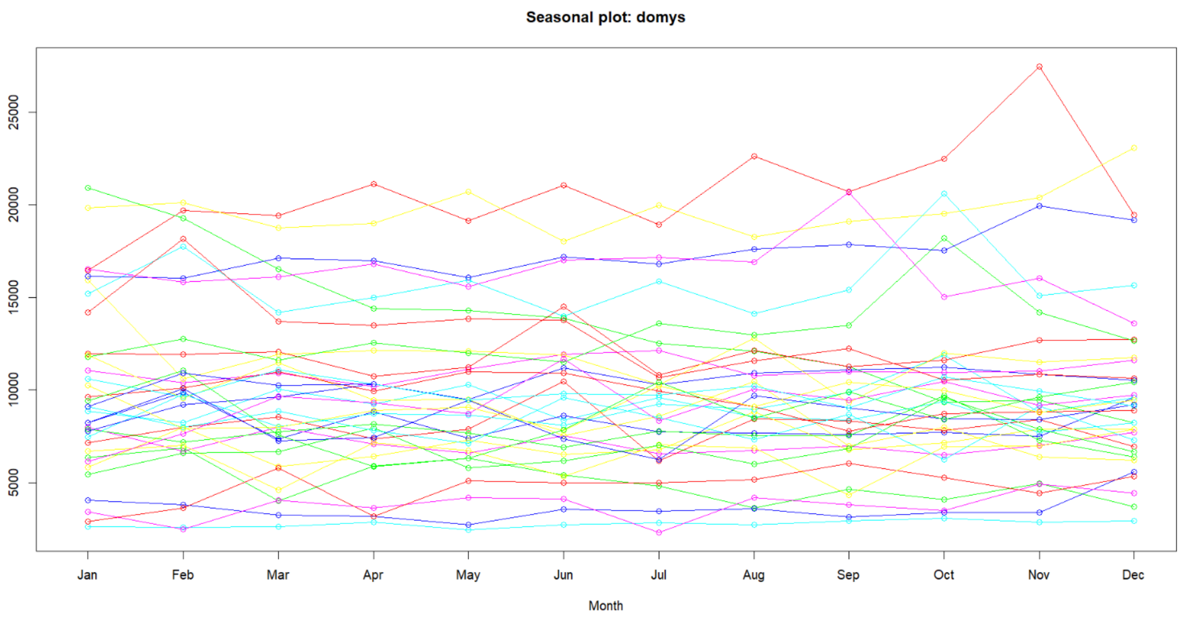
- Dane I



- Dane II

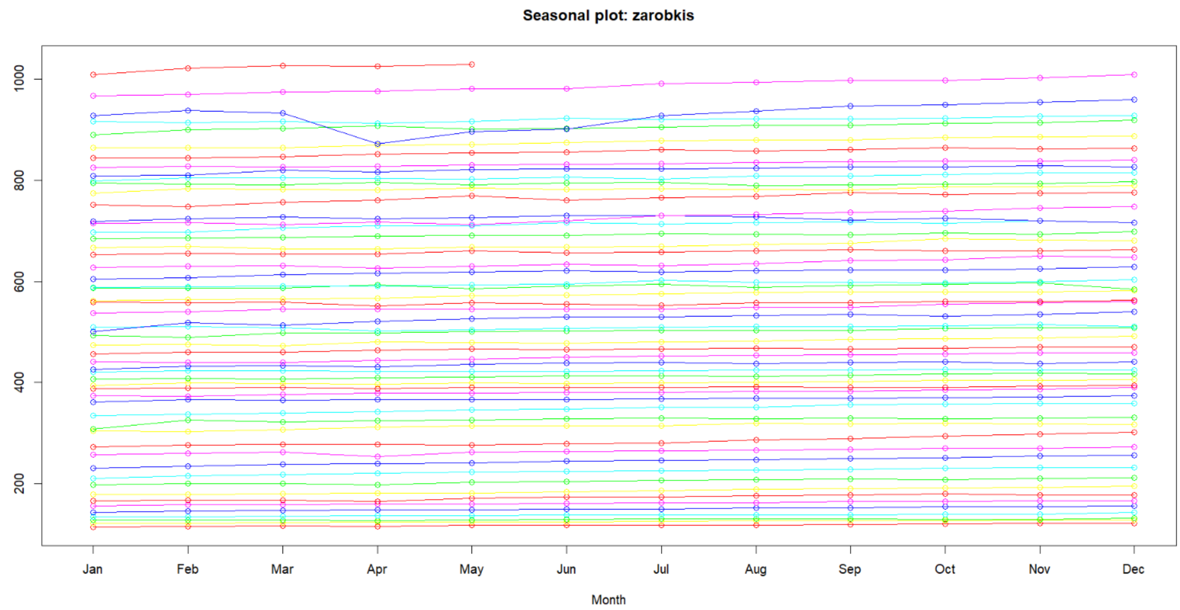


- Dane I



Rysunek 7 seasonal plot

- Dane II



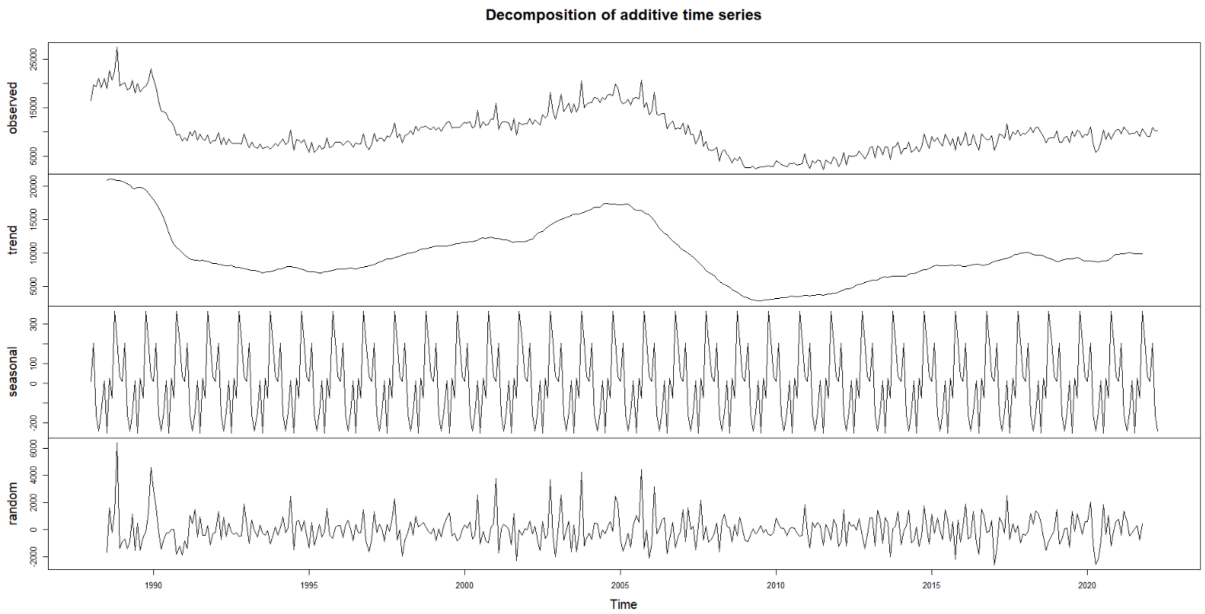
Rysunek 8 seasonal plot dane2

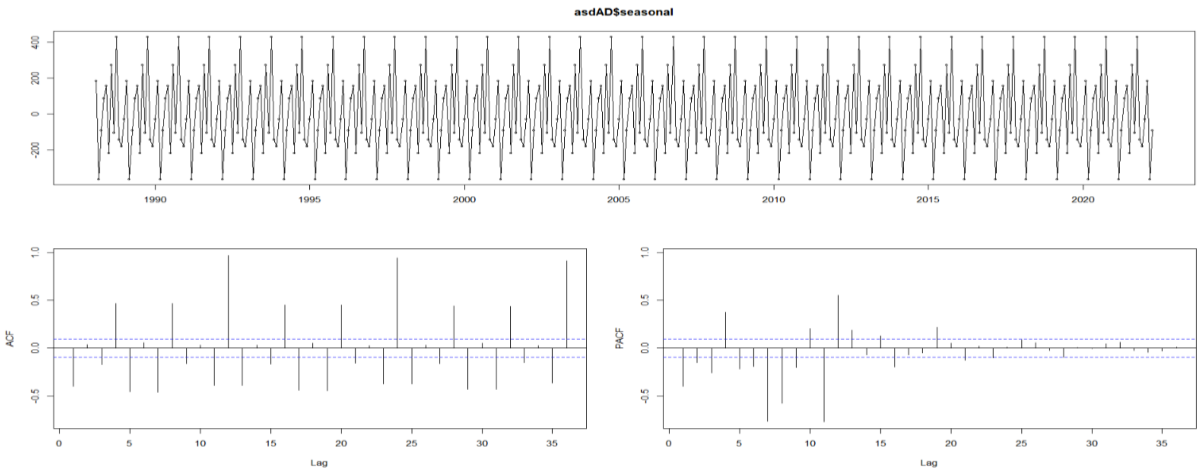
5. Dekompozycja

Zrobiłem dekompozycję multiplikatywną, oraz addytywną. Wyznaczyłem również trendy z uwzględnieniem sezonowości.

- Dane I

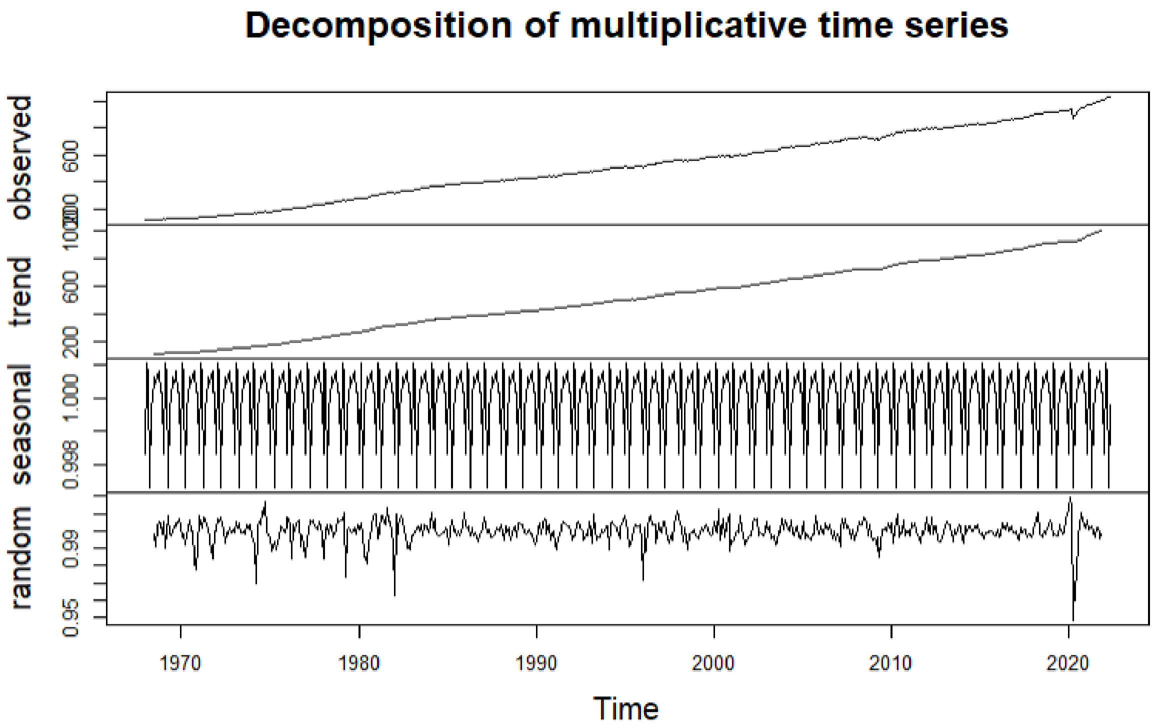
```
#dekompozycja----
#I dane
adddomy <- decompose(domys, type = "additive")
plot(adddomy)
asdd<-diff(domys, lag.max = 1)
asdAD <- decompose(asdd, type="additive")
plot(asdAD)
plot(asdd)
tsdisplay(asdAD$seasonal)

```





- Dane II

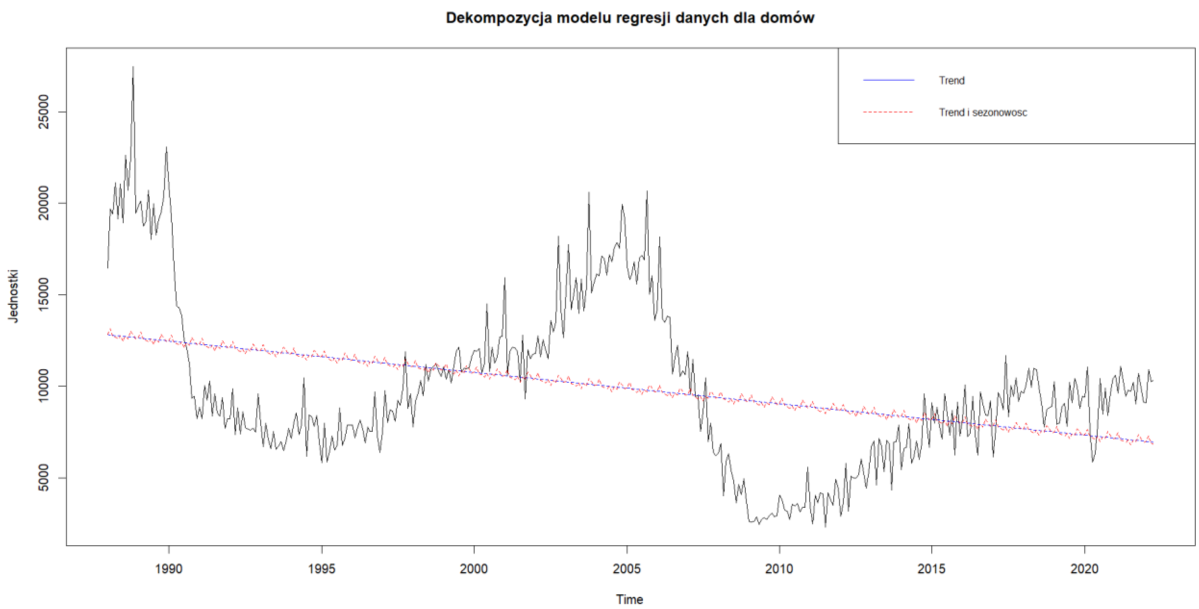
```
#II dane
zarobkim <- decompose(zarobkis, type = "multiplicative")
plot(zarobkim)
adzarobki <- decompose(zarobkis, type = "additive")
plot(adzarobki)
```



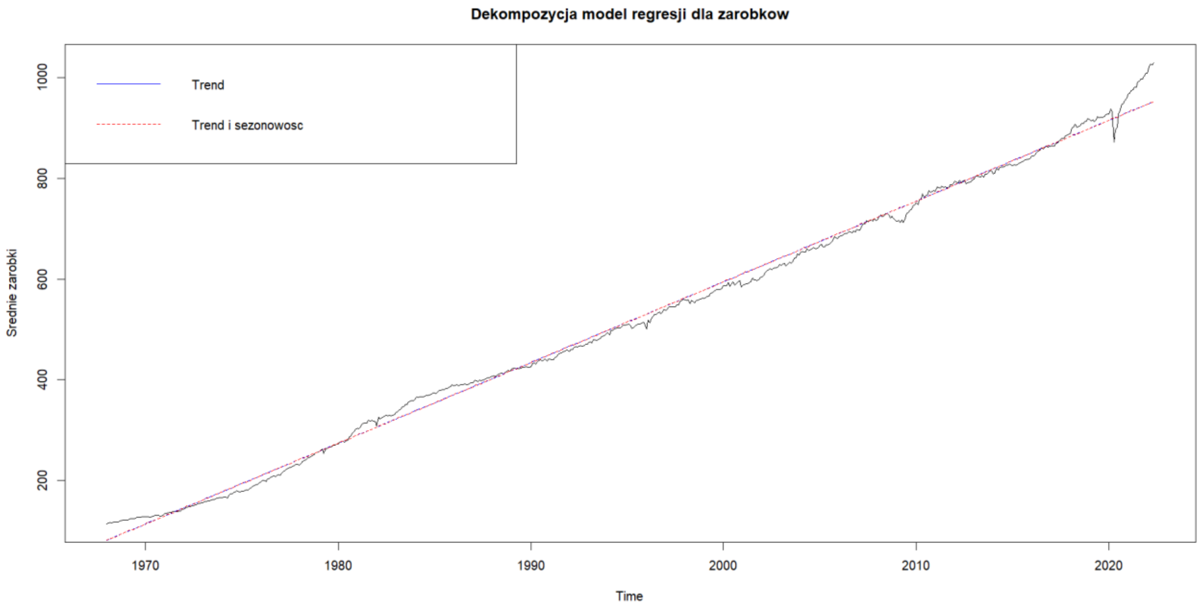
5.1 Dekompozycja model regresji

Wyznaczyłem trendy liniowe z uwzględnieniem sezonowości

- Dane I

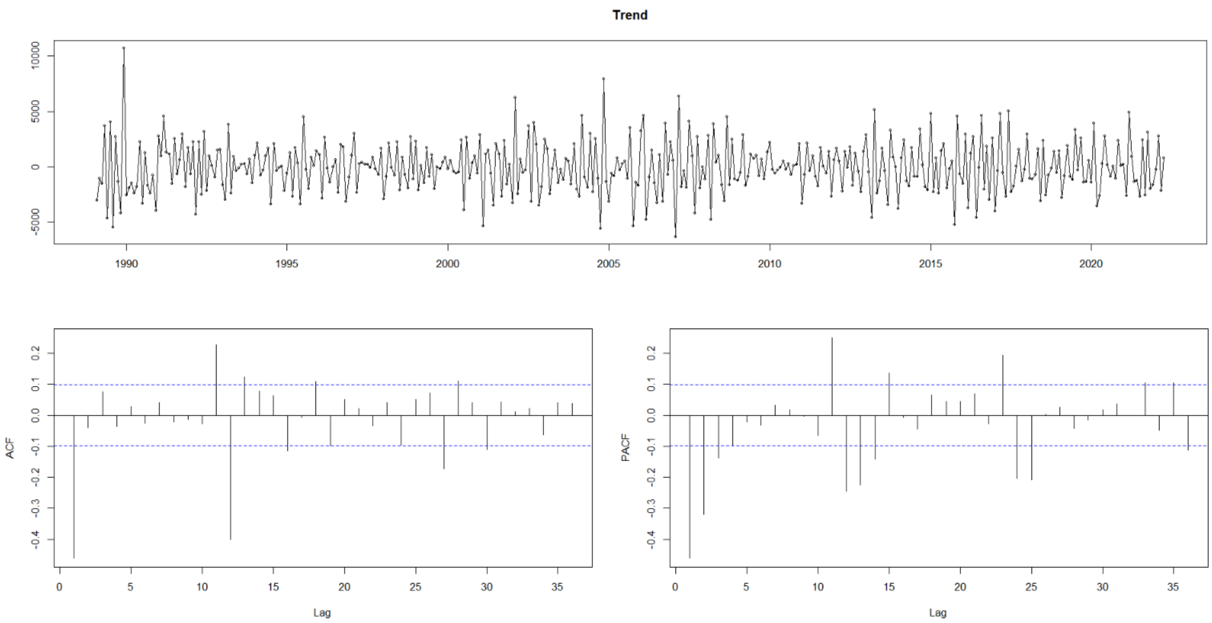


- Dane II



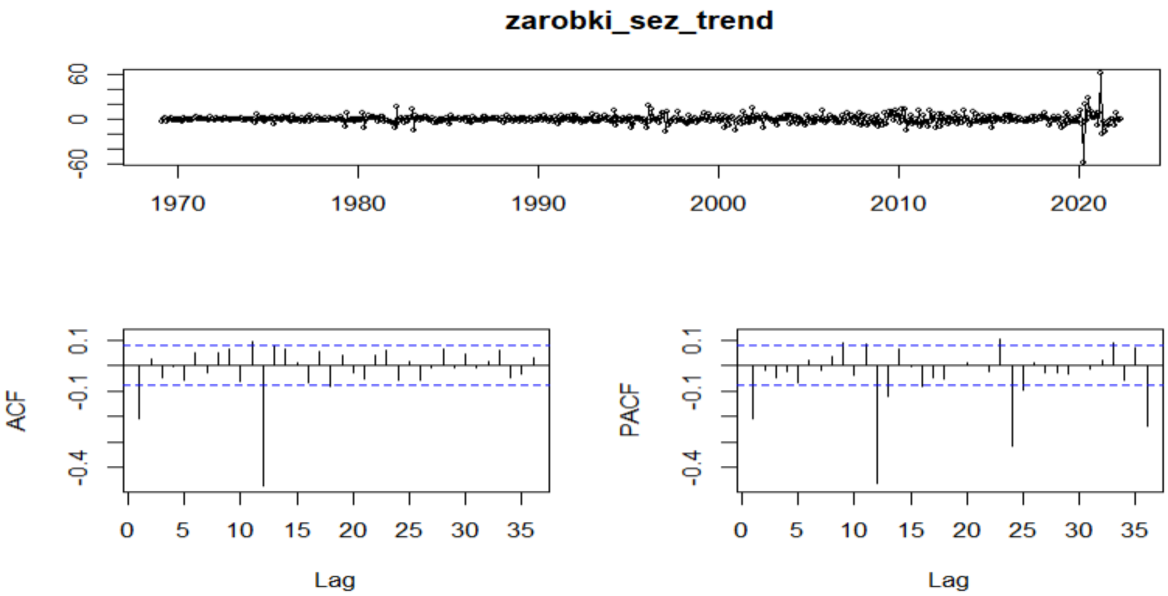
6. Usuwanie trendu i sezonowości

- Dane I




```
domy_sez <- diff(domys, lag = 12)
tsdisplay(domy_sez)
domy_sez_trend <- diff(domy_sez, lag = 1)
tsdisplay(domy_sez_trend, main = "Trend")
```

- Dane II

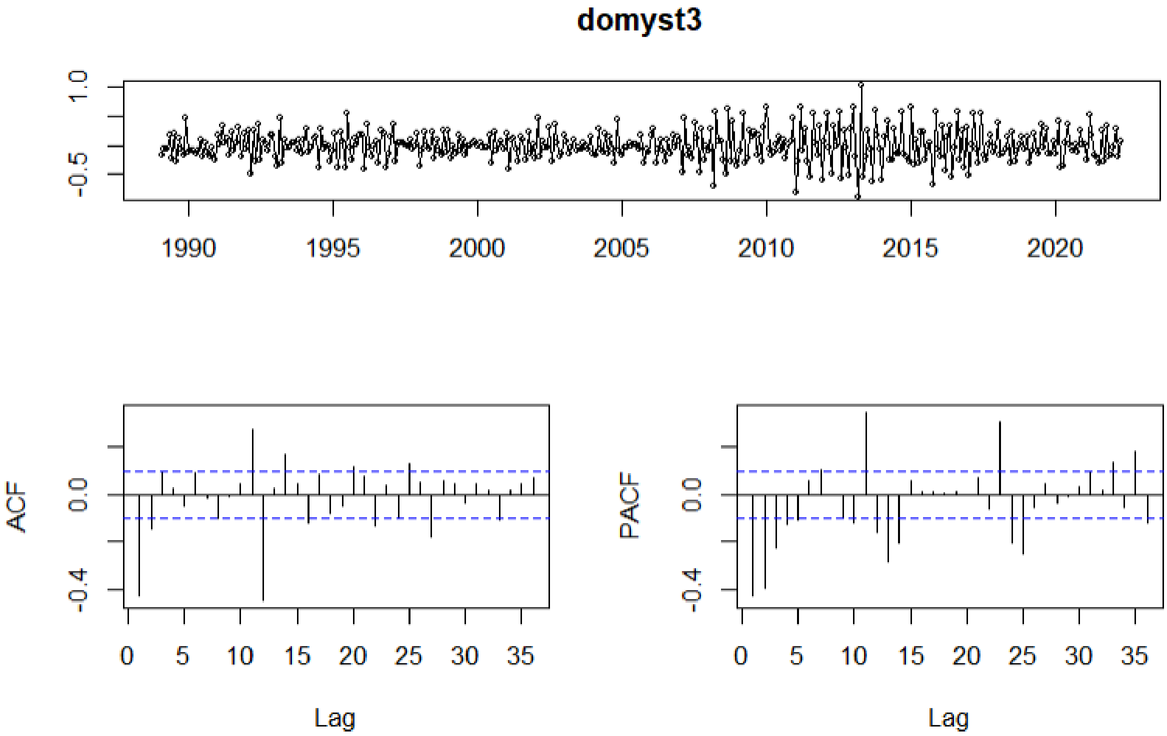


7. Stworzenie szeregów stacjonarnych

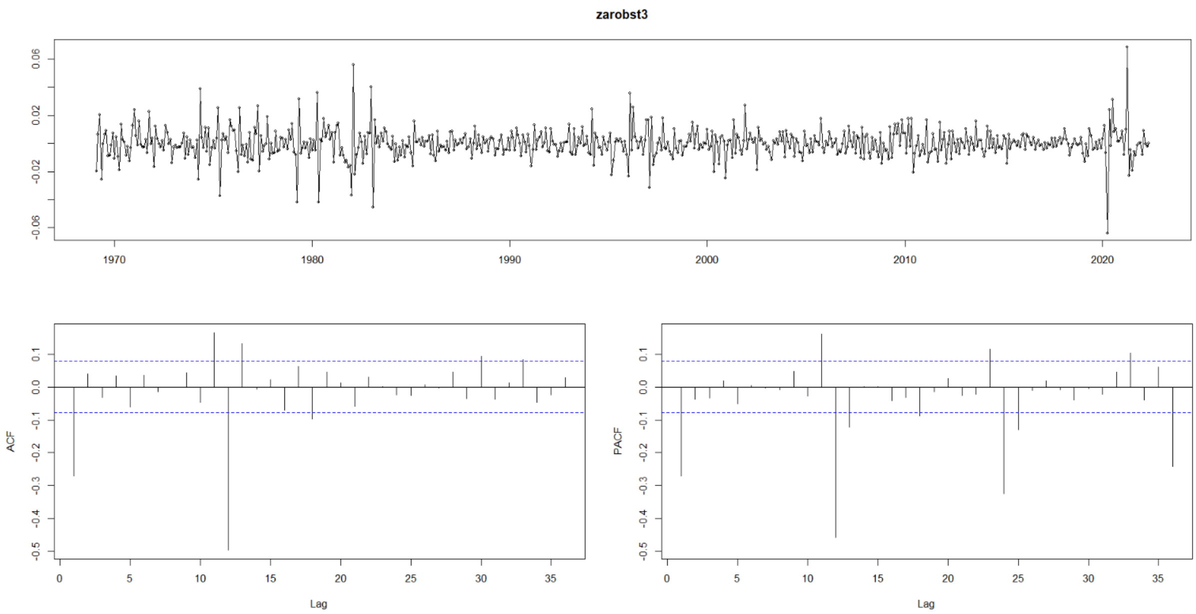
Przed pozbyciem się sezonowości musiałem wcześniej pozbyć się trendu. Zrobiłem to powyżej. Teraz tworzę szereg stacjonarny, a następnie sprawdzam czy szereg jest realizacją szumu białego.

- Dane I

```
tsdisplay(domys)
domyst <- BoxCox(domys, lambda = 0)
tsdisplay(domyst)
domyst2 <- diff(domyst, lag = 12)
tsdisplay(domyst2)
domyst3 <- diff(domyst2, lag = 1)
tsdisplay(domyst3)
```



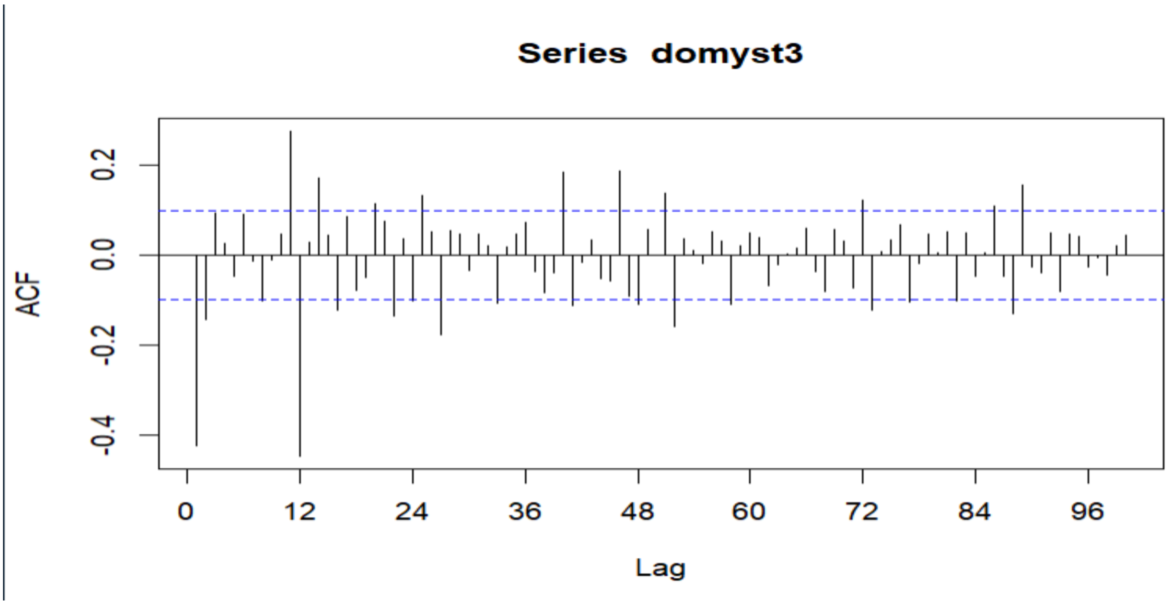
- Dane II



```r
#IIdane
tsdisplay(zarobkis)
zarobst <- BoxCox(zarobkis, lambda = 0)
tsdisplay(zarobst)
zarobst2 <- diff(zarobst, lag = 12)
tsdisplay(zarobst2)
zarobst3 <- diff(zarobst2, lag = 1)
tsdisplay(zarobst3)
```

7.2 Sprawdzanie czy szereg jest realizacją szumu białego

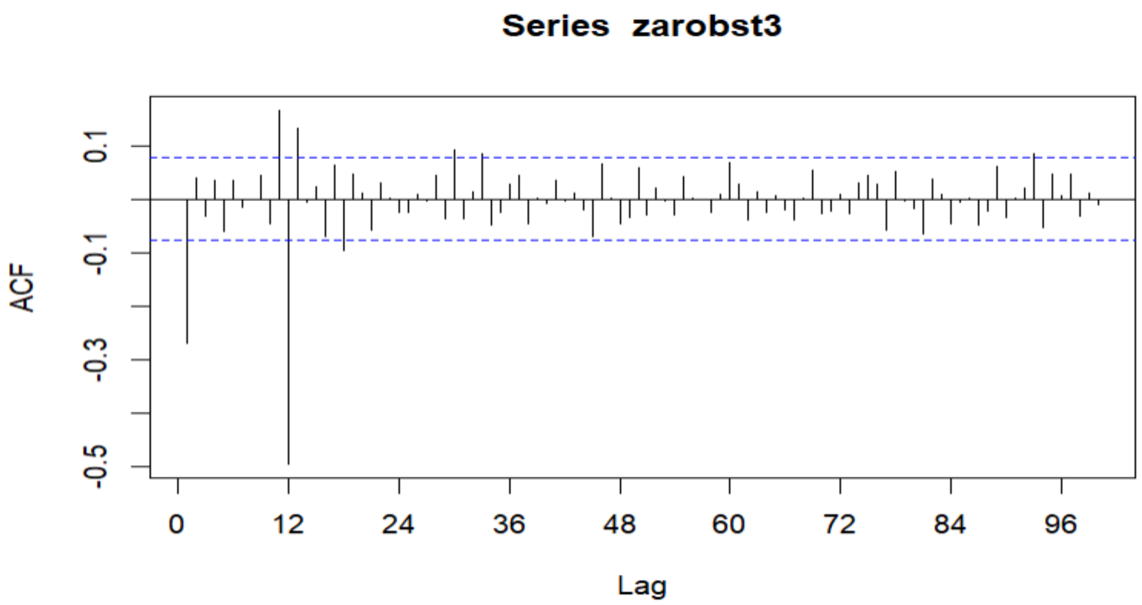
- Dane I



Wniosek:

Szereg ten nie jest realizacją szumu białego, ponieważ mniej niż 95% autokorelacji próbkowych mieści się w przedziale ufności, oraz występują wartości istotnie wychodzące poza ten przedział

- Dane II



Wniosek:

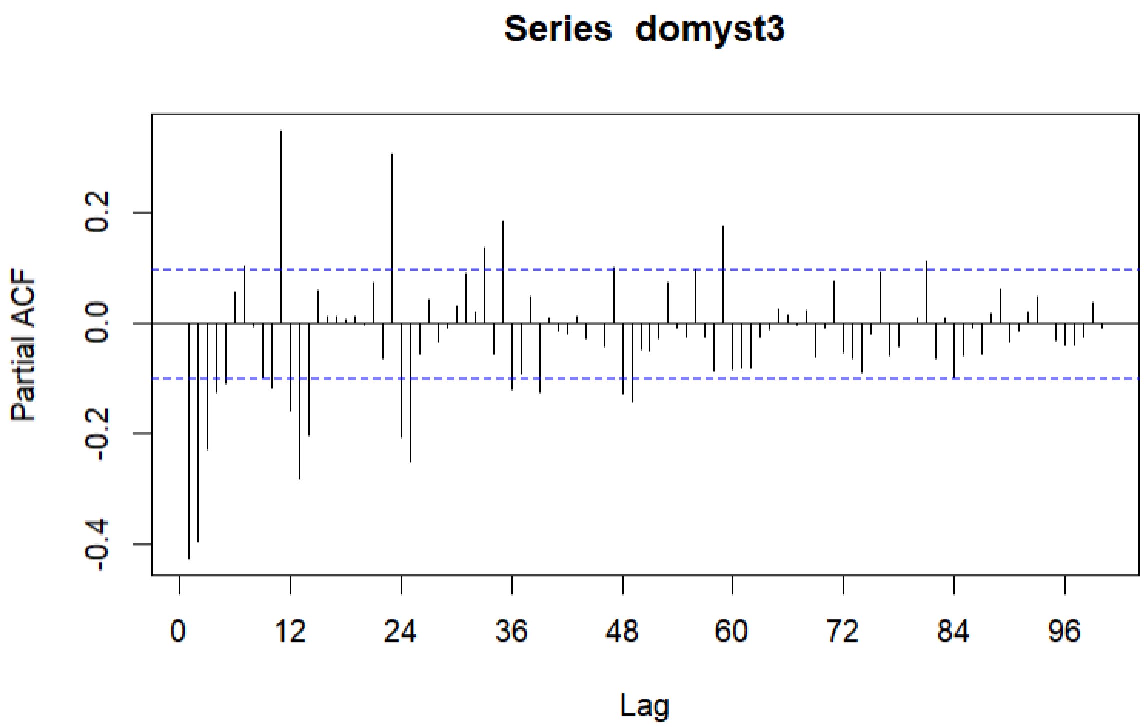
Szereg ten nie jest realizacją szumu białego, ponieważ występują wartości istotnie wychodzące poza ten przedział.

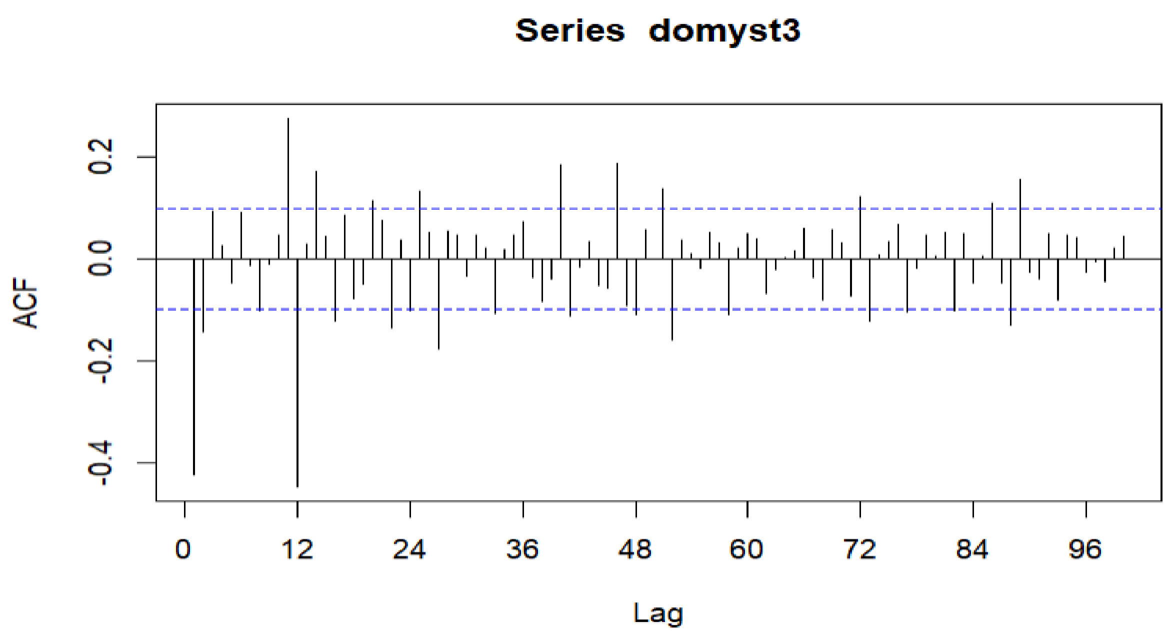
8. Wyznaczanie rzędów dla modeli AF, oraz MA

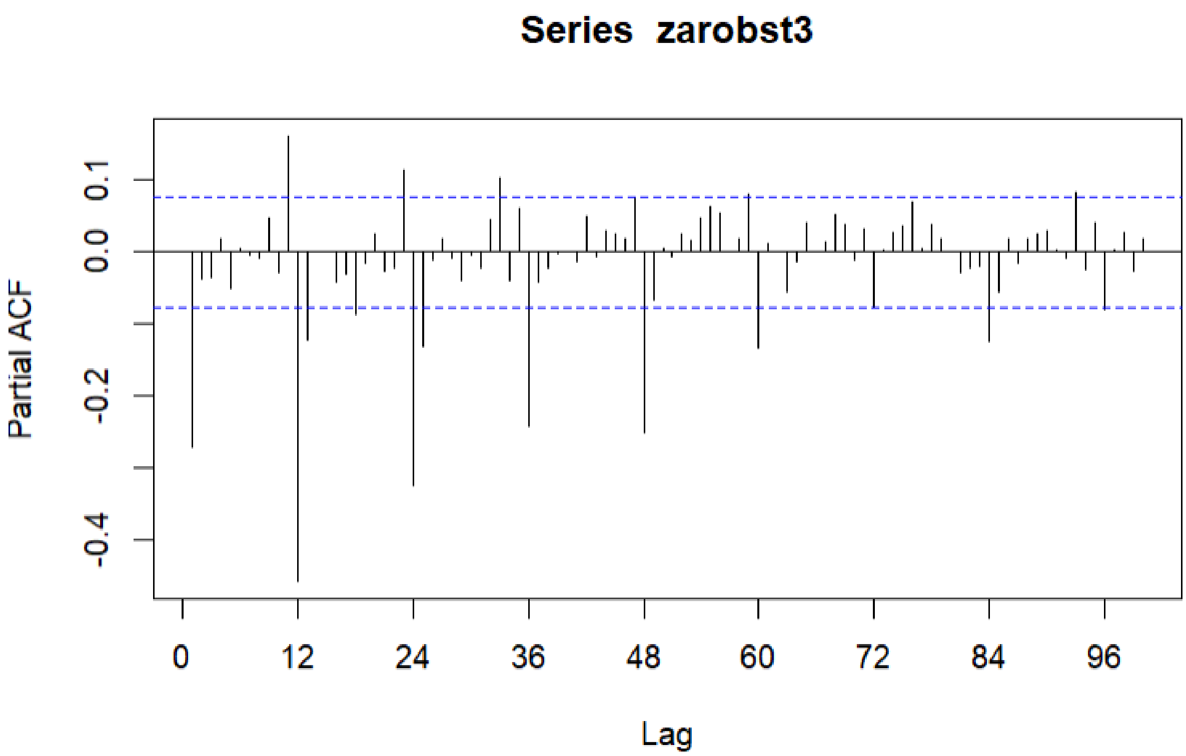
Kolejnym etapem jest wyznaczenie rzędów, będą nam one potrzebne do wyznaczenia modeli AR i MA

```
Acf(domyst3, lag.max = 100) #q=12
Pacf(domyst3, lag.max = 100) #p=62

#ar(zarobst3,aic = TRUE, order.max = 100)
Acf(zarobst3, lag.max = 100) #q=12
Pacf(zarobst3, lag.max = 100) #p=60

```

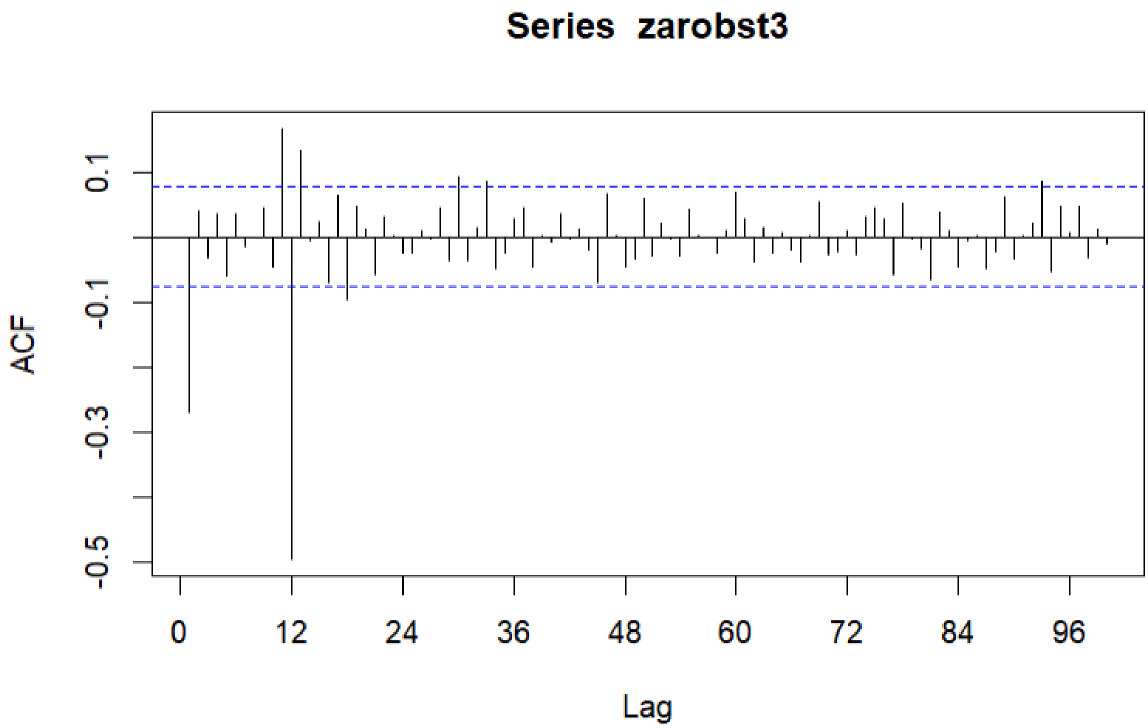
- Dane I


Series domyst3
Partial ACF
0.2
0.0
-0.2
-0.4
0
12
24
36
48
60
72
84
96
Lag


Series domyst3
ACF
0.2
0.0
-0.2
-0.4
0
12
24
36
48
60
72
84
96
Lag



Series zarobst3
Partial ACF
0.1
0.0
-0.2
-0.4
0
12
24
36
48
60
72
84
96
Lag



Series zarobst3
ACF
0.1
-0.1
-0.3
-0.5
0
12
24
36
48
60
72
84
96
Lag

9. Wyznaczenie współczynnika modelu AR

- Dane I

```
#Wspolczynnik modelu AR----
#Idane
(domyar <- ar(domyst3 , aic = FALSE, order.max = 62, method = "yule-walker"))
(domyar2 <- ar(domyst3, aic = FALSE, order.max = 62, method = "burg"))

(domyarauto <- ar(domyst3, aic = TRUE, order.max = 100))
```

```
Call:
ar(x = domyst3, aic = FALSE, order.max = 62, method = "yule-walker")

Coefficients:
      1        2        3        4        5        6        7        8        9       10       11  
-0.6587  -0.4264  -0.1779  -0.0918   0.0077   0.1780   0.1767   0.0366   0.0559   0.0490   0.1147  
     12       13       14       15       16       17       18       19       20       21       22  
-0.7840  -0.5217  -0.1801   0.0239  -0.0122   0.0869   0.2102   0.2405   0.1499   0.1757   0.0409  
     23       24       25       26       27       28       29       30       31       32       33  
 0.0914  -0.6852  -0.4233  -0.1476  -0.0476   0.0248   0.1461   0.2179   0.2705   0.1720   0.1464  
     34       35       36       37       38       39       40       41       42       43       44  
 0.0122   0.0993  -0.4406  -0.2995  -0.1477  -0.1087   0.0214   0.0464   0.0504   0.0987   0.0377  
     45       46       47       48       49       50       51       52       53       54       55  
-0.0366  -0.0493   0.0193  -0.3661  -0.2870  -0.1392  -0.0013   0.0215   0.0613   0.0399   0.0606  
     56       57       58       59       60       61       62  
 0.0803  -0.0253  -0.0234   0.0739  -0.1672  -0.1314  -0.0783  

Order selected 62  sigma^2 estimated as  0.02505
```

```
Call:
ar(x = domyst3, aic = FALSE, order.max = 62, method = "burg")

Coefficients:
      1        2        3        4        5        6        7        8        9       10       11  
-0.6508  -0.4124  -0.1709  -0.0964   0.0082   0.1719   0.1836   0.0545   0.0880   0.0811   0.1741  
     12       13       14       15       16       17       18       19       20       21       22  
-0.8281  -0.5448  -0.1985   0.0274  -0.0314   0.0761   0.2092   0.2568   0.1517   0.2076   0.0784  
     23       24       25       26       27       28       29       30       31       32       33  
 0.1777  -0.7576  -0.4859  -0.1913  -0.0670  -0.0193   0.1155   0.2214   0.3038   0.1680   0.1846  
     34       35       36       37       38       39       40       41       42       43       44  
 0.0771   0.2349  -0.4722  -0.3407  -0.1660  -0.1313  -0.0184   0.0113   0.0399   0.0969   0.0040  
     45       46       47       48       49       50       51       52       53       54       55  
-0.0466  -0.0174   0.1056  -0.3945  -0.3385  -0.1435  -0.0272   0.0051   0.0527   0.0481   0.0717  
     56       57       58       59       60       61       62  
 0.0817  -0.0242   0.0072   0.1427  -0.1517  -0.1504  -0.0620  

Order selected 62  sigma^2 estimated as  0.01833
```

```
Call:
ar(x = domyst3, aic = TRUE, order.max = 100)

Coefficients:
      1        2        3        4        5        6        7        8        9       10       11  
-0.6587  -0.4264  -0.1779  -0.0918   0.0077   0.1780   0.1767   0.0366   0.0559   0.0490   0.1147  
     12       13       14       15       16       17       18       19       20       21       22  
-0.7840  -0.5217  -0.1801   0.0239  -0.0122   0.0869   0.2102   0.2405   0.1499   0.1757   0.0409  
     23       24       25       26       27       28       29       30       31       32       33  
 0.0914  -0.6852  -0.4233  -0.1476  -0.0476   0.0248   0.1461   0.2179   0.2705   0.1720   0.1464  
     34       35       36       37       38       39       40       41       42       43       44  
 0.0122   0.0993  -0.4406  -0.2995  -0.1477  -0.1087   0.0214   0.0464   0.0504   0.0987   0.0377  
     45       46       47       48       49       50       51       52       53       54       55  
-0.0366  -0.0493   0.0193  -0.3661  -0.2870  -0.1392  -0.0013   0.0215   0.0613   0.0399   0.0606  
     56       57       58       59       60       61       62  
 0.0803  -0.0253  -0.0234   0.0739  -0.1672  -0.1314  -0.0783  

Order selected 62  sigma^2 estimated as  0.02505
```

Z powyższych screenów możemy wnioskować, że wartość policzona metodą Yule-Walkera, oraz Burga dla p=62 jest taka samo jak obliczona automatycznie.

- Dane II

```
#IIdane
(zarar <- ar(zarobst3 , aic = FALSE, order.max = 60, method = "yule-walker"))
(zarar2 <- ar(zarobst3, aic = FALSE, order.max = 60, method = "burg"))

(zarar2rauto <- ar(zarobst3, aic = TRUE, order.max = 100))

```

```
Call:
ar(x = zarobst3, aic = FALSE, order.max = 60, method = "yule-walker")

Coefficients:
      1        2        3        4        5        6        7        8        9       10       11  
-0.2556  -0.0436  -0.0089   0.0264  -0.0710  -0.0070  -0.0348   0.0417   0.0479  -0.0099   0.0478  
     12       13       14       15       16       17       18       19       20       21       22  
-0.8540  -0.2100  -0.0259   0.0276   0.0154  -0.0756  -0.0481  -0.0435   0.0814   0.0465  -0.0077  
     23       24       25       26       27       28       29       30       31       32       33  
 0.0432  -0.7217  -0.1646  -0.0275   0.0423   0.0535  -0.0360   0.0509  -0.0123   0.1161   0.1158  
     34       35       36       37       38       39       40       41       42       43       44  
 0.0132   0.0351  -0.5345  -0.0638  -0.0193   0.0146   0.0532   0.0149   0.0937   0.0292   0.0760  
     45       46       47       48       49       50       51       52       53       54       55  
 0.0409   0.0517   0.0446  -0.3722  -0.0497   0.0023   0.0072   0.0378   0.0288   0.0705   0.0647  
     56       57       58       59       60  
 0.0590   0.0083   0.0333   0.0459  -0.1327  

Order selected 60  sigma^2 estimated as  6.109e-05
```

```
Call:
ar(x = zarobst3, aic = FALSE, order.max = 60, method = "burg")

Coefficients:
      1        2        3        4        5        6        7        8        9       10       11  
-0.2611  -0.0167   0.0235   0.0143  -0.0772   0.0073  -0.0293   0.0248   0.0589   0.0198   0.0610  
     12       13       14       15       16       17       18       19       20       21       22  
-0.9304  -0.2473   0.0056   0.0862   0.0084  -0.0802  -0.0225  -0.0384   0.0562   0.0668   0.0385  
     23       24       25       26       27       28       29       30       31       32       33  
 0.0737  -0.8288  -0.2181  -0.0058   0.1078   0.0504  -0.0369   0.0867  -0.0066   0.0787   0.1253  
     34       35       36       37       38       39       40       41       42       43       44  
 0.0699   0.0788  -0.6190  -0.0948  -0.0063   0.0592   0.0477   0.0337   0.1337   0.0356   0.0389  
     45       46       47       48       49       50       51       52       53       54       55  
 0.0347   0.1028   0.0948  -0.4177  -0.0638   0.0120   0.0364   0.0321   0.0475   0.1059   0.0715  
     56       57       58       59       60  
 0.0317  -0.0030   0.0650   0.0830  -0.1435  

Order selected 60  sigma^2 estimated as  4.981e-05
```

```
Call:
ar(x = zarobst3, aic = TRUE, order.max = 100)

Coefficients:
      1        2        3        4        5        6        7        8        9       10       11  
-0.2556  -0.0436  -0.0089   0.0264  -0.0710  -0.0070  -0.0348   0.0417   0.0479  -0.0099   0.0478  
     12       13       14       15       16       17       18       19       20       21       22  
-0.8540  -0.2100  -0.0259   0.0276   0.0154  -0.0756  -0.0481  -0.0435   0.0814   0.0465  -0.0077  
     23       24       25       26       27       28       29       30       31       32       33  
 0.0432  -0.7217  -0.1646  -0.0275   0.0423   0.0535  -0.0360   0.0509  -0.0123   0.1161   0.1158  
     34       35       36       37       38       39       40       41       42       43       44  
 0.0132   0.0351  -0.5345  -0.0638  -0.0193   0.0146   0.0532   0.0149   0.0937   0.0292   0.0760  
     45       46       47       48       49       50       51       52       53       54       55  
 0.0409   0.0517   0.0446  -0.3722  -0.0497   0.0023   0.0072   0.0378   0.0288   0.0705   0.0647  
     56       57       58       59       60  
 0.0590   0.0083   0.0333   0.0459  -0.1327  

Order selected 60  sigma^2 estimated as  6.109e-05
```

Z powyższych screenów możemy wnioskować, że wartość policzona metodą Yule-Walkera, oraz Burga dla p=60 jest taka samo jak obliczona automatycznie.

10. Współczynnik modelu MA

- Dane I

```
#Wspolczynnik modelu MA----
#dane1
wspama_domy <- Arima(domys, order = c(0,0,12))
summary(wspama_domy)
```

```
Series: domys 
ARIMA(0,0,12) with non-zero mean 

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8     ma9    ma10    ma11    ma12
      0.5667  0.6394  0.7352  0.6235  0.6364  0.6693  0.6205  0.5756  0.5647  0.4013  0.3489  0.0835
s.e.  0.0490  0.0557  0.0614  0.0671  0.0683  0.0675  0.0633  0.0610  0.0596  0.0626  0.0550  0.0494
           mean
      9966.6317
s.e.   539.1009

sigma^2 = 2271059:  log likelihood = -3594.6
AIC=7217.19   AICc=7218.25   BIC=7273.49

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set -17.57573 1483.037 1110.009 -4.897129 13.37214 0.5562116 0.004823892
```

- Dane II

```
#dane2
wspama_zarobki <- Arima(zarobkis, order =c(0,0,12))
summary(wspama_zarobki)
```

```
Series: zarobkis
ARIMA(0,0,12) with non-zero mean

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8     ma9    ma10    ma11    ma12
      2.0547  3.4672  4.8979  6.1293  6.8413  6.9627  6.3787  5.2904  3.8964  2.4546  1.2365  0.4323
s.e.  0.0370  0.0739  0.1174  0.1597  0.1950  0.2157  0.2184  0.1991  0.1603  0.1127  0.0645  0.0325
          mean
      517.9963
s.e.   17.0692

sigma^2 = 75.8:  log likelihood = -2340.27
AIC=4708.54   AICc=4709.2   BIC=4771.28

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE     MASE      ACF1
Training set 0.1102859 8.619093 6.147855 -0.8357828 1.734022 0.358764 0.1176729
```

11. Wyznaczanie optymalnych modeli z użyciem auto.arima()

```
(ar1 <- auto.arima(domyst3, ic ="aicc"))
```

```
Series: domyst3
ARIMA(2,0,0)(2,0,0)[12] with non-zero mean

Coefficients:
          ar1      ar2     sar1     sar2    mean
      -0.6957  -0.4228  -0.7278  -0.4305  6e-04
s.e.   0.0465   0.0455   0.0468   0.0461  2e-03

sigma^2 = 0.03068:  log likelihood = 126.84
AIC=-241.68   AICc=-241.46   BIC=-217.74
```

```
(ar2 <- auto.arima(domyst3, ic ="aic"))
```

```
> (ar2 <- auto.arima(domyst3, ic ="aic"))
Series: domyst3
ARIMA(2,0,0)(2,0,0)[12] with non-zero mean

Coefficients:
          ar1      ar2     sar1     sar2    mean
      -0.6957  -0.4228  -0.7278  -0.4305  6e-04
s.e.   0.0465   0.0455   0.0468   0.0461  2e-03

sigma^2 = 0.03068:  log likelihood = 126.84
AIC=-241.68   AICc=-241.46   BIC=-217.74
```

```
(ar3 <- auto.arima(domyst3, ic ="bic"))
```

```
> (ar3 <- auto.arima(domyst3, ic ="bic"))
Series: domyst3
ARIMA(3,0,0)(2,0,0)[12] with zero mean

Coefficients:
          ar1      ar2      ar3     sar1     sar2
      -0.7151  -0.4592  -0.0544  -0.7105  -0.4234
s.e.   0.0502   0.0577   0.0532   0.0498   0.0466

sigma^2 = 0.03064:  log likelihood = 127.32
AIC=-242.63   AICc=-242.42   BIC=-218.7
```

```
(ar4 <- auto.arima(zarobst3 , ic ="aicc"))
(ar5 <- auto.arima(zarobst3 , ic ="aic"))
(ar6 <- auto.arima(zarobst3 , ic ="bic"))
```

```
> (ar4 <- auto.arima(zarobst3 , ic ="aicc"))
Series: zarobst3
ARIMA(1,0,0)(1,0,0)[12] with non-zero mean

Coefficients:
          ar1     sar1    mean
      -0.2580  -0.4994  0e+00
s.e.   0.0383   0.0344  2e-04

sigma^2 = 7.863e-05:  log likelihood = 2115.86
AIC=-4223.72   AICc=-4223.66   BIC=-4205.88
```

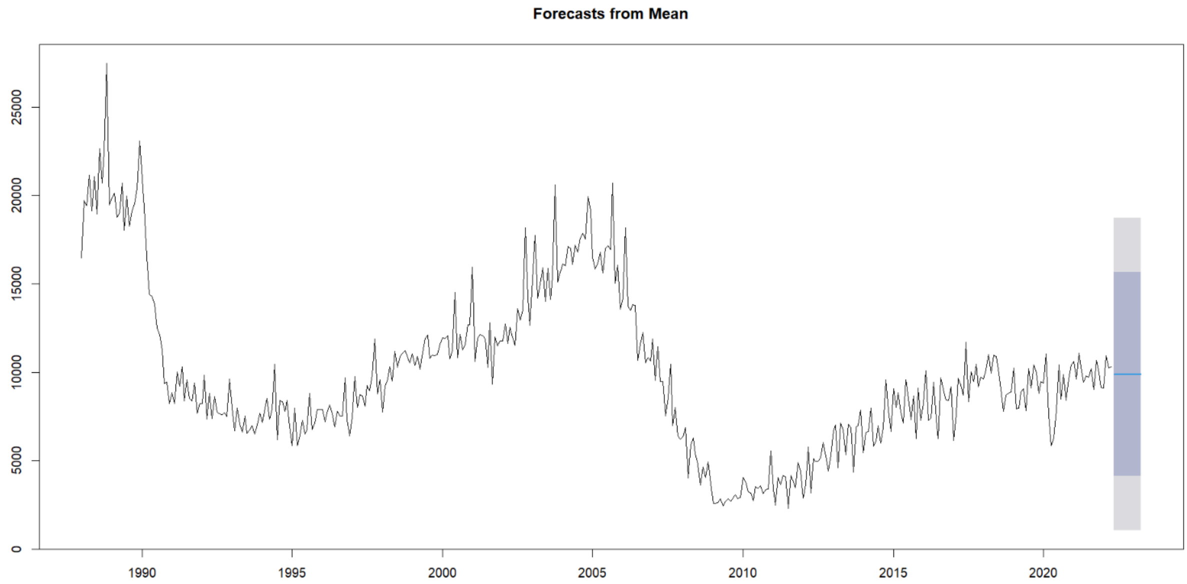
```
> (ar5 <- auto.arima(zarobst3 , ic ="aic"))
Series: zarobst3
ARIMA(1,0,0)(1,0,0)[12] with non-zero mean

Coefficients:
          ar1     sar1    mean
      -0.2580  -0.4994  0e+00
s.e.   0.0383   0.0344  2e-04

sigma^2 = 7.863e-05:  log likelihood = 2115.86
AIC=-4223.72   AICc=-4223.66   BIC=-4205.88
>
```

```
> (ar6 <- auto.arima(zarobst3 , ic ="bic"))
Series: zarobst3
ARIMA(0,0,1)(1,0,0)[12] with zero mean

Coefficients:
          ma1     sar1
      -0.2593  -0.4984
s.e.   0.0375   0.0344

sigma^2 = 7.851e-05:  log likelihood = 2115.88
AIC=-4225.76   AICc=-4225.73   BIC=-4212.38
>
```

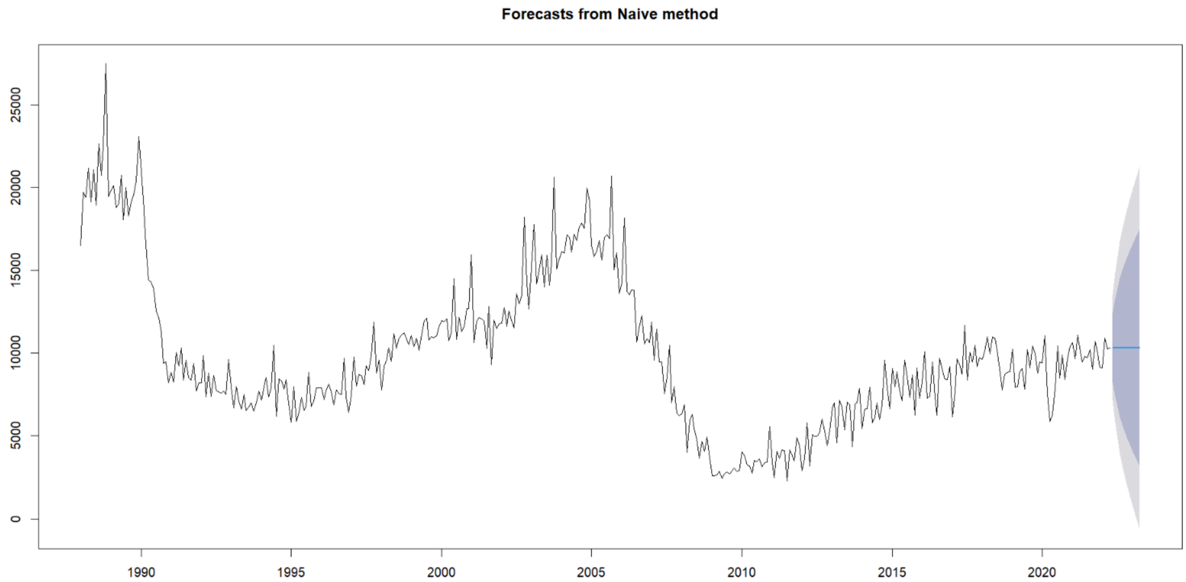
12. Prognozowanie z wykorzystaniem metod naiwnych

Zrobiłem prognozowanie 4 metodami: opartą na średniej, naiwną, s-naiwną, oraz naiwną z uwzględnieniem dryfu

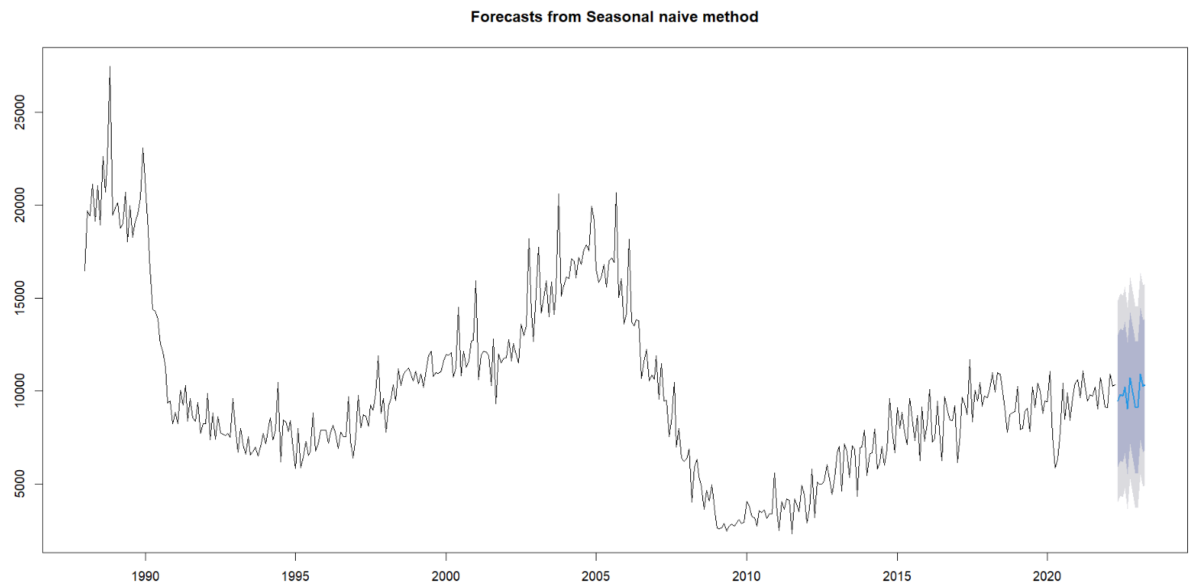
```
#prognozowanie z wykorzystaniem metod naiwnych----
#Na podstawie sredniej dane 1
domysmeanf <- meanf(domys, 12)
plot(domysmeanf)
#metoda naive dane 1
domysnaive <- naive(domys, 12)
plot(domysnaive)
#metoda snaive dane 1
domyssnaive <- snaive(domys,12)
plot(domyssnaive)
#z uzwgl?dnieniem dryfu dane 2
domysdryf <- rwf(domys, 12, drift = TRUE)
plot(domysdryf)
```

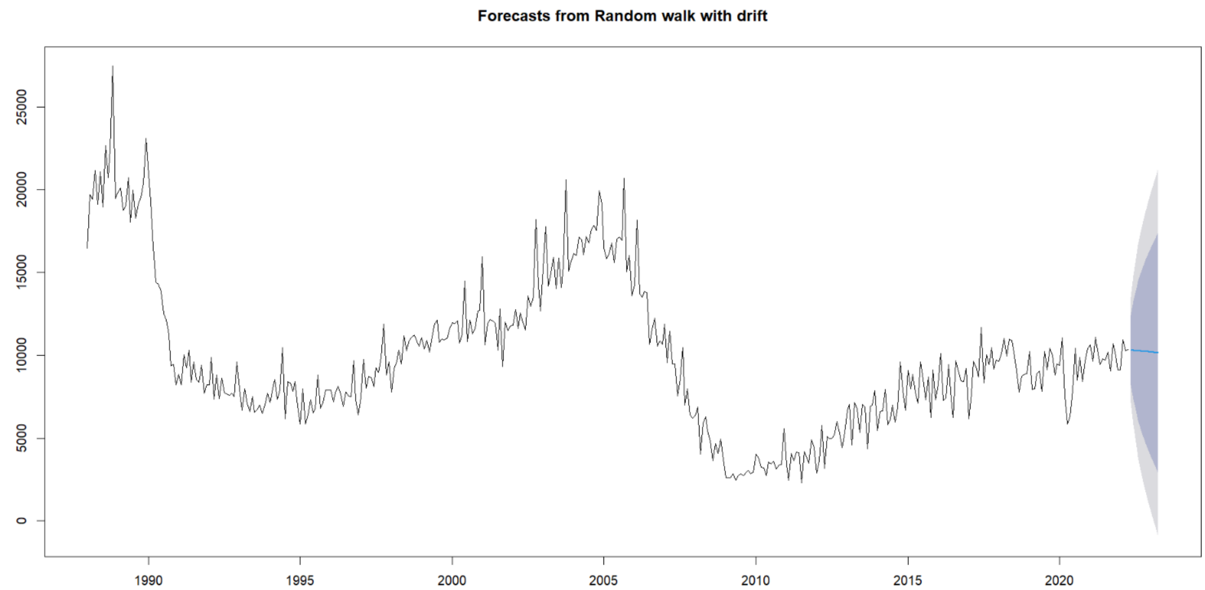


Forecasts from Mean
0
5000
10000
15000
20000
25000
1990
1995
2000
2005
2010
2015
2020



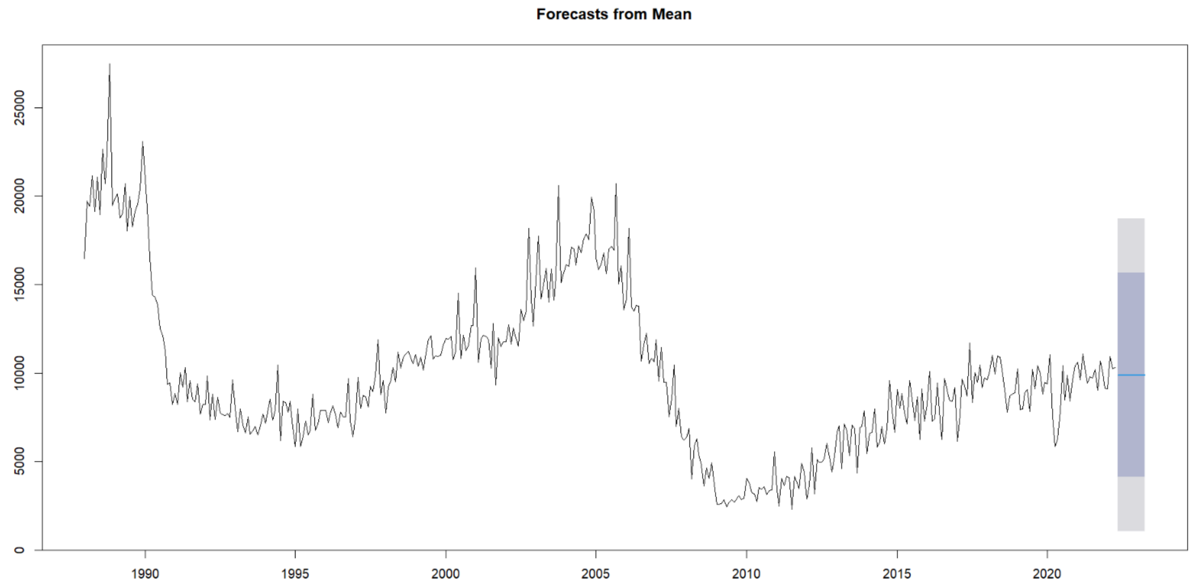
Forecasts from Naive method
0
5000
10000
15000
20000
25000
1990
1995
2000
2005
2010
2015
2020

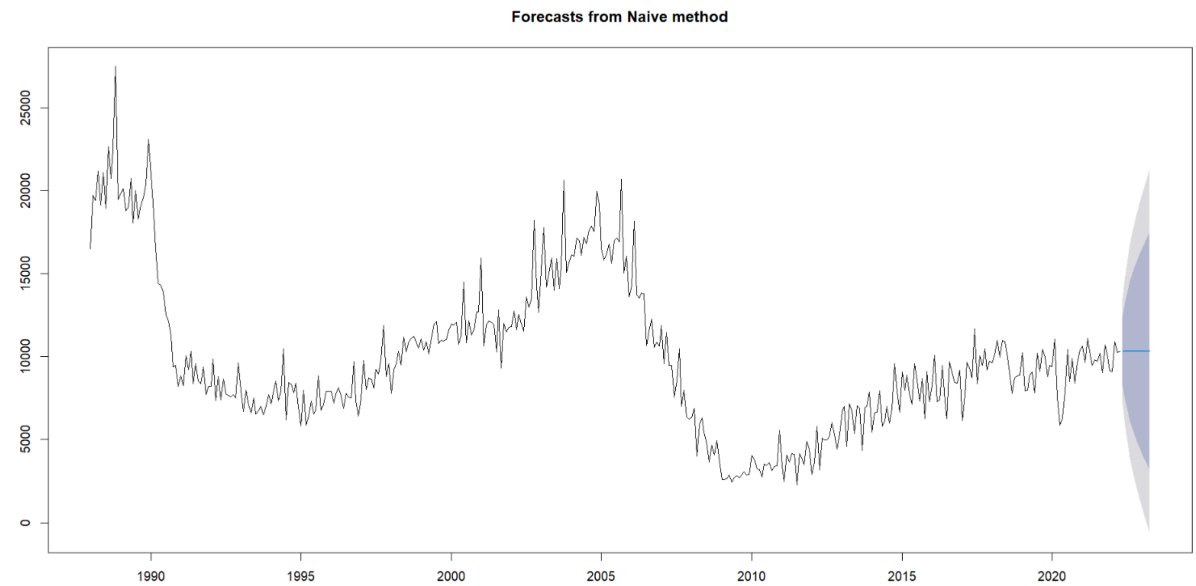

Forecasts from Seasonal naive method
25000
20000
15000
10000
5000
1990
1995
2000
2005
2010
2015
2020

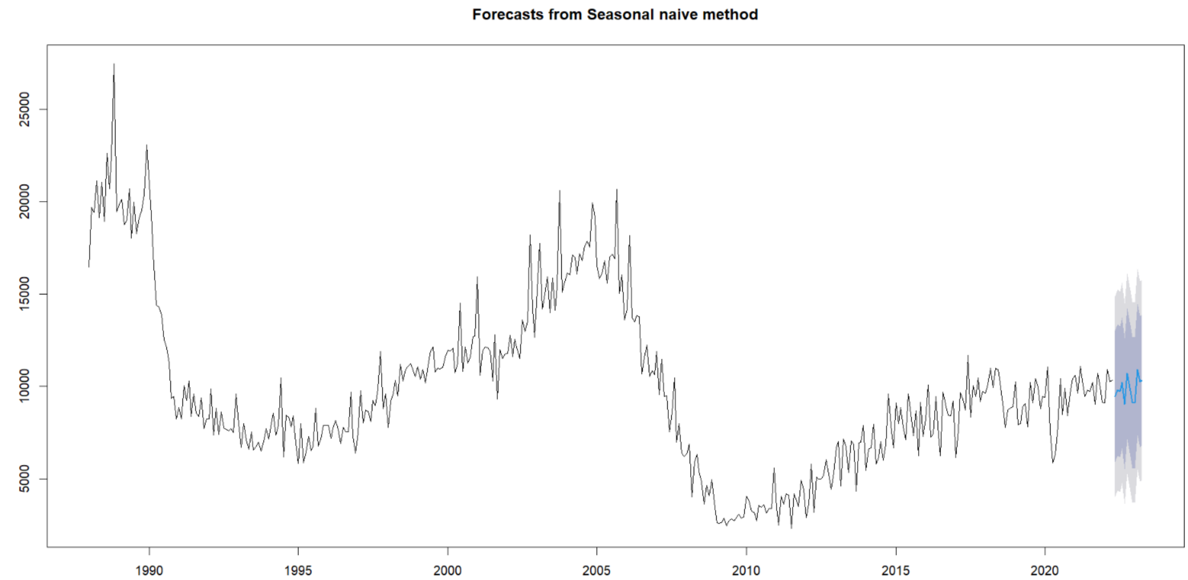

Forecasts from Random walk with drift
25000
20000
15000
10000
5000
0
1990
1995
2000
2005
2010
2015
2020

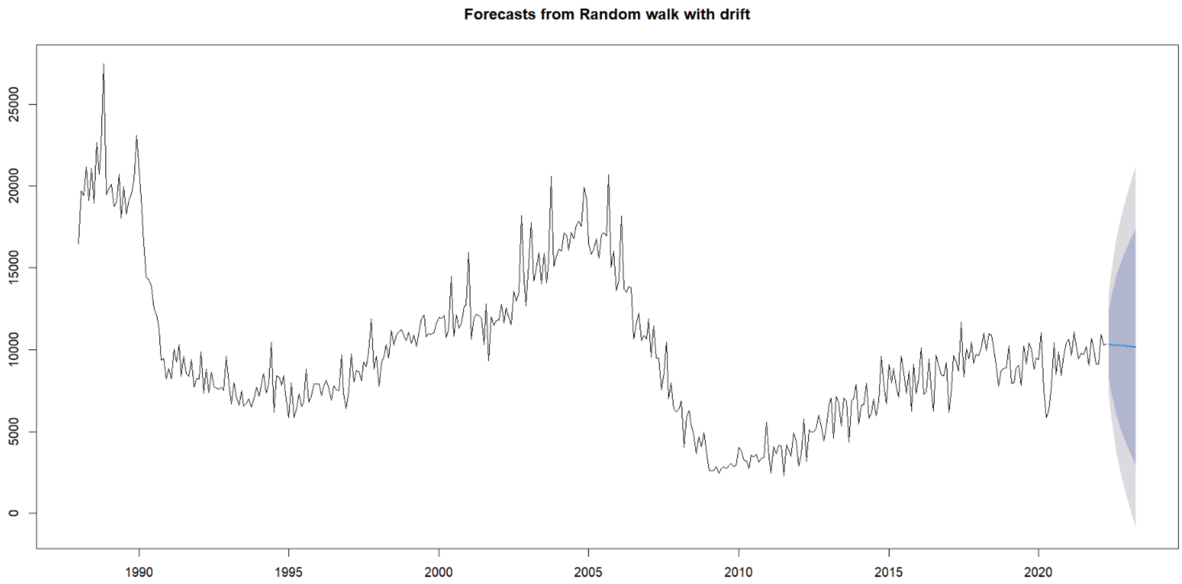
- **Dane II**

```
#Na podstawie sredniej dane 2
zarobkimeanf <- meanf(zarobkis, 12)
plot(domysmeanf)
#metoda naive dane 2
zarobkinaive <- naive(zarobkis, 12)
plot(domysnaive)
#metoda snaive dane 2
zarobkisnaive <- snaive(zarobkis,12)
plot(domyssnaive)
#z uzwgl?dnieniem dryfu dane 2
zarobkidryf <- rwf(zarobkis, 12, drift = TRUE)
plot(domysdryf)
```




Forecasts from Naive method
0
5000
10000
15000
20000
25000
1990
1995
2000
2005
2010
2015
2020


Forecasts from Seasonal naive method
5000
10000
15000
20000
25000
1990
1995
2000
2005
2010
2015
2020



12.1 Wybór najlepszej metody

```r
#Wybor najlepszej metody----
#dane 1
(accuracy(domysmeanf))
(accuracy(domysnaive))
(accuracy(domyssnaive))
(accuracy(domysdryf))

#dane 2
(accuracy(zarobkimeanf))
(accuracy(zarobkinaive))
(accuracy(zarobkisnaive))
(accuracy(zarobkidryf))
```

- Dane I

```
> (accuracy(domysmeanf))
                       ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
Training set 7.342967e-15 4481.865 3403.597 -26.06977 46.72169 1.705499 0.9319739
> (accuracy(domysnaive))
                   ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set -14.92021 1622.867 1200.531 -1.750579 13.66605 0.6015709 -0.4605427
> (accuracy(domyssnaive))
                   ME     RMSE     MAE       MPE     MAPE MASE      ACF1
Training set -325.0626 2767.563 1995.66 -7.092817 24.53309    1 0.6410127
> (accuracy(domysdryf))
                        ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set -1.173244e-13 1622.798 1201.231 -1.560067 13.65994 0.6019219 -0.4605427
> 
```

Najwięcej wartości, które są najmniejsze przypadają dla metody z naiwnej z uwzględnieniem dryfu. Oznacza to, że możemy uznać tą metodę za najlepszą.

- Dane II

```
> #dane 2
> (accuracy(zarobkimeanf))
                        ME    RMSE      MAE       MPE     MAPE     MASE     ACF1
Training set -1.651078e-14 252.569 215.9852 -42.89621 69.79133 12.60402 0.994739
> (accuracy(zarobkinaive))
                   ME     RMSE      MAE       MPE      MAPE      MASE       ACF1
Training set 1.405675 4.348168 2.624969 0.3348529 0.5647019 0.1531826 -0.1975384
> (accuracy(zarobkisnaive))
                  ME     RMSE      MAE      MPE     MAPE MASE      ACF1
Training set 16.6593 20.04475 17.13621 3.905685 3.967628    1 0.8566073
> (accuracy(zarobkidryf))
                       ME     RMSE     MAE         MPE      MAPE      MASE       ACF1
Training set 1.789353e-14 4.114686 2.25892 -0.05231636 0.4812015 0.1318214 -0.1975384
> |
```

Najwięcej wartości, które są najmniejsze przypadają dla metody z naiwnej z uwzględnieniem dryfu. Oznacza to, że możemy uznać tą metodę za najlepszą.