# Szeregi czasowe: Średnia cena biletów lotniczych w USA w latach 1989-2023

Żaneta Sado, Gabriela Ryszka

2024-01-28

## Spis Treści



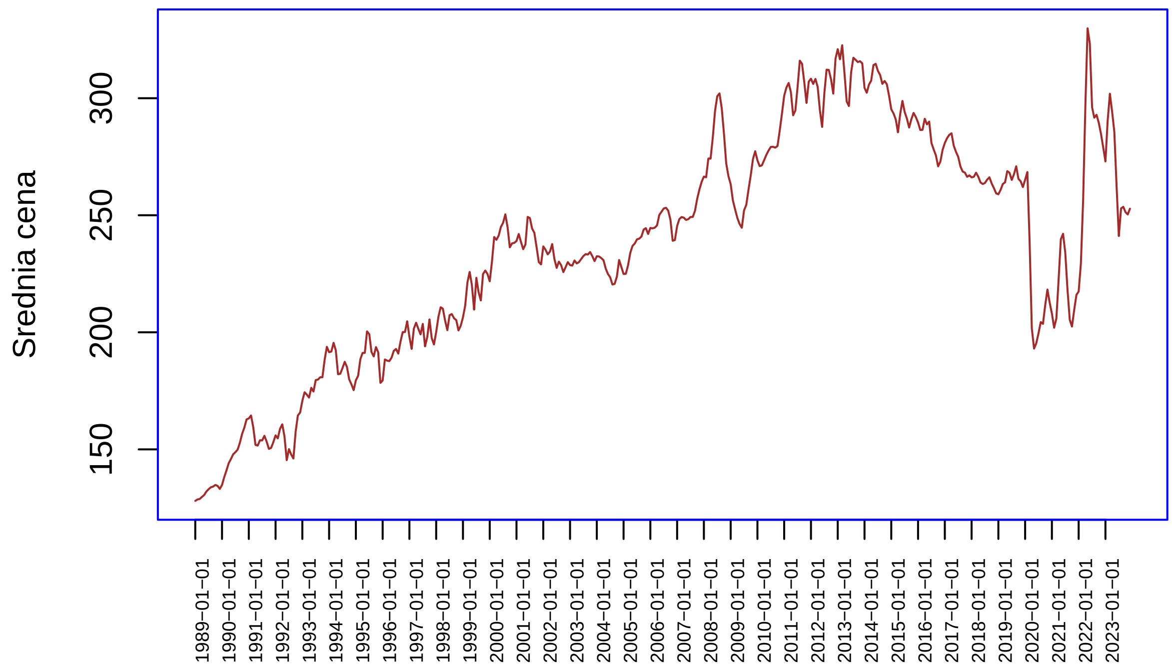
### 0.1 Wstęp

Dane zostały pobrane ze strony: https://fred.stlouisfed.org/series/CUSR0000SETG01?fbclid=IwAR0k8EK9yj0KOWWWEL k. Zakres danych jest pomiędzy 01.07.1989r. a 01.12.2023r. Mamy 420 obserwacji, a nasze dane zawierają 2 zmienne:

- Data-> Data, będąca początkiem każdego miesiąca
- Numbers-> Uśredniona cena biletów

# 1 Wstępna analiza danych

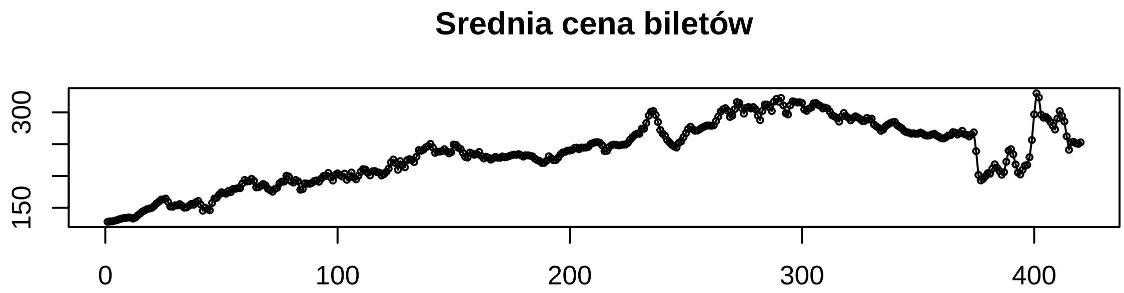
Wczytujemy teraz nasze dane:

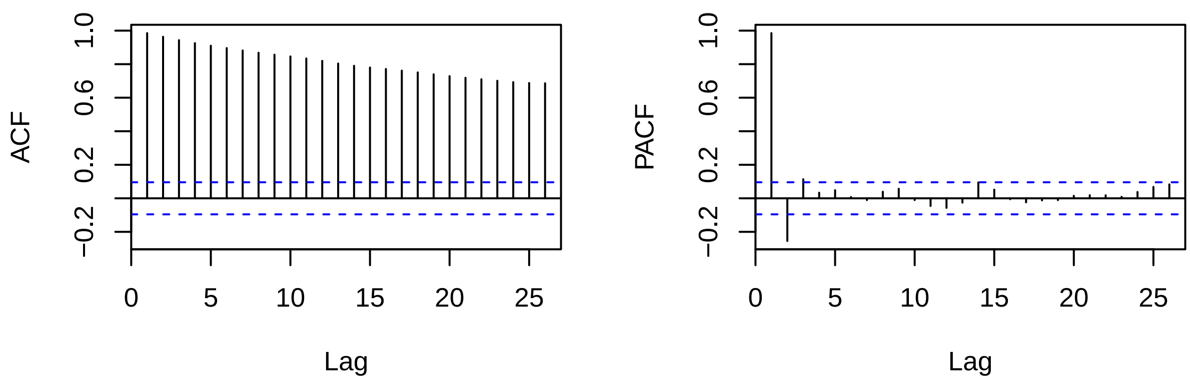



Widzimy, że średnie ceny biletów lotniczych w USA rosły, natomiast znaczny spadek został spowodowany Covidem i obostrzeniami z nim związanymi.

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

```
## Registered S3 methods overwritten by 'forecast':
##   method       from
##   fitted.Arima TSA
##   plot.Arima   TSA
```




Wszystkie słupki w ACF znajdują się poza przedziałem. Odrzucamy zatem hipotezę o tym, że szereg jest realizacją procesu IID. Widać, że ACF powoli wygasa, co jest związane z występowaniem trendu liniowego.

# 2 Badanie stacjonarności

## 2.1 Bazowy szereg czasowy

Hipoteza zerowa w teście Augmented Dickey-Fuller: nie wiemy, czy proces jest stacjonarny, natomiast H1 - jest stacjonarny. Przez to, że p-value wyszło większe - nie możemy odrzucić hipotezy zerowej.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Dane_b
## Dickey-Fuller = -2.0612, Lag order = 7, p-value = 0.5516
## alternative hypothesis: stationary
```

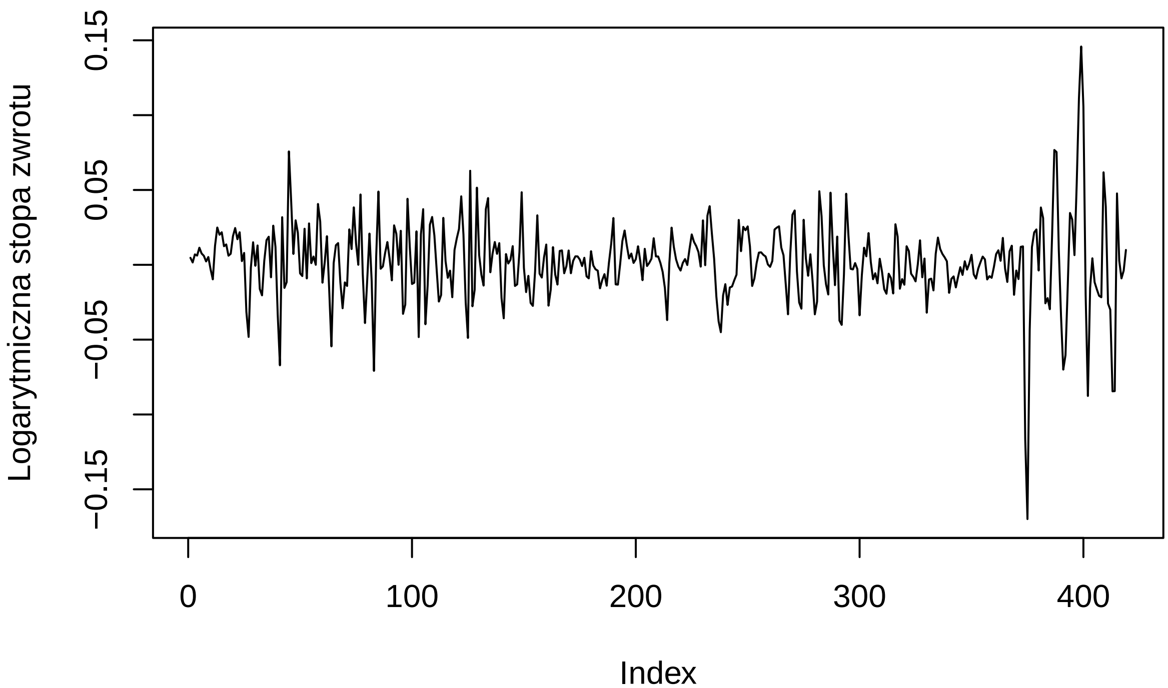
Natomiast test KPSS, gdzie hipoteza zerowa mówi o stacjonarności - zwrócił p-value = 0.01. Jest ona poniżej 0.05, dlatego hipoteza zerowa mówiąca o stacjonarności zostaje odrzucona.

```
##
##  KPSS Test for Level Stationarity
##
## data:  Dane_b
## KPSS Level = 4.9753, Truncation lag parameter = 5, p-value = 0.01
```

Odrzuciliśmy zatem stacjonarność naszego pierwotnego szeregu.

## 2.2 Stacjonarność logarytmicznych stóp zwrotu.

Ponieważ pierwotny szereg nie jest stacjonarny, rozważmy logarytmiczne stopy zwrotu:




Przeprowadźmy dla nich testy ADF oraz KPSS.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  lnreturns
## Dickey-Fuller = -9.1518, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

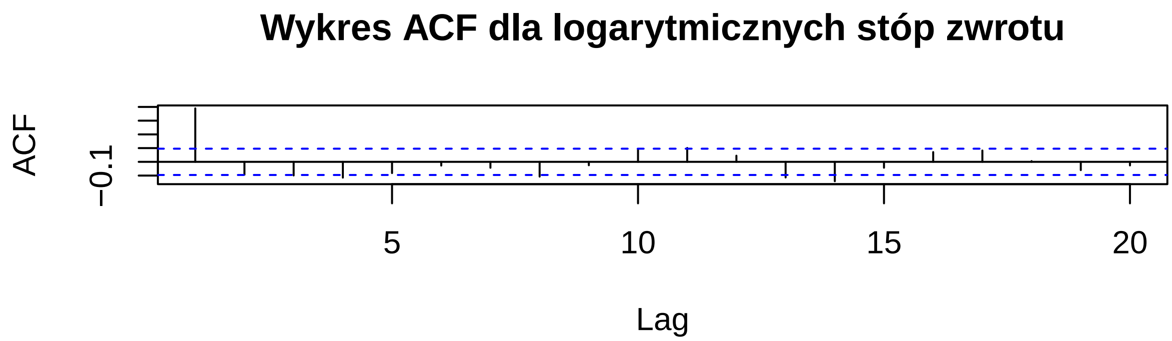
Test ADF odrzuca nam hipotezę zerową na rzecz alternatywnej - że szereg jest stacjonarny.

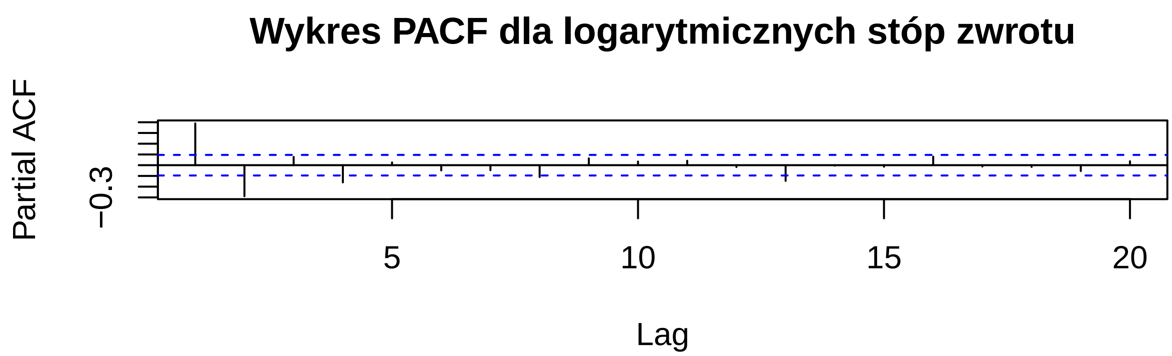
```
##
##  KPSS Test for Level Stationarity
##
## data:  lnreturns
## KPSS Level = 0.20451, Truncation lag parameter = 5, p-value = 0.1
```

Zaś z KPSS nie możemy odrzucić hipotezy zerowej, która mówi że szereg jest stacjonarny.

# 3 Proces ARMA(p,q)

Ponieważ ARMA zakłada, że dane są stacjonarne, dlatego przyjrzymy się ponownie logarytmicznym stopom zwrotu, a zwłaszcza ich wykresom ACF i PACF:




Uwzględniając, że istotne są w tym przypadku słupki do opóźnienia (log(n)), czyli do 6, kandydatami procesu ARMA są: ARMA(1,0), ARMA(0,1), ARMA(1,1), ARMA(1, 2) oraz ARMA(0, 2). Rozważając każdy z tych modeli:

```
##
## Call:
## stats::arima(x = lnreturns, order = c(1, 0, 1))
##
## Coefficients:
##           ar1     ma1  intercept
##       -0.1032  0.6623     0.0017
## s.e.   0.0795  0.0596     0.0017
##
## sigma^2 estimated as 0.0005509:  log likelihood = 977.34,  aic = -1946.67
```

```
##
## Call:
## TSA::arima(x = lnreturns, order = c(1, 0, 1))
##
## Coefficients:
##           ar1     ma1  intercept
##       -0.1032  0.6623     0.0017
## s.e.   0.0795  0.0596     0.0017
##
## sigma^2 estimated as 0.0005509:  log likelihood = 977.34,  aic = -1948.67
```

```
##
## Call:
```

```
## stats::arima(x = lnreturns, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.3887     0.0016
## s.e.  0.0449     0.0020
##
## sigma^2 estimated as 0.0006152:  log likelihood = 954.34,  aic = -1902.68

##
## Call:
## stats::arima(x = lnreturns, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.5960     0.0017
## s.e.  0.0421     0.0018
##
## sigma^2 estimated as 0.000553:  log likelihood = 976.51,  aic = -1947.02

##
## Call:
## stats::arima(x = lnreturns, order = c(1, 0, 2))
##
## Coefficients:
##          ar1      ma1      ma2  intercept
##       0.8235  -0.2850  -0.5701     0.0016
## s.e.  0.0761   0.0739   0.0456     0.0009
##
## sigma^2 estimated as 0.0005408:  log likelihood = 981.13,  aic = -1952.26

##
## Call:
## stats::arima(x = lnreturns, order = c(0, 0, 2))
##
## Coefficients:
##          ma1      ma2  intercept
##       0.5531  -0.0707     0.0017
## s.e.  0.0504   0.0516     0.0017
##
## sigma^2 estimated as 0.0005506:  log likelihood = 977.45,  aic = -1946.89

## [1] 4 1 3 5 2
```

Powyższe wyniki wskazują na to, że względem kryterium AIC model ARMA(1,2) jest najlepszy.

Przejdźmy teraz do testu Ljungi-Boxa tego modelu:

```
##
##  Box-Ljung test
##
## data:  lnreturns
## X-squared = 117.93, df = 20, p-value = 0.0000000000000006661
```

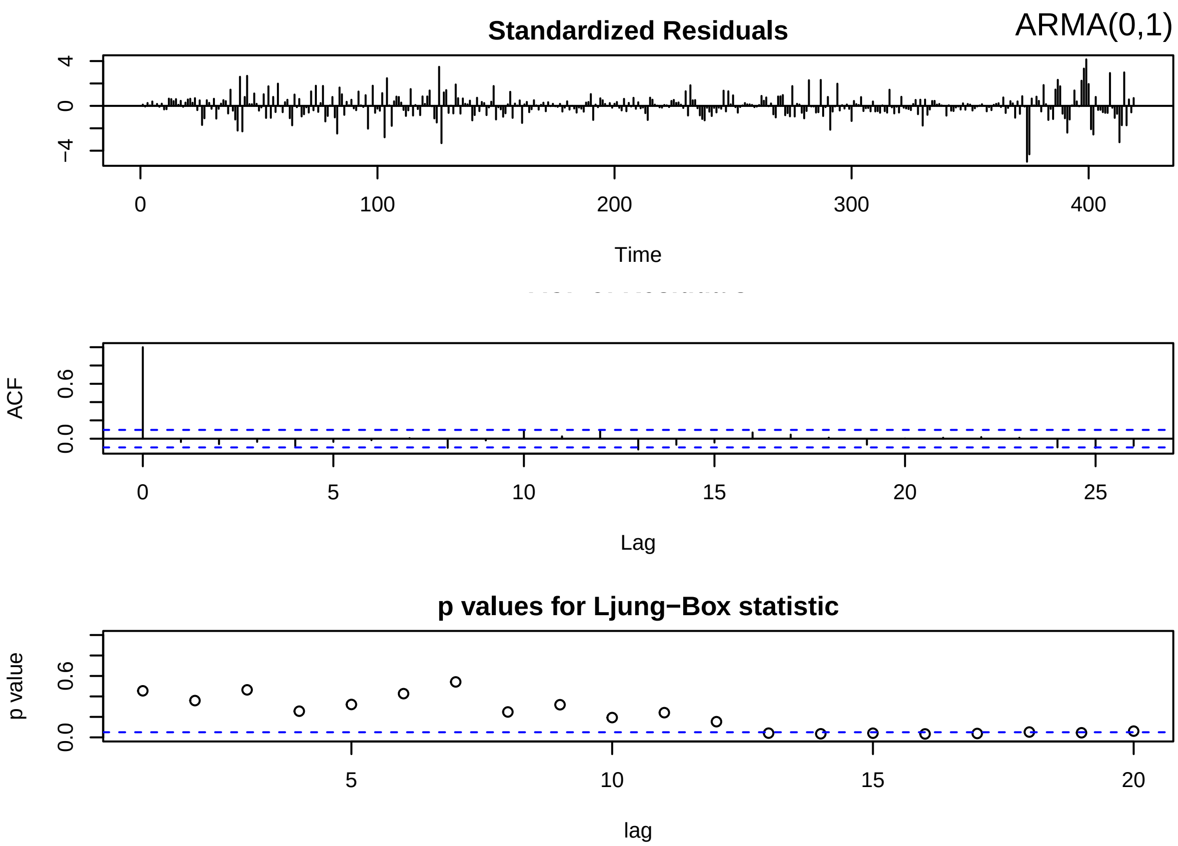
Tutaj p-value jest zdecydowanie mniejsza niż 0.05, więc mamy podstawę do odrzucenia hipotezy zerowej. Nie możemy przyjąć, że błędy są niezależne.

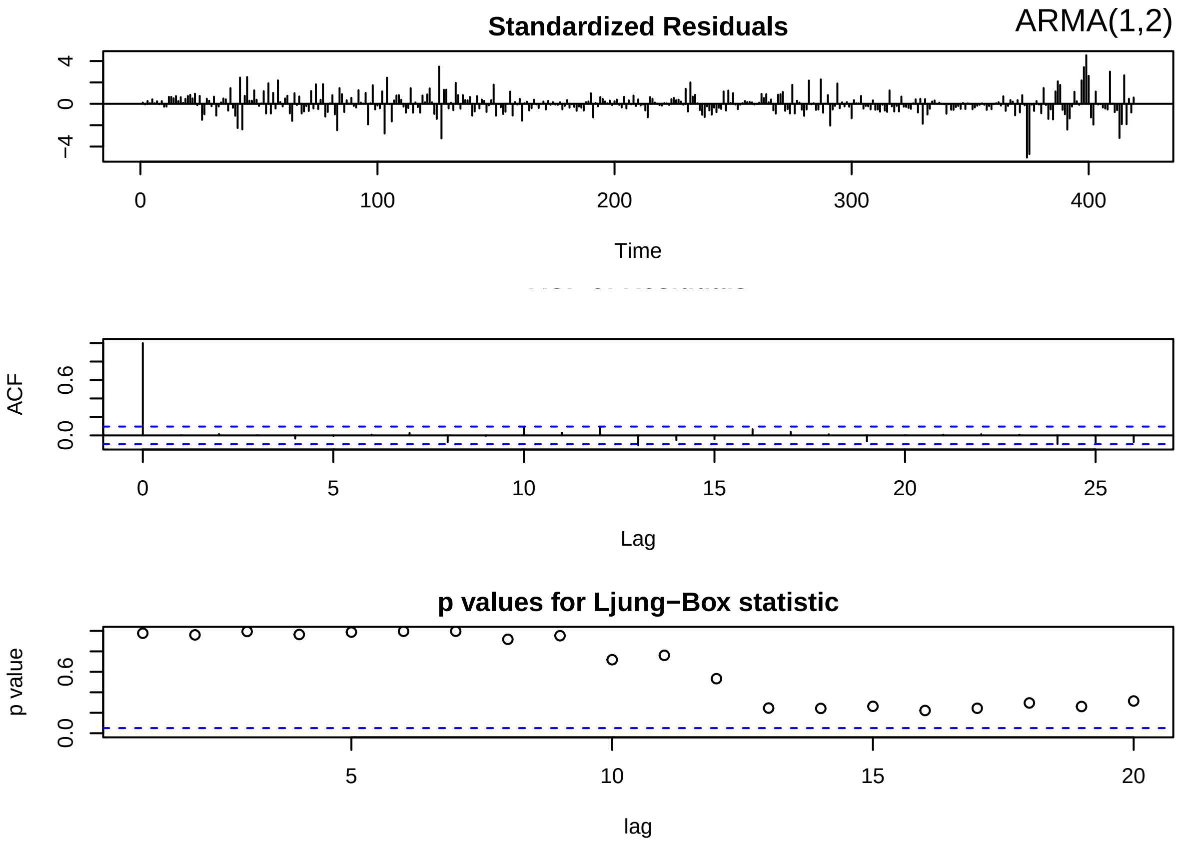
Sprawdzamy teraz który model będzie najlepszy z kryterium AIC, przy metodzie największej wiarygodności:

```r
dane.fitML1<-arima(lnreturns, order = c(1, 0, 1),method ="ML")
dane.fitML2<-arima(lnreturns, order = c(1, 0, 0),method ="ML")
dane.fitML3<-arima(lnreturns, order = c(0, 0, 1),method ="ML")
dane.fitML4<-arima(lnreturns, order = c(1, 0, 2),method ="ML")
dane.fitML5<-arima(lnreturns, order = c(0, 0, 2),method ="ML")
akaike2 <- c(
  dane.fitML1$aic,
  dane.fitML2$aic,
  dane.fitML3$aic,
  dane.fitML4$aic,
  dane.fitML5$aic
)
order(akaike2)
```

```
## [1] 4 3 5 1 2
```

Najlepszy okazał się tutaj również model ARMA(1,2).





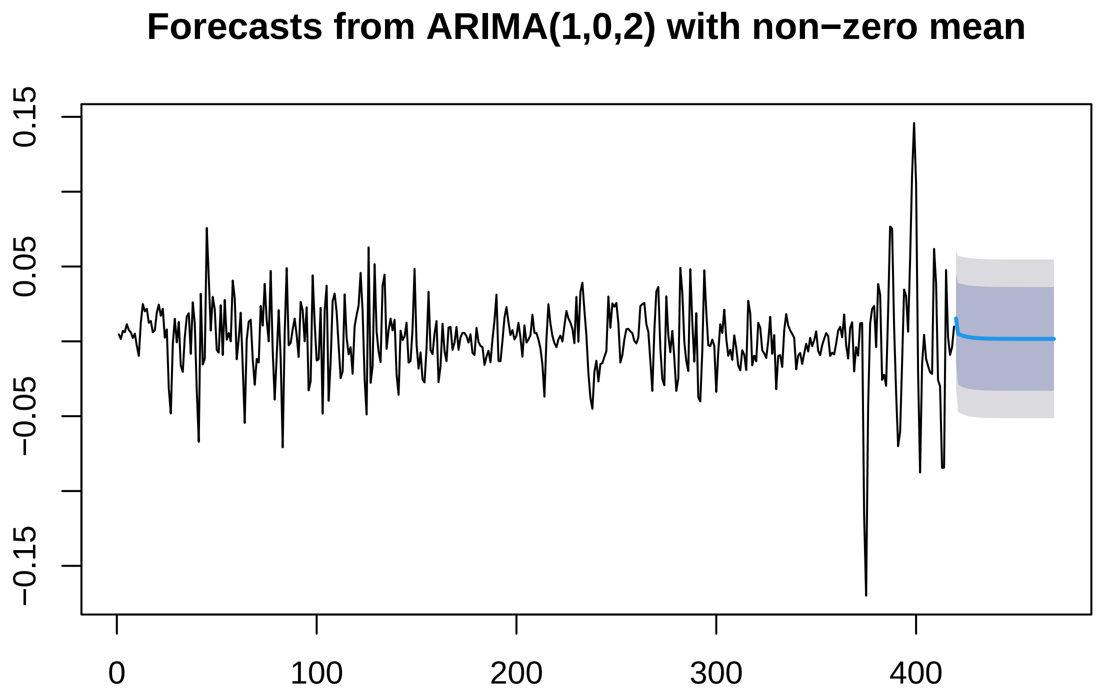

Model ARMA(1,2) możemy uznać za dobry, ponieważ jego reszty znajdują się powyżej linii przerywanej, czyli powyżej 5%. Nie ma zatem podstaw, by odrzucić hipotezę zerową, że autokorelacja opóźnień odpowiednio 1,2...,20 jest równa 0.

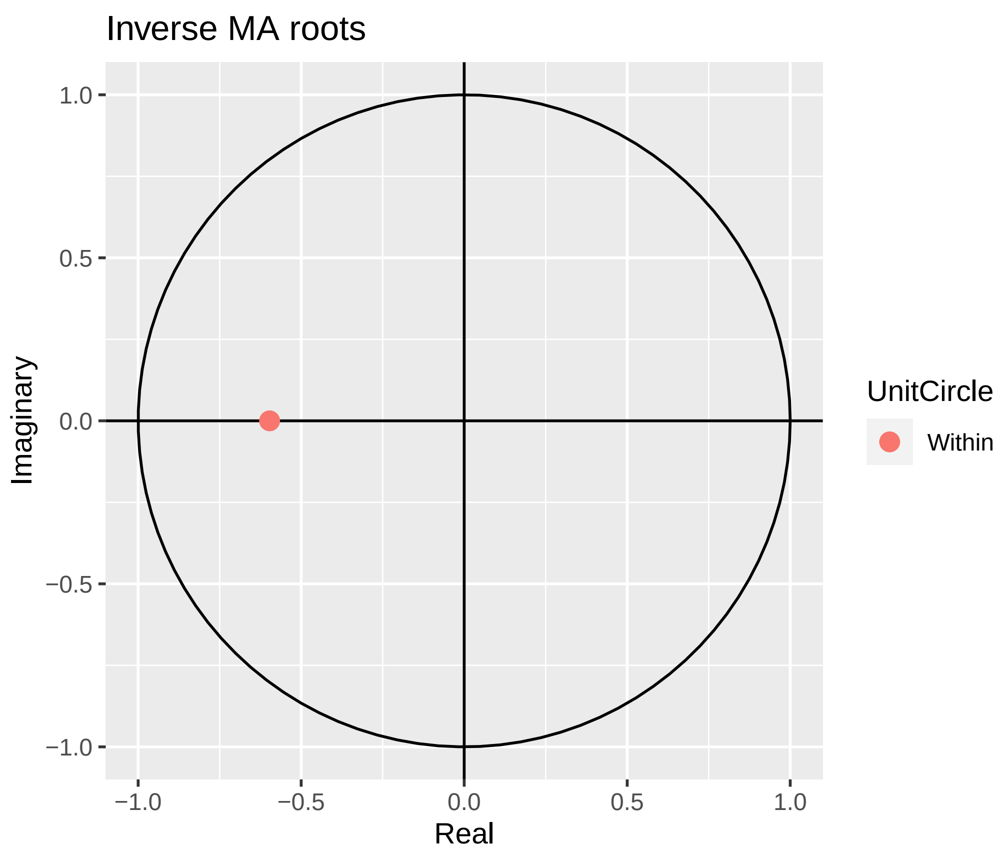
Przechodzimy teraz do automatycznego dopasowania modelu do szeregu czasowego:

```
## Series: lnreturns
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##          ma1
##       0.5968
## s.e.  0.0420
##
## sigma^2 = 0.0005554:  log likelihood = 976.1
## AIC=-1948.21   AICc=-1948.18   BIC=-1940.13
```

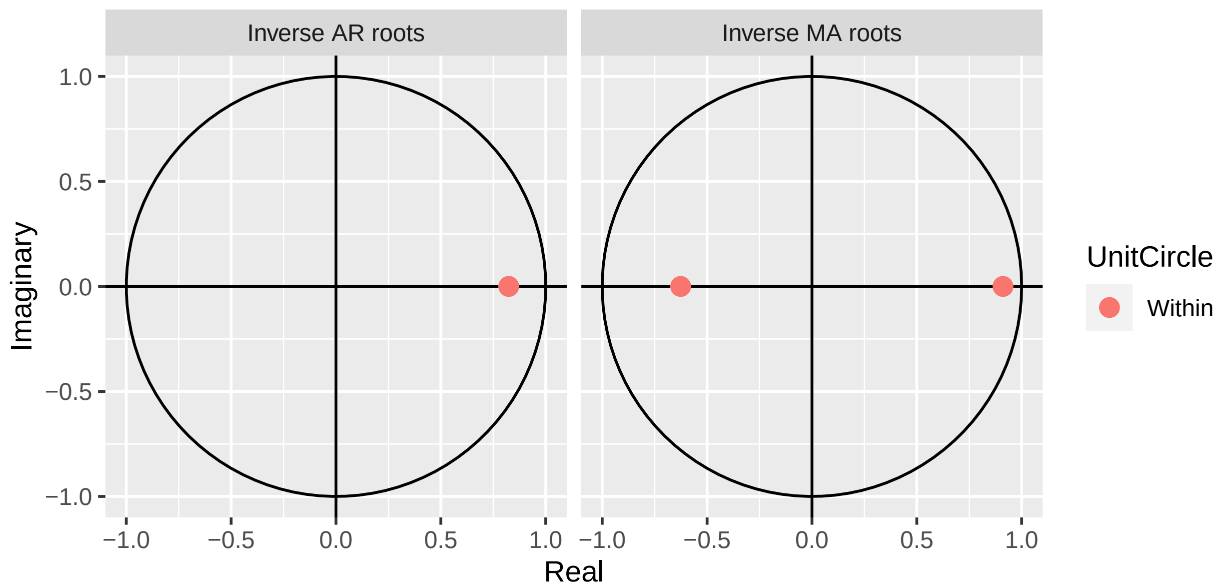
```
## Series: lnreturns
## ARIMA(1,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ma1      ma2    mean
##       0.8235  -0.2850  -0.5701  0.0016
## s.e.  0.0761   0.0739   0.0456  0.0009
##
## sigma^2 = 0.000546:  log likelihood = 981.13
## AIC=-1952.26   AICc=-1952.12   BIC=-1932.07
```

Względem kryterium AIC model ARMA(1,2) jest lepszy. Prognoza dla tego modelu wygląda następująco:





Inverse MA roots
Imaginary
Real
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
UnitCircle
Within




Model ARMA(0,1) ma pierwiastek rzeczywisty, którego odwrotność jest wewnątrz okręgu. Również w przypadku modelu ARMA(1,2), zarówno dla AR(1) jak i MA(2) pierwiastki są rzeczywiste, a ich odwrotność znajdują się w środku. Oznacza to, że procesy, które zostały wyestymowane, są procesami odwracalnymi.

Rozważmy teraz reszty obu modeli: ARMA(0,1) i ARMA(1,2)

```
##
##  Box-Ljung test
##
## data:  reszty1
## X-squared = 5.9552, df = 6, p-value = 0.4282
```

```
##
##  Box-Ljung test
##
## data:  reszty2
## X-squared = 0.62945, df = 6, p-value = 0.9959
```

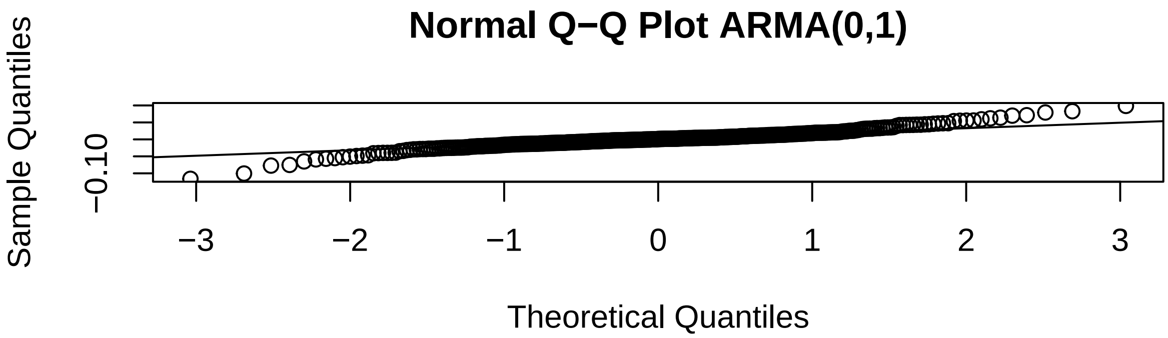
W teście Ljungi-Boxa nie mamy podstaw do odrzucenia hipotezy zerowej. Może być tak, że błędy obu modeli są niezależne. Spójrzmy na test Shapiro-Wilka:

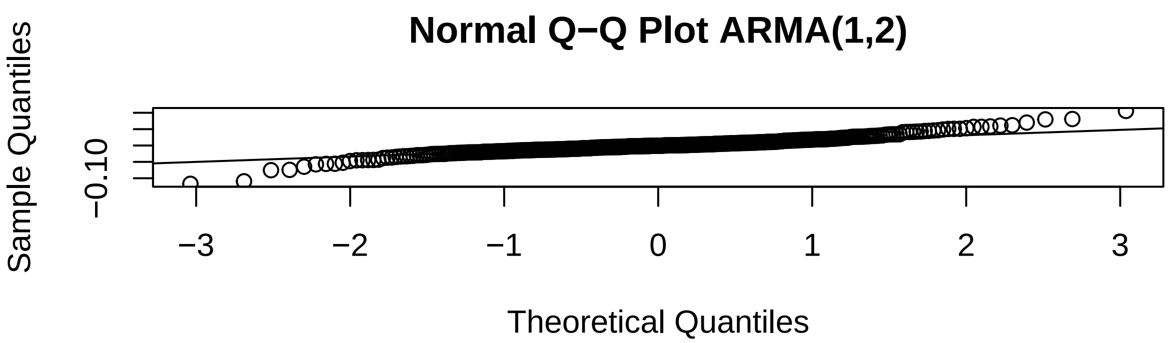
```
##
##  Shapiro-Wilk normality test
##
## data:  reszty1
## W = 0.94072, p-value = 0.000000000007016
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  reszty2
## W = 0.93599, p-value = 0.000000000001952
```

Z tego wynika, że przez p-value poniżej 0.05 nie można odrzucić hipotezy zerowej o normalności próbki danych w obu modelach.




Zatem podsumowując wszystkie kryteria lepszym modelem mimo wszystko okazał się nasz ARMA(1,2).

Ma on stacjonarne pierwiastki, jest kauzalny i jest procesem odwracalnym.

# 4 Prognozowanie ARIMA

Rozważmy teraz proces logarytmicznych stóp zwrotu z jednokrotnym różnicowaniem. Naszych obserwacji jest 420, do prognozowania wybieramy 400 obserwacji, a pozostałych nie znamy. Dopasowanie modelu ARIMA do naszych danych:

```
##
## Call:
## arima(x = dane1[insample], order = c(1, 1, 2), method = "ML")
##
## Coefficients:
##           ar1      ma1      ma2
##       -0.0762  -0.3539  -0.6461
## s.e.   0.0875   0.0686   0.0683
##
## sigma^2 estimated as 0.0005106:  log likelihood = 943.25,  aic = -1880.49
```

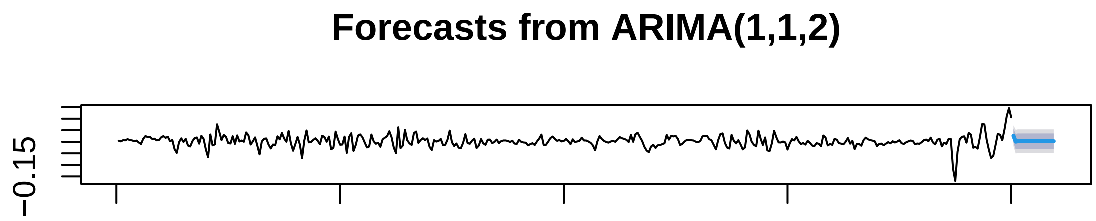
Spójrzmy teraz na prognozę punktową dla 20 obserwacji licząc od 401:

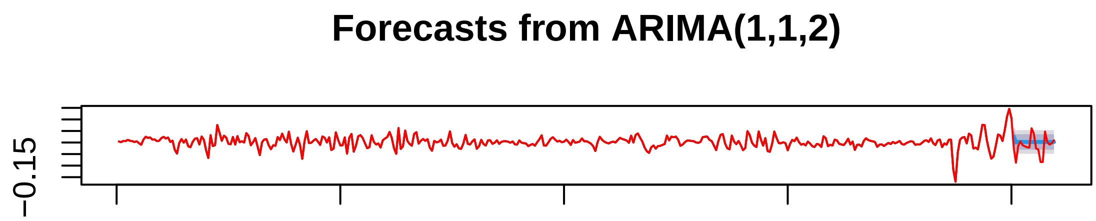
```
## $pred
## Time Series:
## Start = 401
## End = 419
## Frequency = 1
##  [1] 0.0263591979 0.0006040729 0.0025678616 0.0024181257 0.0024295428
##  [6] 0.0024286723 0.0024287387 0.0024287336 0.0024287340 0.0024287340
## [11] 0.0024287340 0.0024287340 0.0024287340 0.0024287340 0.0024287340
## [16] 0.0024287340 0.0024287340 0.0024287340 0.0024287340
##
## $se
## Time Series:
## Start = 401
## End = 419
## Frequency = 1
##  [1] 0.02262448 0.02606810 0.02608330 0.02608366 0.02608364 0.02608364
##  [7] 0.02608364 0.02608364 0.02608364 0.02608364 0.02608364 0.02608364
## [13] 0.02608364 0.02608364 0.02608364 0.02608364 0.02608364 0.02608364
## [19] 0.02608364
```

Szybko osiągana jest zbieżność, dlatego dalsze prognozy są na tym samym poziomie. Miara dokładności tej prognozy jest następująca:

```
##                   ME      RMSE        MAE      MPE     MAPE
## Test set -0.01759467 0.0435398 0.03347185 107.2401 107.2401
```

Rozważmy ten model, który ocenia jakość prognoz, porównując prognozowane wartości z rzeczywistymi wartościami w próbie testowej.

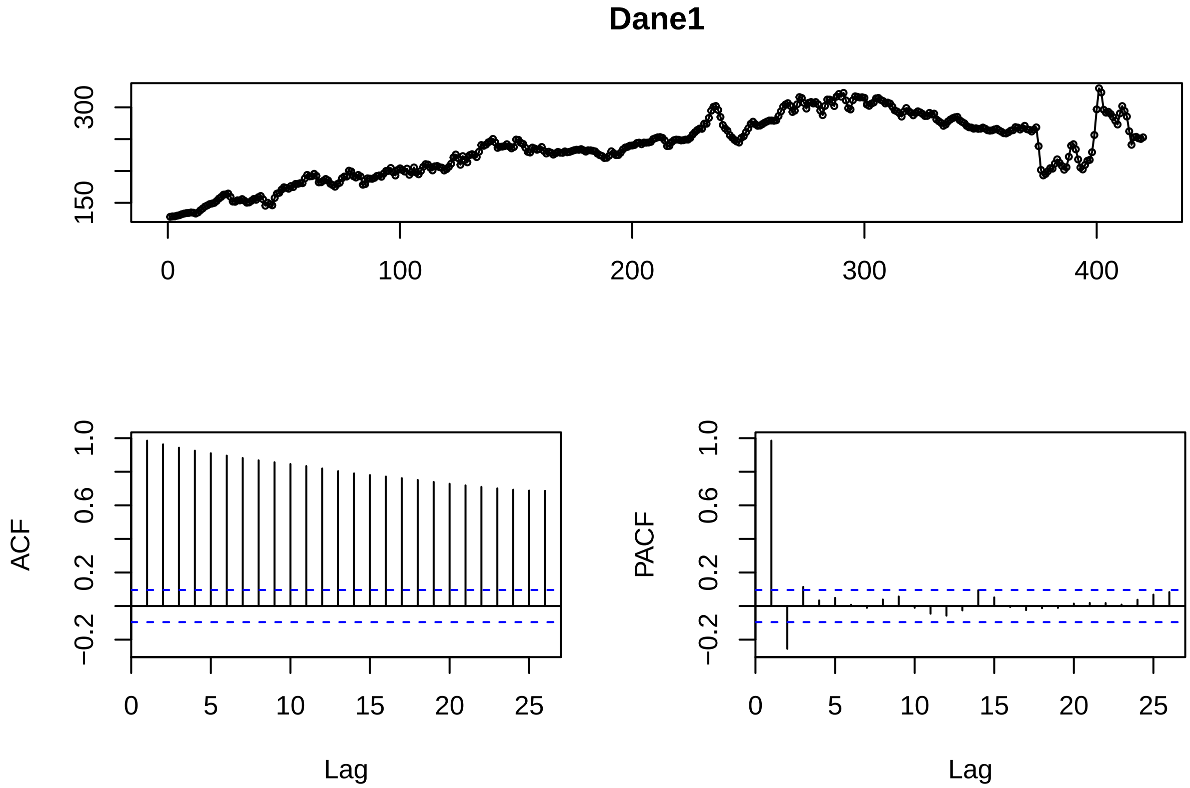


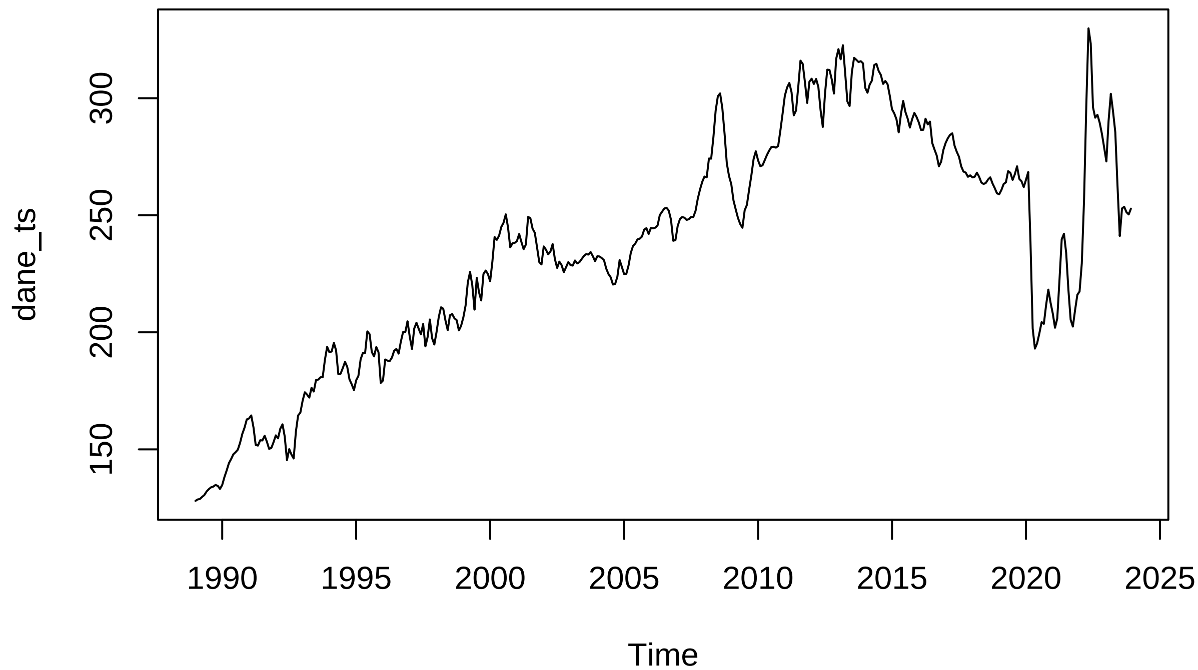

Czarny wykres to wykres dostępnych danych z pewną prognozą, natomiast niżej znajduje się ten sam wykres z tą różnicą, że została nałożona czerwona linia prezentująca dane rzeczywiste.

## 5 SARIMA(p,d,q)(P,D,Q)[m]

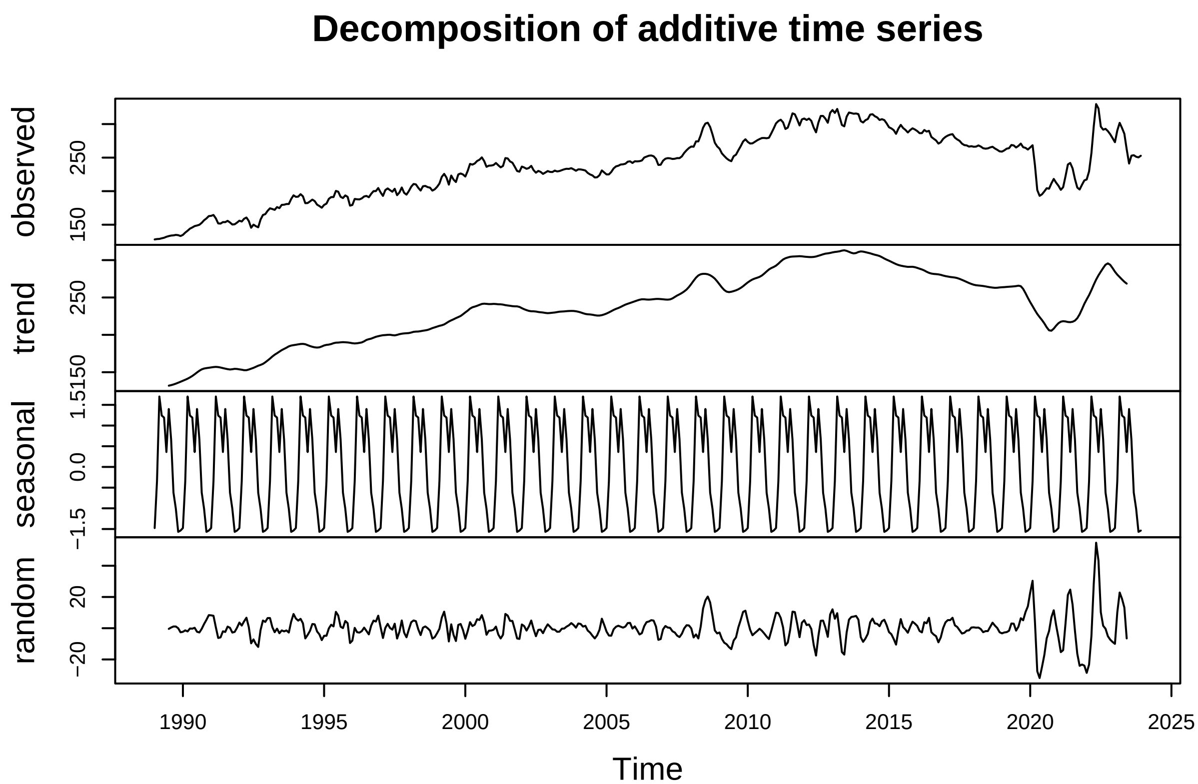
Wracając do naszych danych pierwotnych, czyli średnich wartości cen biletów, przypomnijmy:




Widać, że ACF powoli wygasa, co jest związane z występowaniem trendu liniowego. Natomiast wysoka wartość PACF dla pierwszego opóźnienia oznacza, że AR(1) będzie dobrym kandydatem. Nie pozbywamy się żadnych danych, ponieważ mamy pełne lata. Skoro dane mamy miesięczne, więc częstotliwość jest ustawiona na 12.




Spójrzmy teraz na dekompozycję:

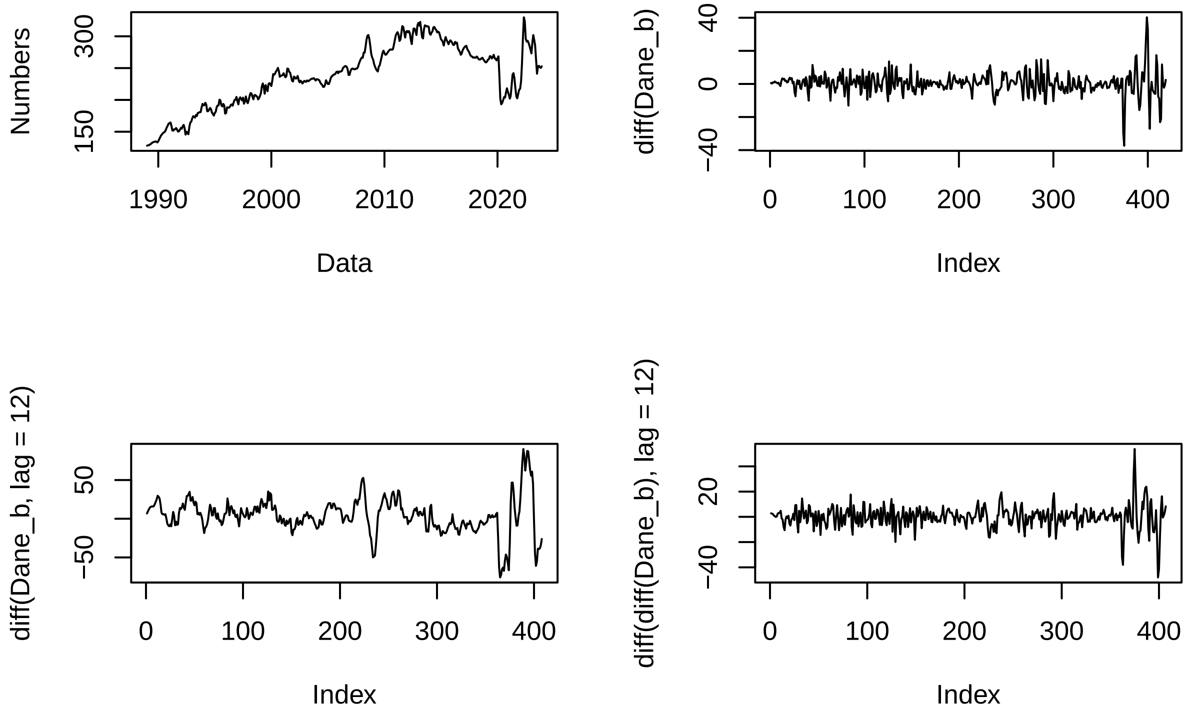



Na wykresie trendu widać że był jeden znaczny spadek w średniej cenie biletów, a w pozostałych przypadkach były umiarkowane wzrosty połączone ze spadkami, mamy do czynienia z trendem rosnącym, możemy przypuszczać że sezonowość w naszych danych spowodowana jest tym, że ludzie częściej podróżują w wakacje albo w okresie świątecznym. Ponieważ w danych występuje sezonowość, wprowadźmy nowy model SARIMA(p,d,q)(P,D,Q)[m], gdzie: m=12, d=1, D=0, gdyż:

```
## [1] 0
```

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

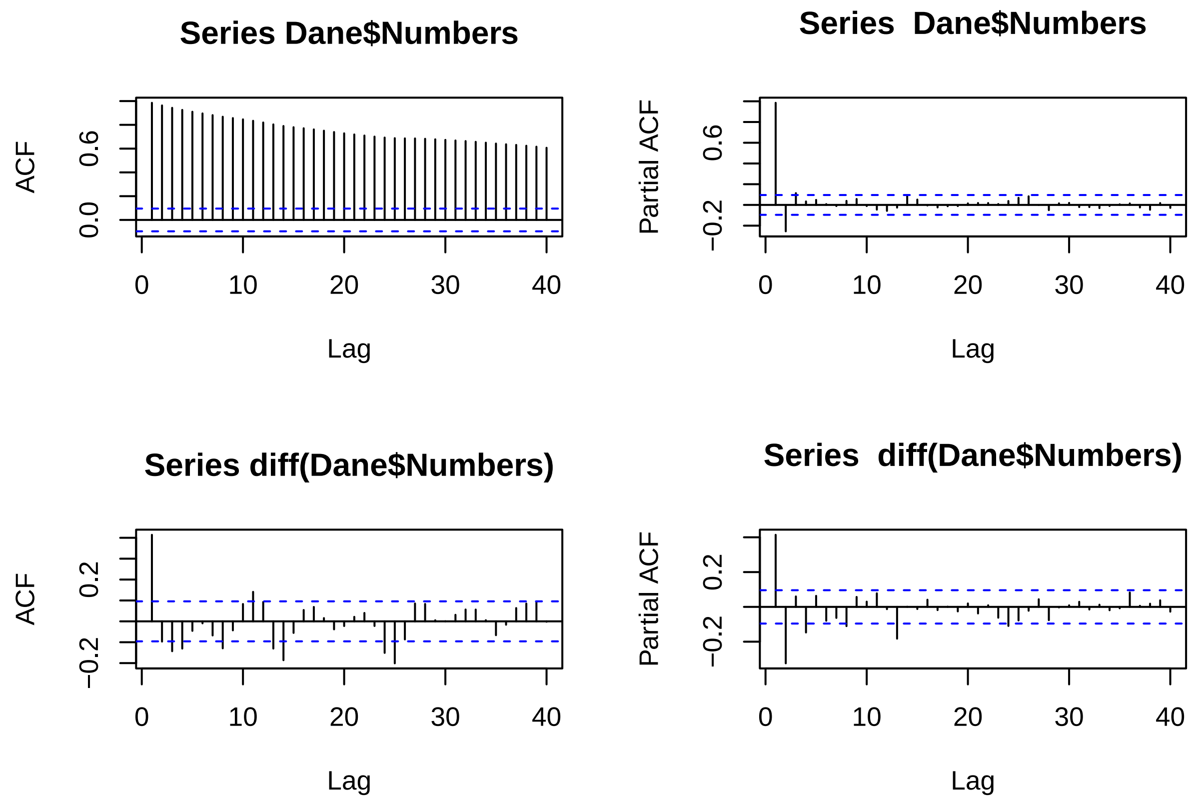
Przyjrzyjmy się teraz szeregom:



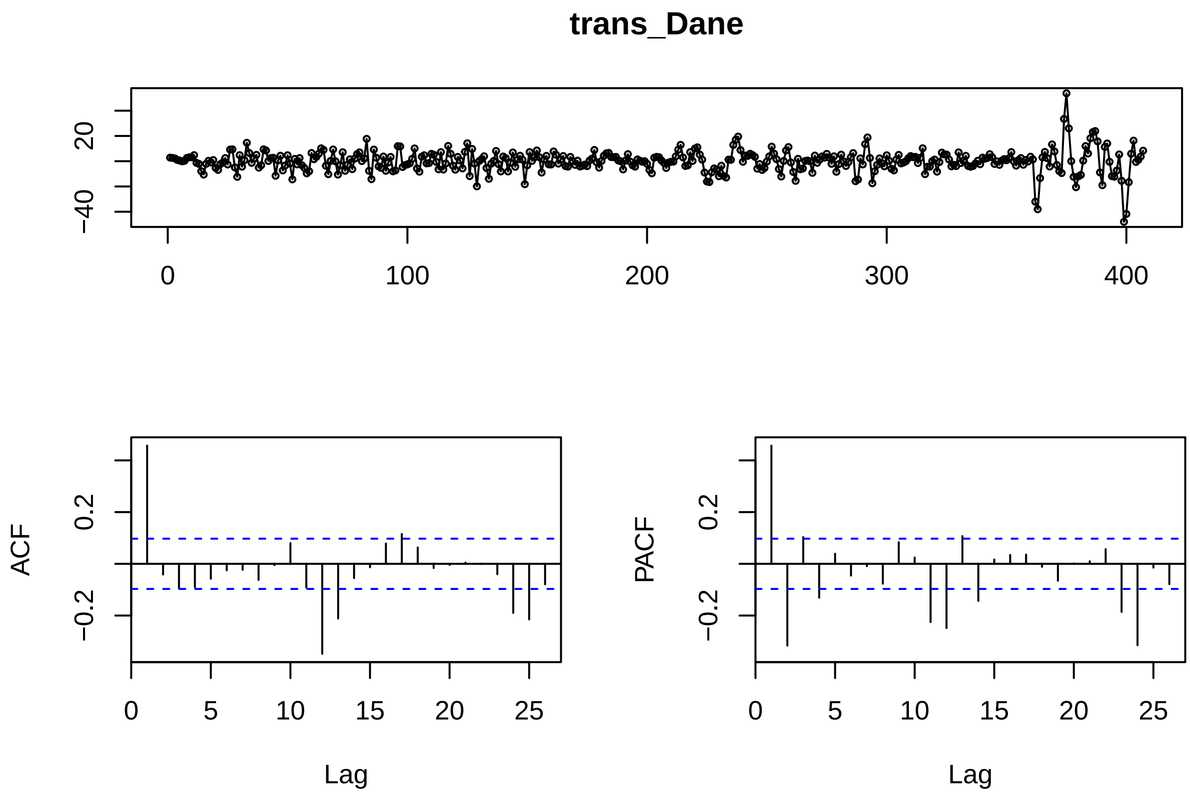

Pierwszy wykres przedstawia wyjściowy szereg. Wykres po jego prawej jest wykresem powstałym przez jednokrotne różnicowanie, co usunęło nam trend. Wykres na dole po lewej jest wykresem powstałym przez 12-krotne różnicowanie. Wykres po jego prawej stronie powstał przez 12-krotne różnicowanie, a następnie jednokrotne. Ten zabieg usunął nam sezonowość.

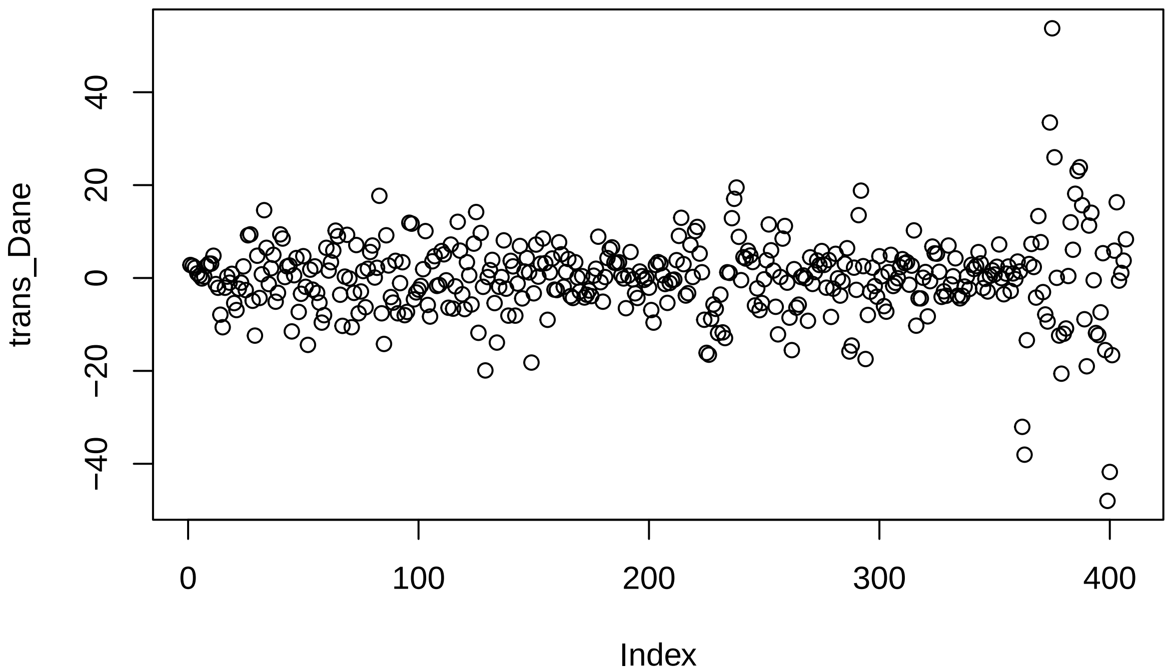
```
## [1] 20.4939
```

Bierzemy pod uwagę opóźnienia do 20.



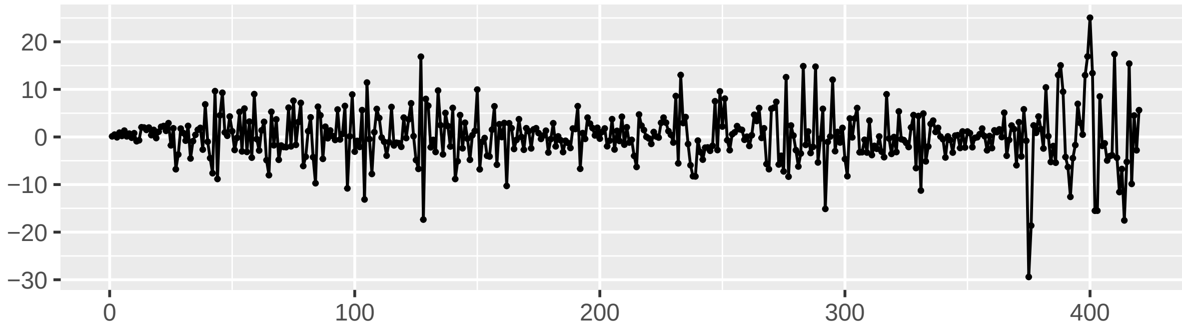

Po jednokrotnym różnicowaniu widać w ACF sezonowość. Bierzemy pod uwagę opóźnienia do 20. Dla modelu jednokrotnie, a następnie dwunastokrotnie różnicowanego mamy:

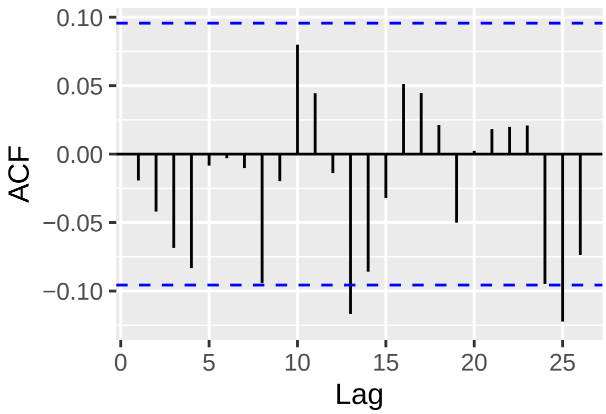

trans_Dane
−40
20
0
100
200
300
400
ACF
−0.2
0.2
0
5
10
15
20
25
Lag
PACF
−0.2
0.2
0
5
10
15
20
25
Lag

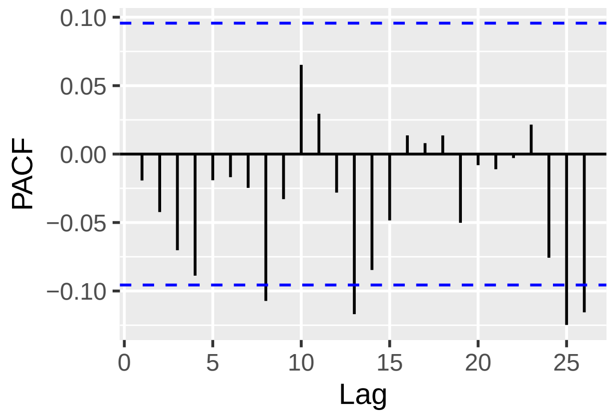



Widać tutaj, że słupek w opóźnieniu 1 zarówno w ACF jak i PACF jest poza pasem. Rozważmy, gdy P=1 lub gdy Q=1.

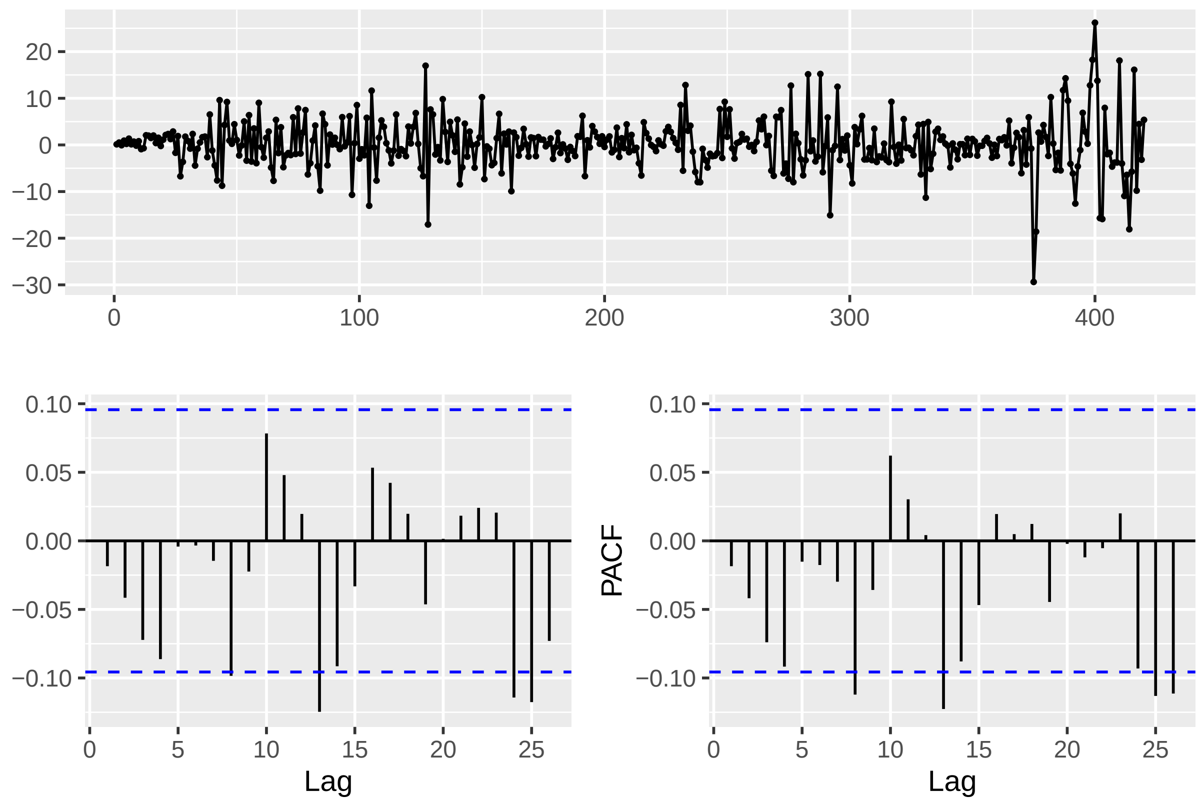
Proponujemy modele: SARIMA(0,1,1)(0,0,1)[12] oraz SARIMA(0,1,1)(1,0,0)[12].




SARIMA(0,1,1)(0,0,1)[12].

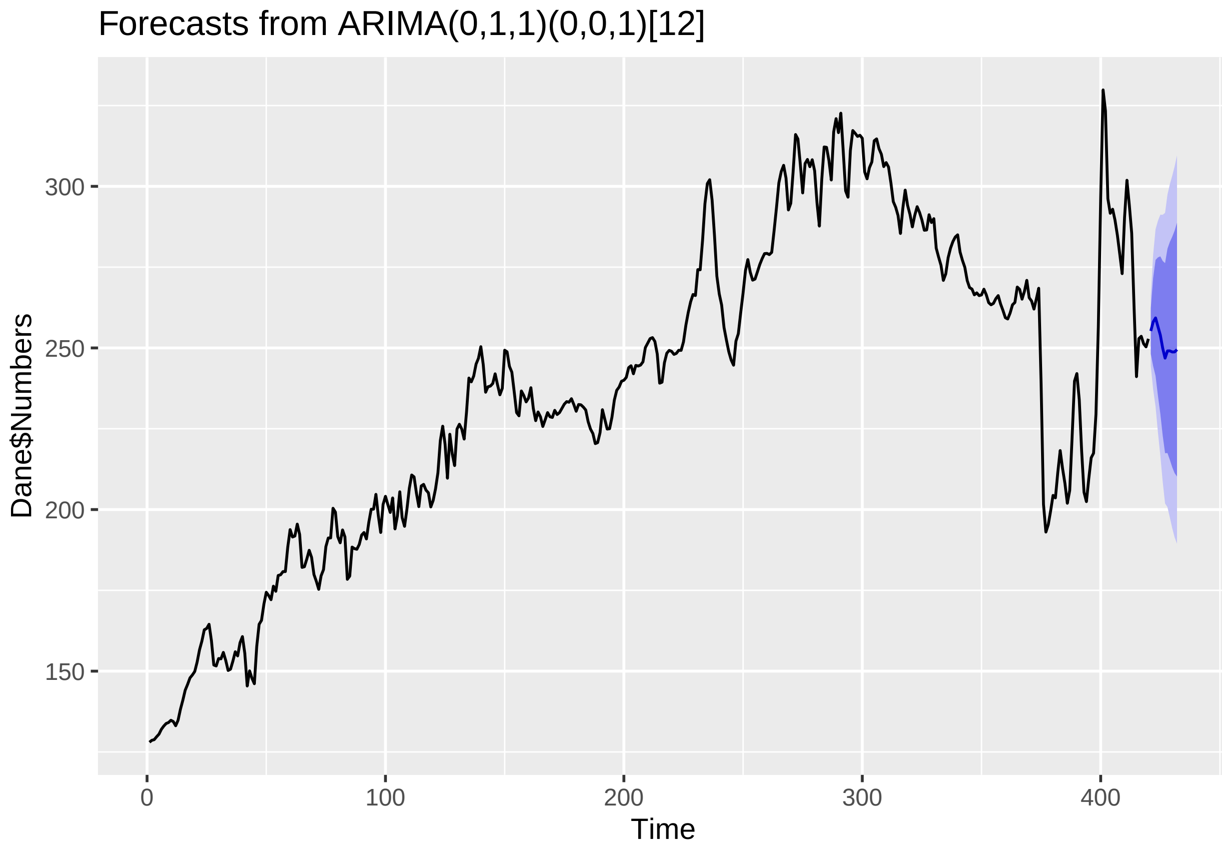



SARIMA(0,1,1)(1,0,0)[12] Słupki w ACF i PACF są wiarygodne do opóźnienia 20/21, gdzie widać, że 95% wartości funkcji ACF i PACF znajduje się w pasie. Można powiedzieć, że reszty zachowują się na poziomie ACF i PACF przyzwoicie.

```
## [1] 2627.734
```

```
## [1] 2630.154
```

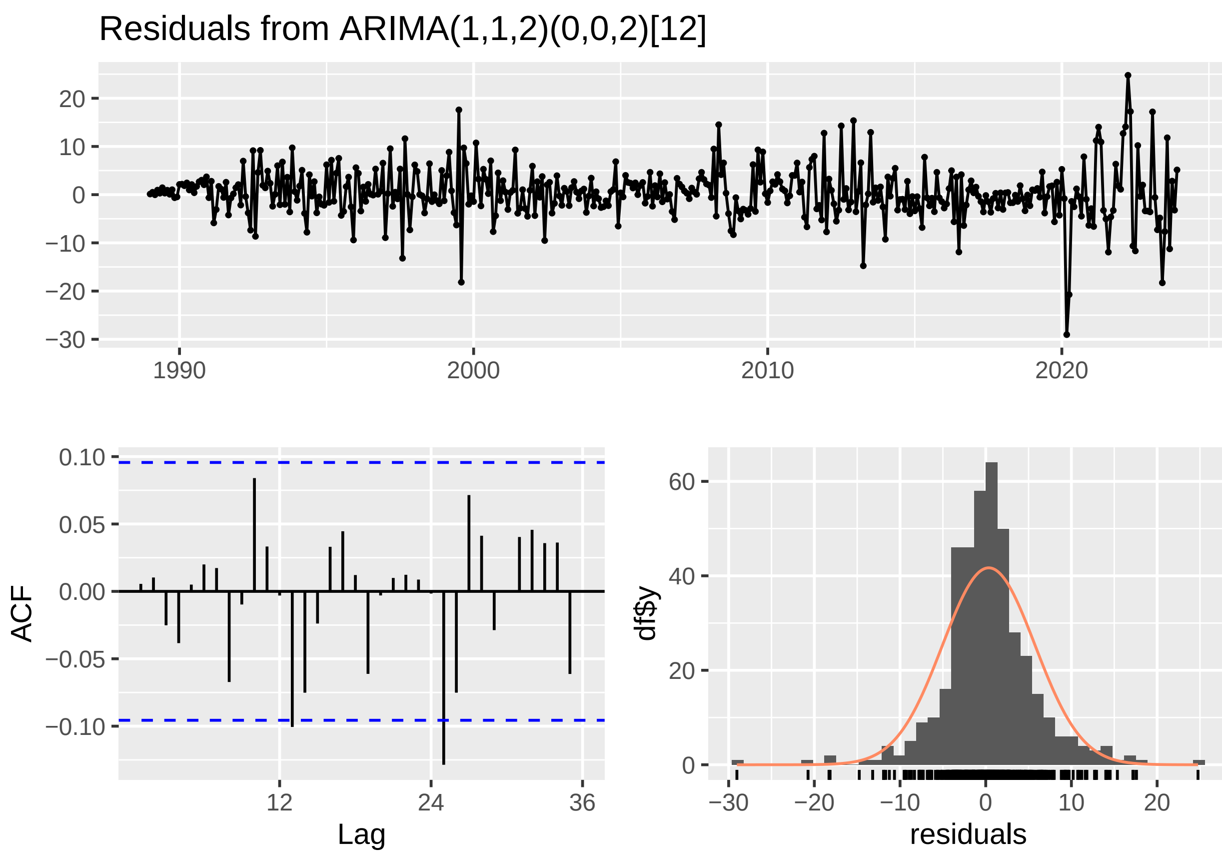
Według kryterium AIC widać, że model pierwszy, czyli SARIMA(0,1,1)(0,0,1)[12] okazał się być lepszy. Poniższy wykres przedstawia prognozy średnich cen na kolejne 12 okresów.



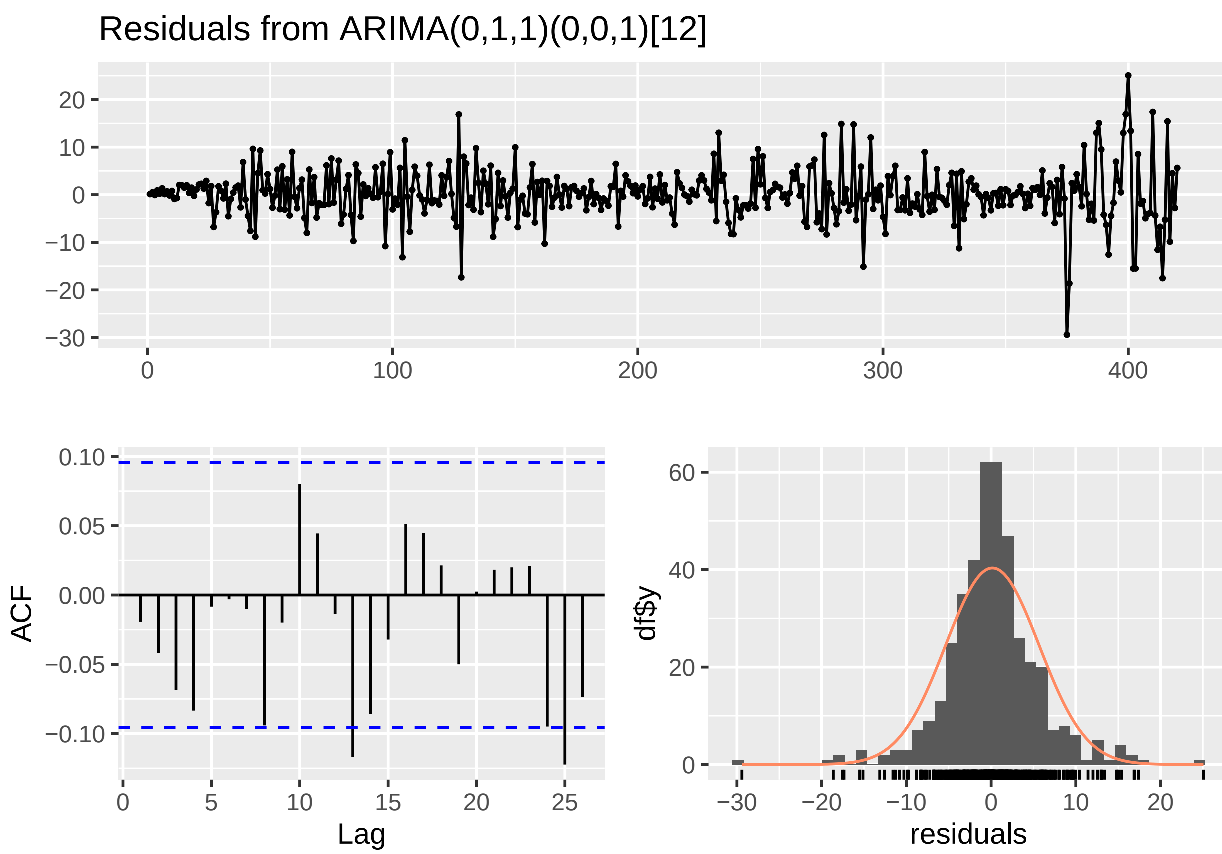

Weźmy pod uwagę również model, który został wygenerowany przez program jako najlepszy:

```
## Series: dane_ts
## ARIMA(1,1,2)(0,0,2)[12]
##
## Coefficients:
##             ar1         ma1         ma2       sma1        sma2
##       0.8334341  -0.2571587  -0.6086689  0.1738150  -0.1333488
## s.e.  0.0643462   0.0624120   0.0432368  0.0517901   0.0562271
##
## sigma^2 = 29.71181:  log likelihood = -1303.31
## AIC=2618.61   AICc=2618.82   BIC=2642.84
```

Auto.arima sugeruje, że lepszym modelem będzie SARIMA(1,1,2)(0,0,2)[12]. Sprawdźmy zatem, który z tych trzech jest najlepszy.

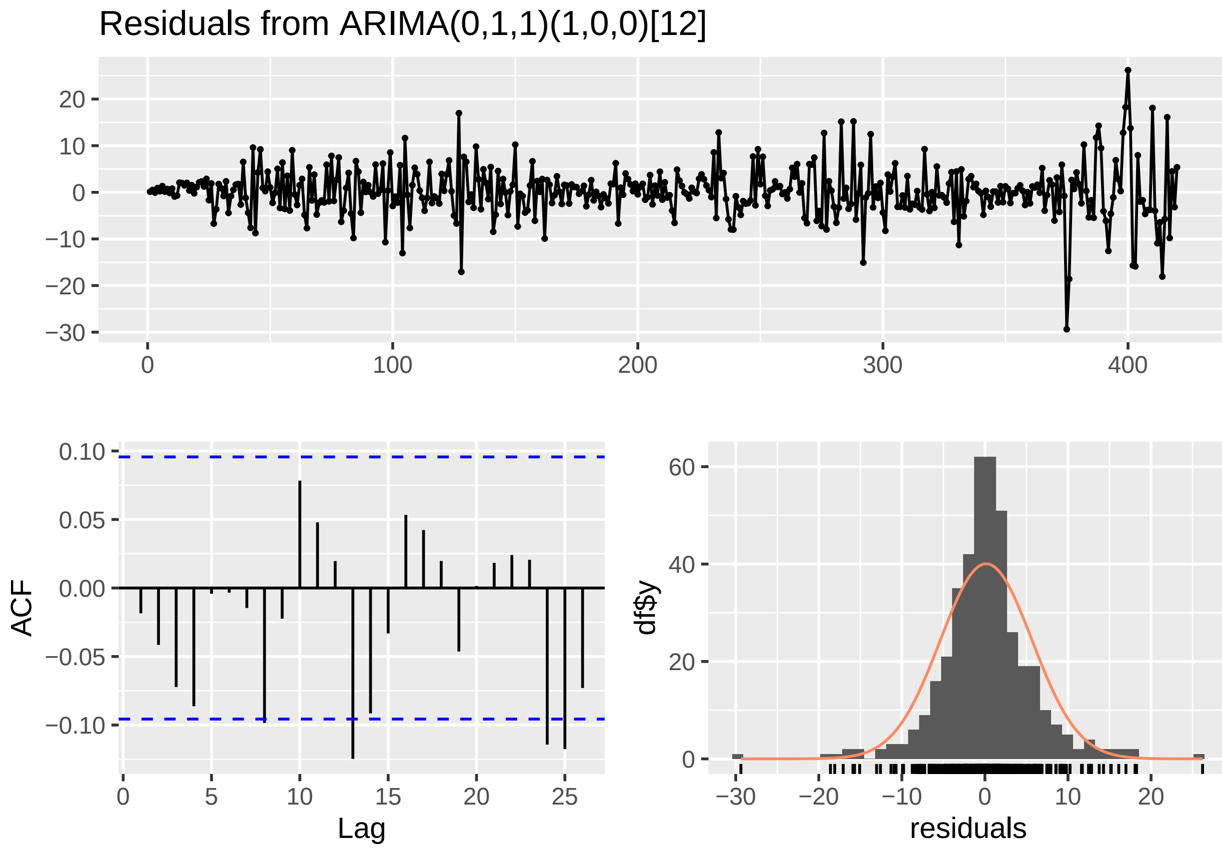



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,2)(0,0,2)[12]
## Q* = 17.136373, df = 19, p-value = 0.5806293
##
## Model df: 5.   Total lags used: 24
## Series: dane_ts
## ARIMA(1,1,2)(0,0,2)[12]
##
## Coefficients:
##             ar1         ma1         ma2        sma1        sma2
##       0.8334341  -0.2571587  -0.6086689  0.1738150  -0.1333488
## s.e.  0.0643462   0.0624120   0.0432368  0.0517901   0.0562271
##
## sigma^2 = 29.71181:  log likelihood = -1303.31
## AIC=2618.61   AICc=2618.82   BIC=2642.84
##
## Training set error measures:
##                        ME        RMSE         MAE          MPE         MAPE
## Training set 0.3540617322 5.411778938 3.723139817 0.1628683032 1.600275364
##                      MASE          ACF1
## Training set 0.2308638276 0.005554213252
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,0,1)[12]
## Q* = 12.675461, df = 8, p-value = 0.1235146
##
## Model df: 2.   Total lags used: 10
```

```
## Series: Dane$Numbers
## ARIMA(0,1,1)(0,0,1)[12]
##
## Coefficients:
##             ma1       sma1
##       0.6424100  0.1859975
## s.e.  0.0392907  0.0560956
##
## sigma^2 = 30.62314:  log likelihood = -1310.87
## AIC=2627.73   AICc=2627.79   BIC=2639.85
##
## Training set error measures:
##                       ME        RMSE         MAE          MPE        MAPE
## Training set 0.149323442 5.514018919 3.804317128 0.07649082421 1.631100705
##                     MASE           ACF1
## Training set 0.8898187642 -0.01927913026
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(1,0,0)[12]
## Q* = 13.399661, df = 8, p-value = 0.09881841
##
## Model df: 2.   Total lags used: 10
```

```
## Series: Dane$Numbers
## ARIMA(0,1,1)(1,0,0)[12]
##
## Coefficients:
##             ma1       sar1
##       0.6413770  0.1440596
## s.e.  0.0392832  0.0504519
##
## sigma^2 = 30.81325:  log likelihood = -1312.08
## AIC=2630.15   AICc=2630.21   BIC=2642.27
##
## Training set error measures:
##                       ME        RMSE         MAE           MPE        MAPE
## Training set 0.15455994 5.531108152 3.797746676 0.07826085922 1.626785245
##                       MASE           ACF1
## Training set 0.8882819544 -0.01847079335
```

Okazuje się, że model wygenerowany automatycznie jest najlepszy nie tylko pod kątem kryterium AIC, ale również błędów średniokwadratowych (RMSE) oraz log likelihood. Zatem z rozważanych modeli najlepszy jest SARIMA(1,1,2)(0,0,2)[12].