ARIMA by Rob J Hyndman

Bartosz L.

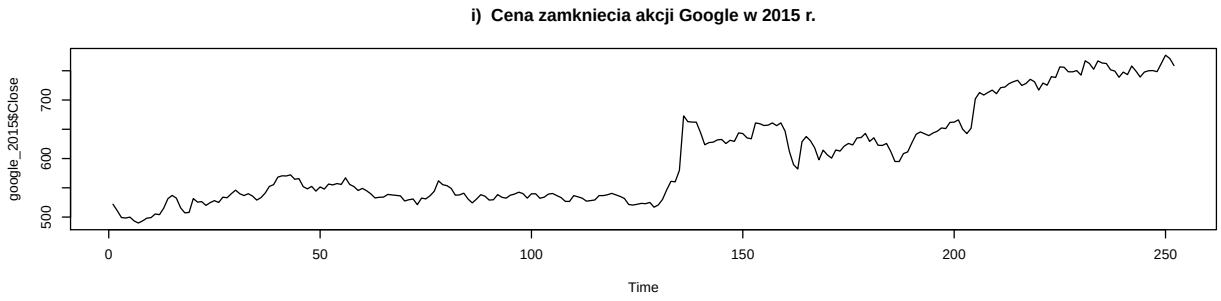
7 05 2021

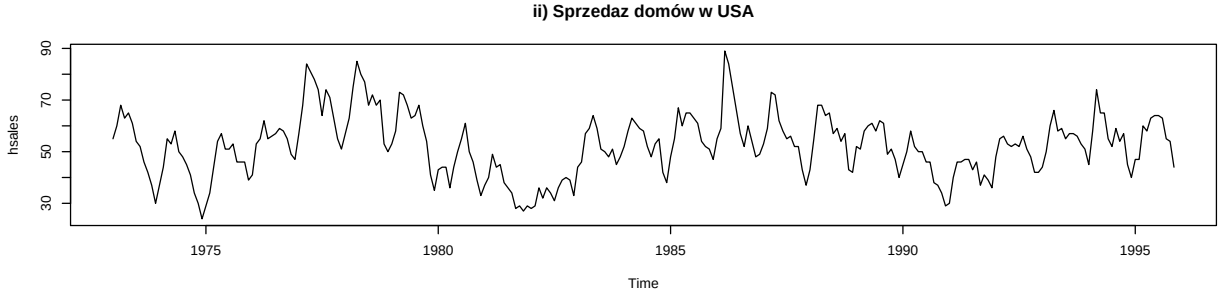
Z książki “Forecasting: Principles and Practice” pozwoliłem sobie wybrać najistotniejsze (moim okiem) informacje z rozdziału 9-tego “ARIMA models”.

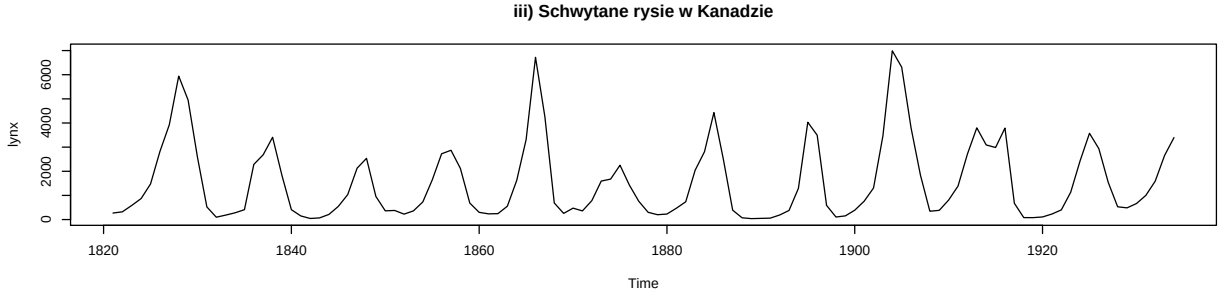
Przypomnę stacjonarność- Stacjonarny szereg czasowy to taki, którego właściwości statystyczne nie zależą od czasu, w którym szereg jest obserwowany.

Niektóre przypadki mogą być mylące - szereg czasowy z zachowaniem cyklicznym (ale bez trendu lub sezonowości) jest stacjonarny. Dzieje się tak dlatego, że cykle nie mają stałej długości, więc przed obserwacją szeregu nie możemy być pewni, gdzie będą szczyty i przełomy cykli.

Dla porównania, wywołam trzy szeregi czasowe.



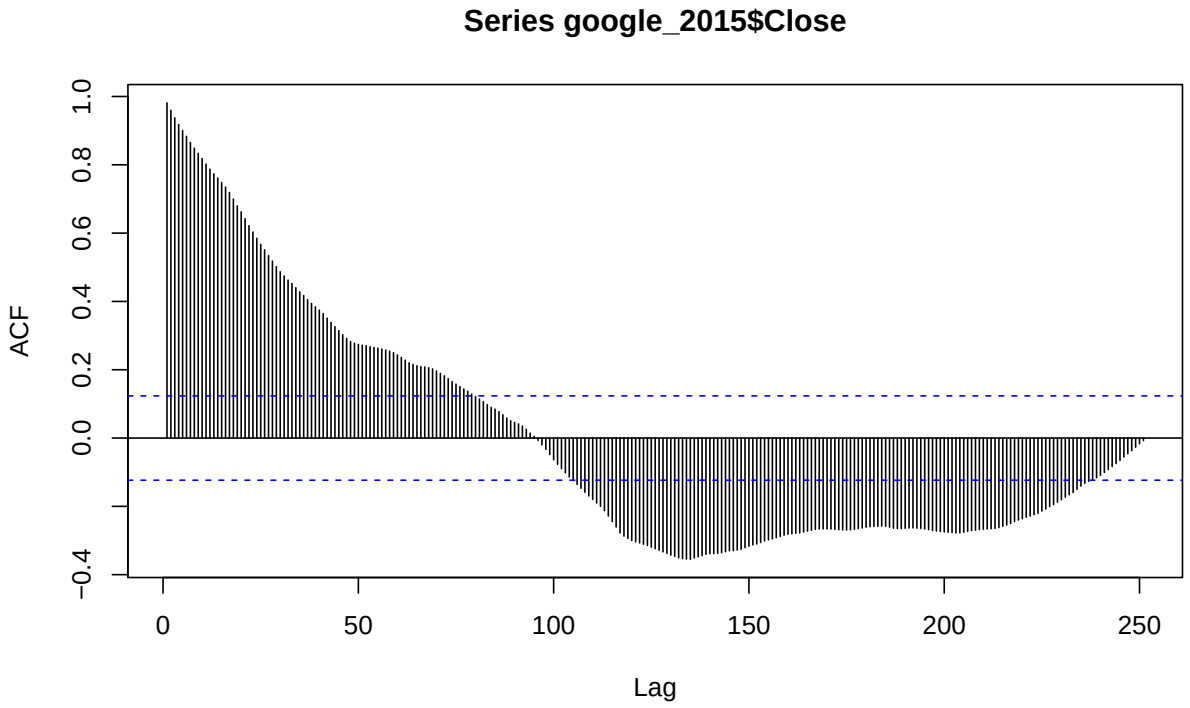


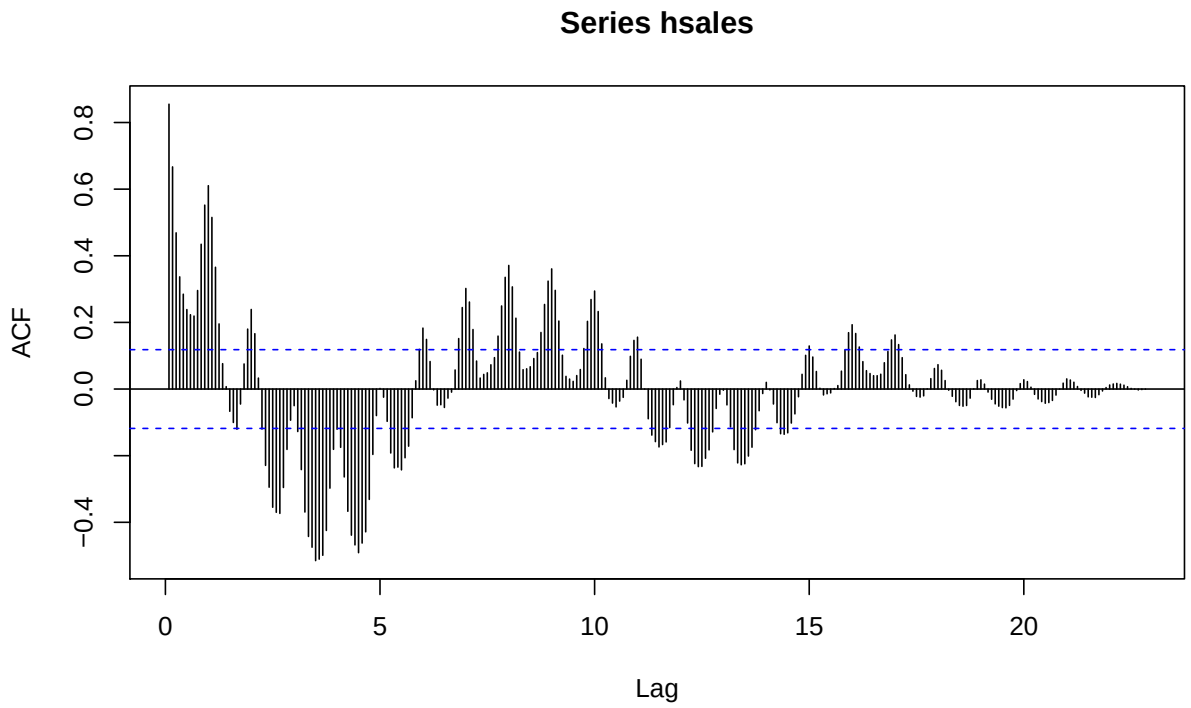


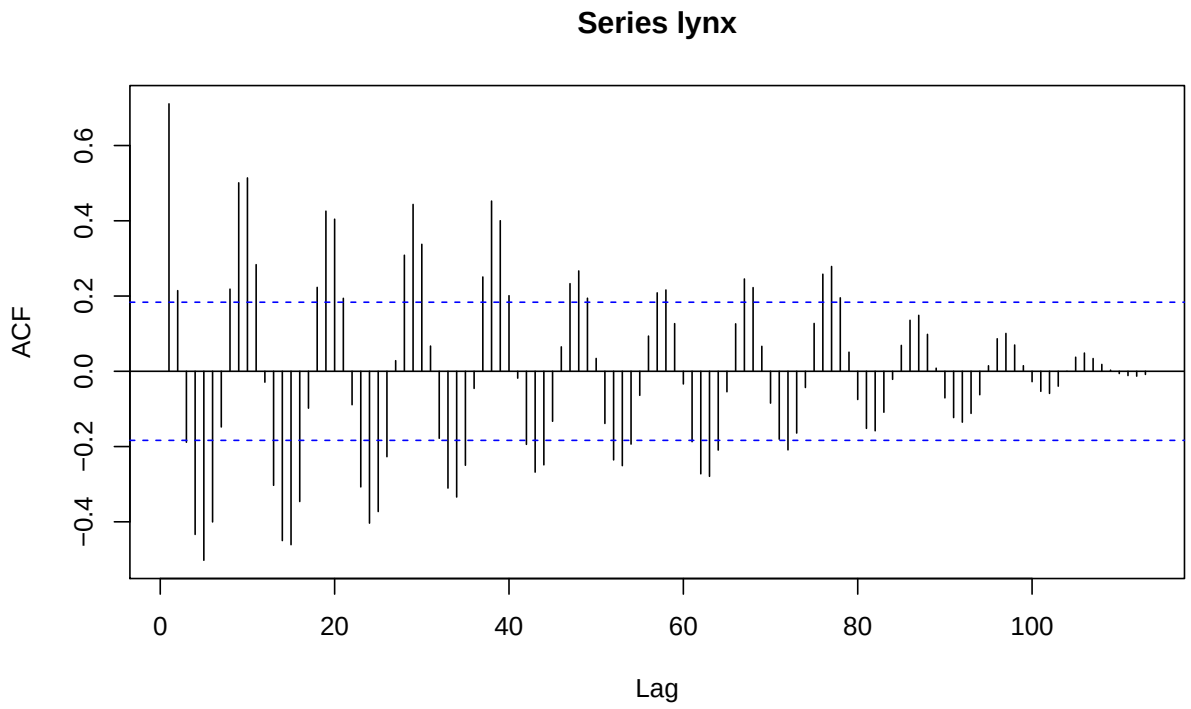
W i) możemy dostrzec trend, natomiast w ii) występuje pewna zauważalna sezonowość. Zatem te szeregi czasowe nie są stacjonarne. Na pierwszy rzut oka może się wydawać, że silne cykle w szeregu czasowym "Lynx" czynią go niestacjonarnym. Jednak cykle te są aperiodyczne - powstają, gdy populacja rysia staje się zbyt duża w stosunku do dostępnego pokarmu, przez co przestaje się on rozmnażać, a liczebność populacji spada do niskiego poziomu, po czym regeneracja źródeł pokarmu pozwala na ponowny wzrost liczebności populacji, i tak dalej. W dłuższej perspektywie czasowej nie da się przewidzieć przebiegu tych cykli. Dlatego też szereg czasowy jest stacjonarny.

Potwierdźmy nasze podejrzenia za pomocą funkcji acf()

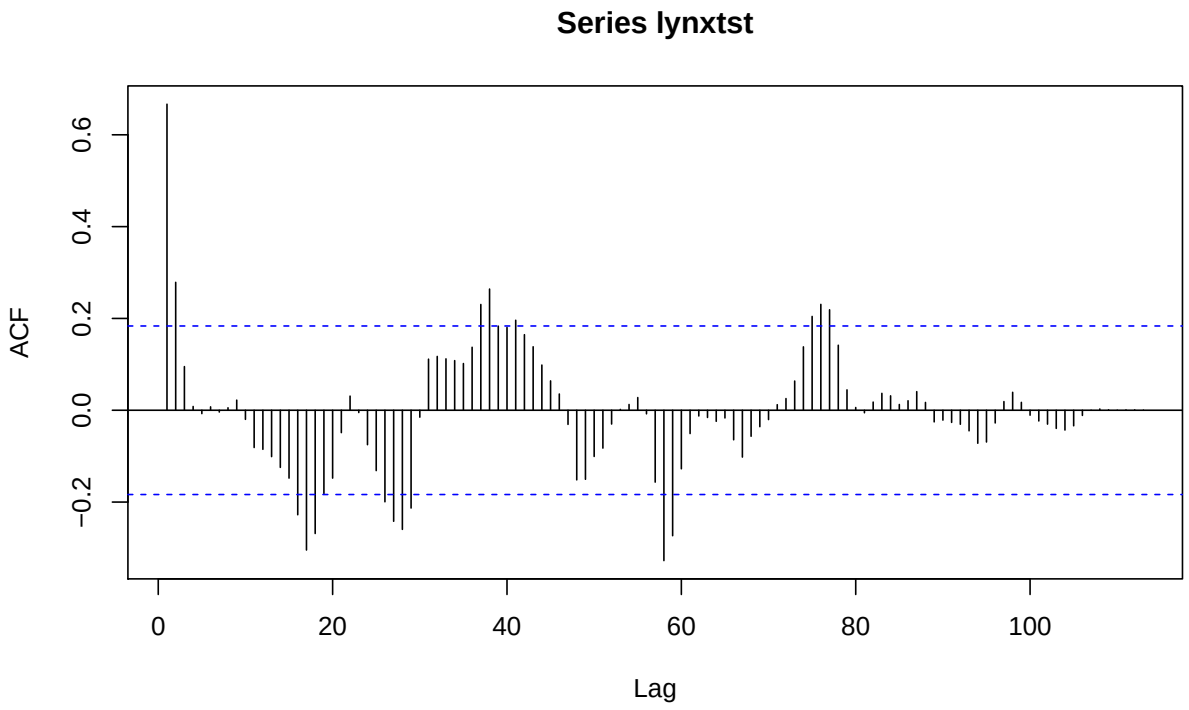
```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast
```

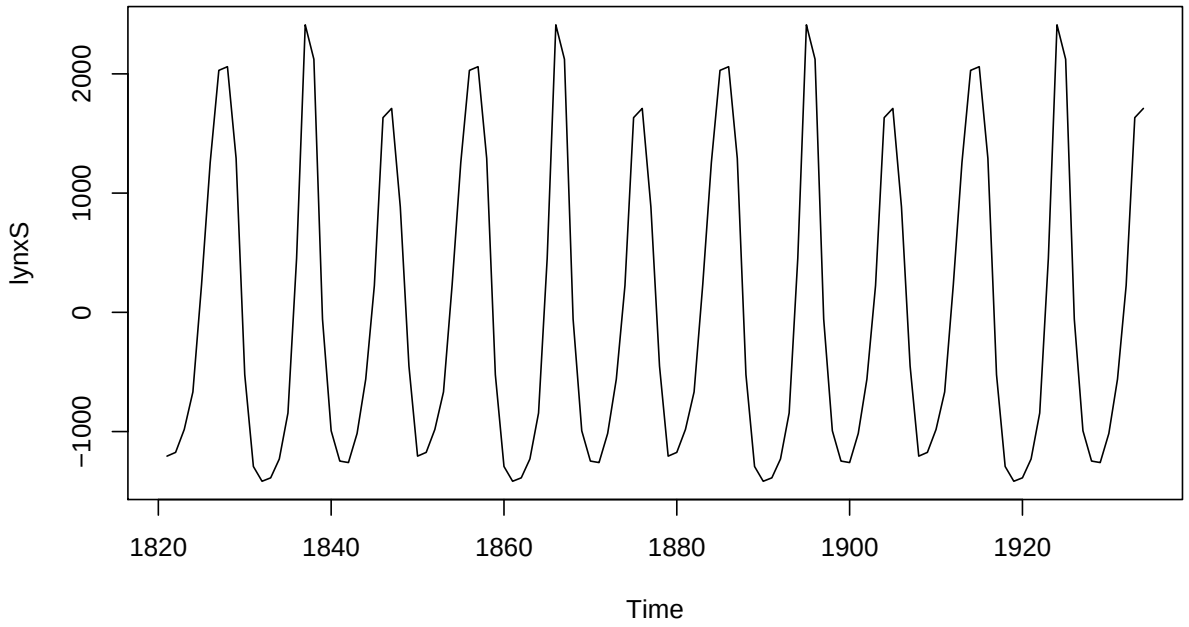


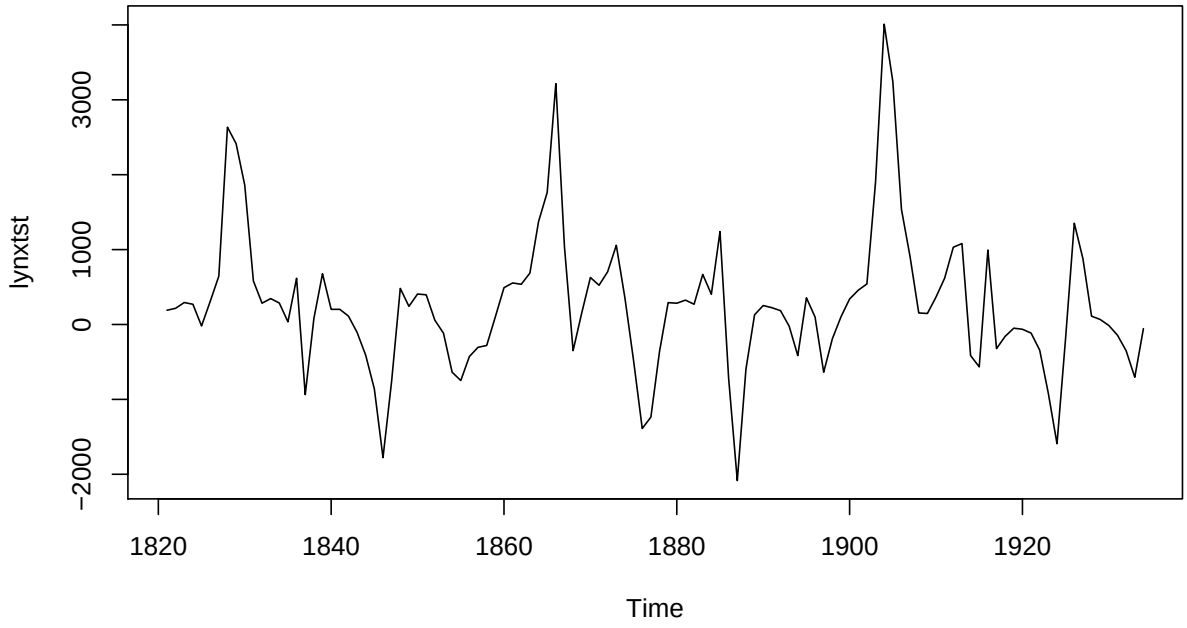





```
##  [1] -0.50221758 -0.40034959 -0.14798466  0.21836506  0.50090800  0.51390728
##  [7]  0.28344559 -0.02889004 -0.30304677 -0.44988622 -0.46078109
```


Series lynxtst
ACF
0.6
0.4
0.2
0.0
−0.2
0
20
40
60
80
100
Lag


lynxS
2000
1000
0
−1000
1820
1840
1860
1880
1900
1920
Time



Wychodzi na to, że w zbiorze "lynx" jednak pojawia się jakiś efekt sezonowości, ale skoro opis zbioru wskazuje na brak ustalonej długosci sezonu (karmienie rysi przy ich określonej populacji), a na uzależnienie od panującej sytuacji, to czy jest to faktyczna sezonowość? Skoro zostało ustalone, że jest to cykliczność, to jak precyzyjnie okreslić kiedy mamy do czynienia z czym? ... do zastanowienia

Ponieważ zajmowałem się zbiorem TSA::milk, dlatego też dalsze rozważania będe przeprowadzał na nim.

Model spaceru losowego, czyli alternatywne podejście do sezonowości

Szereg różnicowy jest zmianą pomiędzy kolejnymi obserwacjami w szeregu pierwotnym i może być zapisany jako $y'_t = y_t - y_{t-1}$. Gdy szereg różnicowany jest białym szumem, model dla szeregu pierwotnego można zapisać jako $y_t - y_{t-1} = \varepsilon_t$ gdzie ε_t oznacza biały szum. Po prostym przekształceniu, otrzymujemy model spaceru losowego: $y_t = y_{t-1} + \varepsilon_t$.

Czasami zróżnicowane dane nie wydają się być stacjonarne i może być konieczne, aby zróżnicować dane po raz drugi, aby uzyskać stacjonarny szereg:

$$y''_t = y'_t - y'_{t-1} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$$

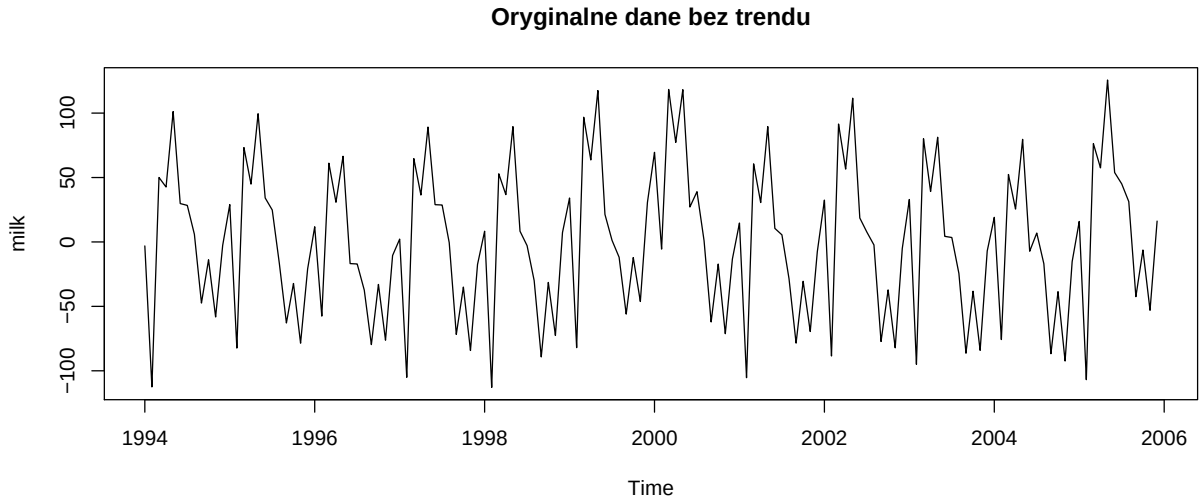
Dla dyskretnego szeregu czasowego różnica drugiego rzędu reprezentuje krzywiznę szeregu w danym punkcie czasowym. Jeśli różnica drugiego rzędu jest dodatnia, to szereg czasowy jest zakrzywiony w górę w tym czasie, a jeśli jest ujemna, to szereg czasowy jest zakrzywiony w dół w tym czasie (analogia z drugą pochodną).

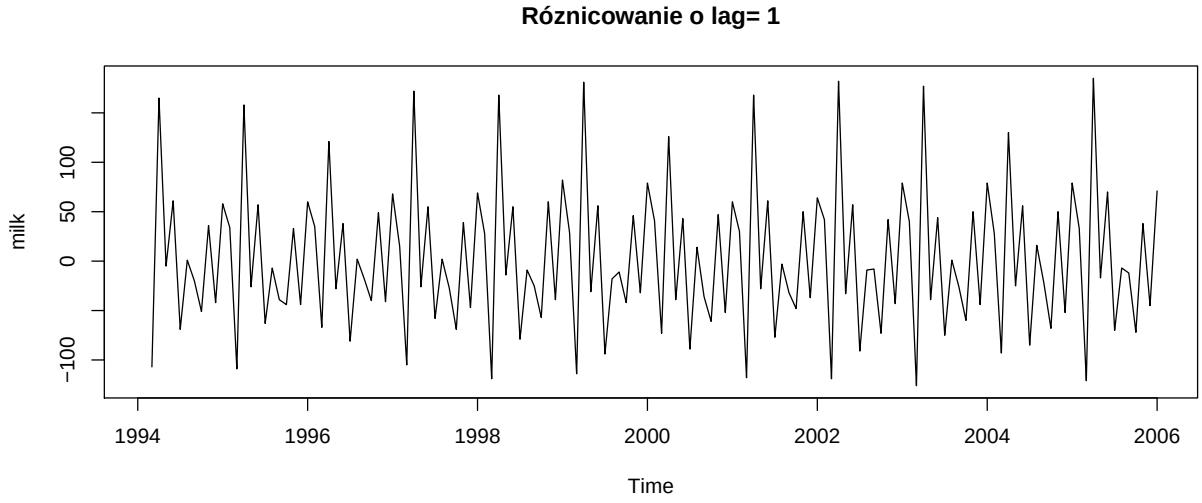
Podobnie postepuję się w przypadku różnicowania sezonowego, które jest różnicą między obserwacją a poprzednią obserwacją z tego samego sezonu. Niech $m \in \mathbb{N}$ będzie m-tym sezonem. Wówczas sezonowy szereg różnicowy jest postaci: $y'_t = y_t - y_{t-m}$. Przy czym, jeżeli jest on białym szumem, to otrzymujemy model postaci $y_t = y_{t-m} + \varepsilon_t$.

Różnice pierwszego stopnia są zmianą pomiędzy jedną obserwacją a drugą. Różnice sezonowe to zmiana

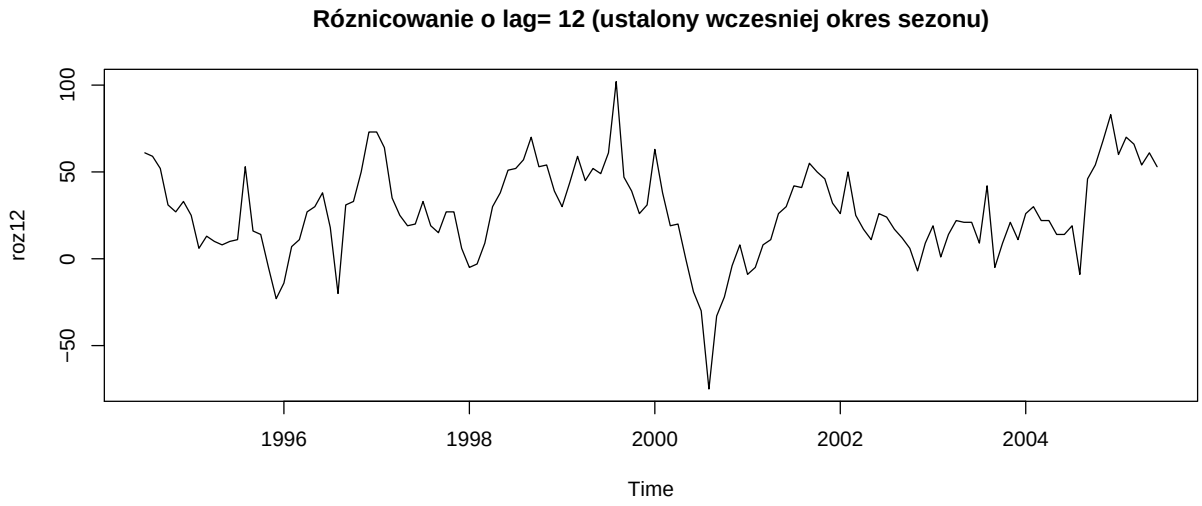
między jednym rokiem a drugim. Inne opóźnienia najprawdopodobniej nie będą miały większego sensu interpretacyjnego i należy ich unikać.

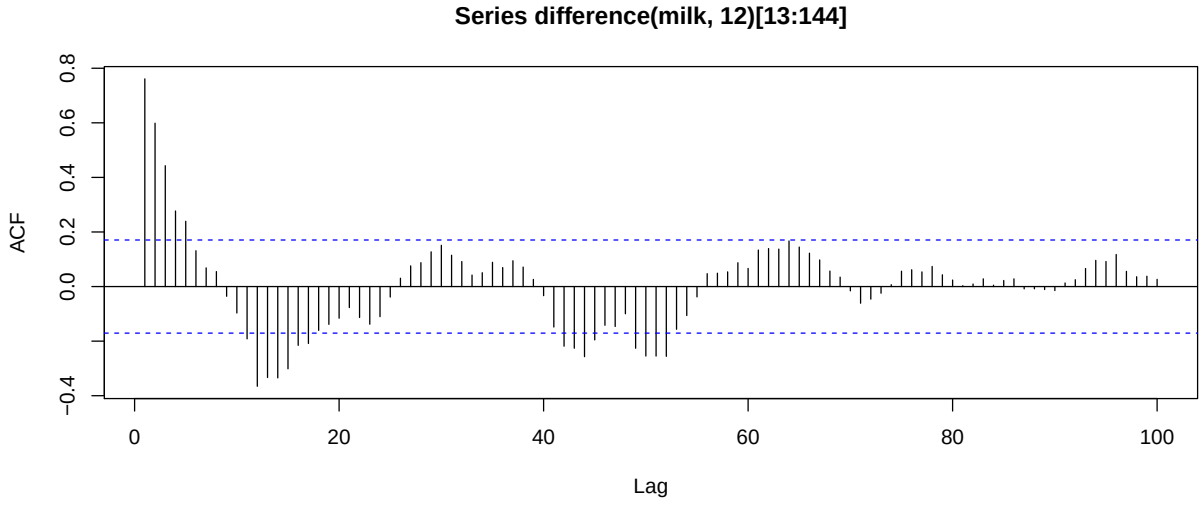
Sprawdźmy to na zbiorze milk, w którym sezonowość jest równa 12. Do przeeprowadzenia różnicowania, skorzystamy z funkcji difference() (z pakietu tibble).



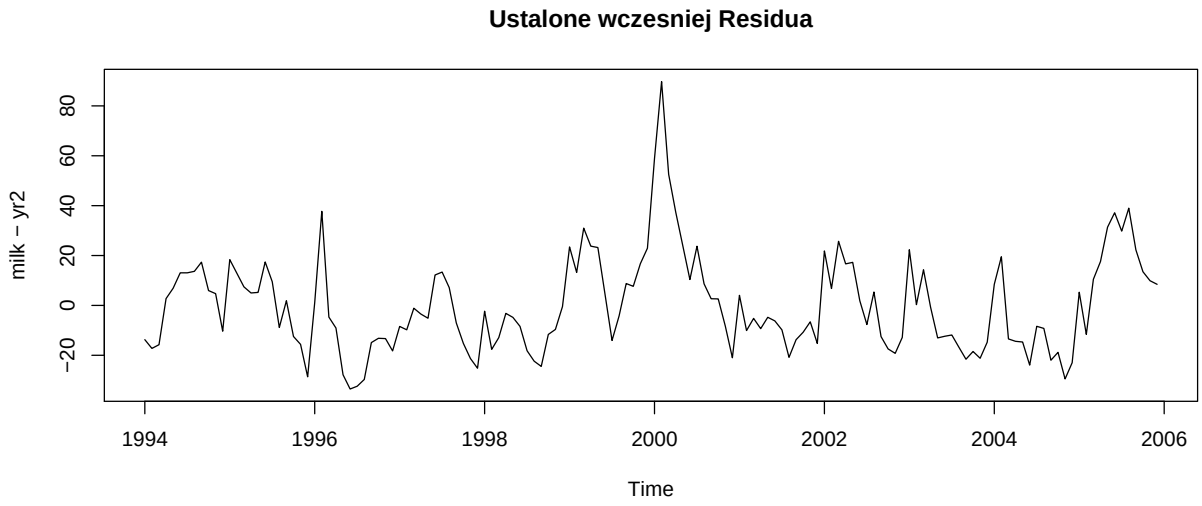


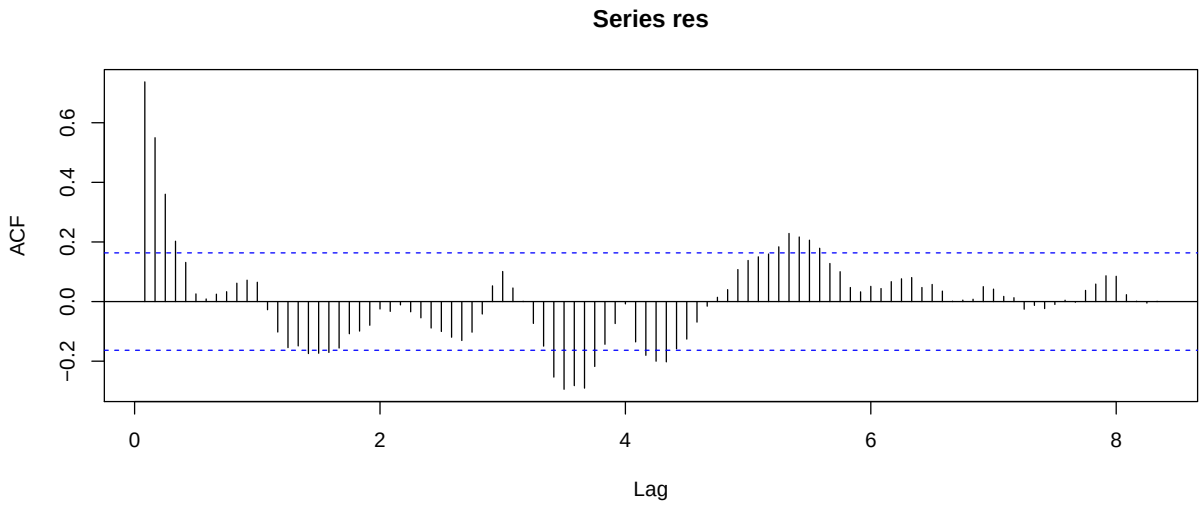

```
## [1] "Brak jakiegokolwiek skutku"
```



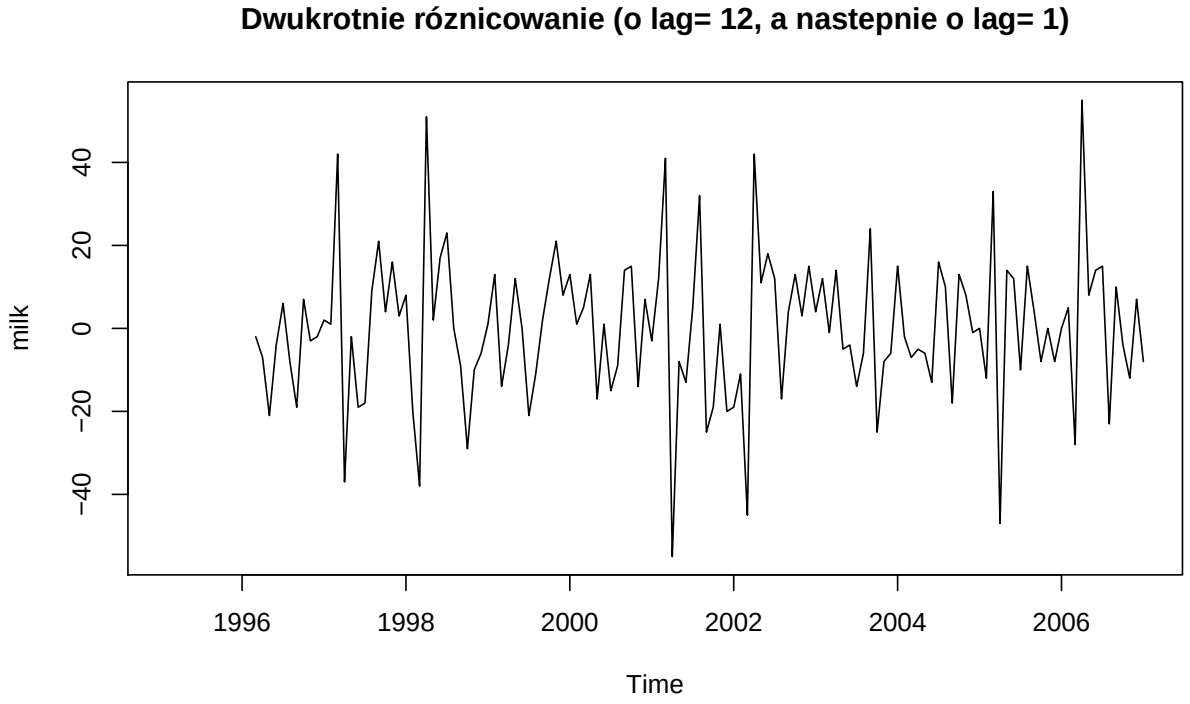


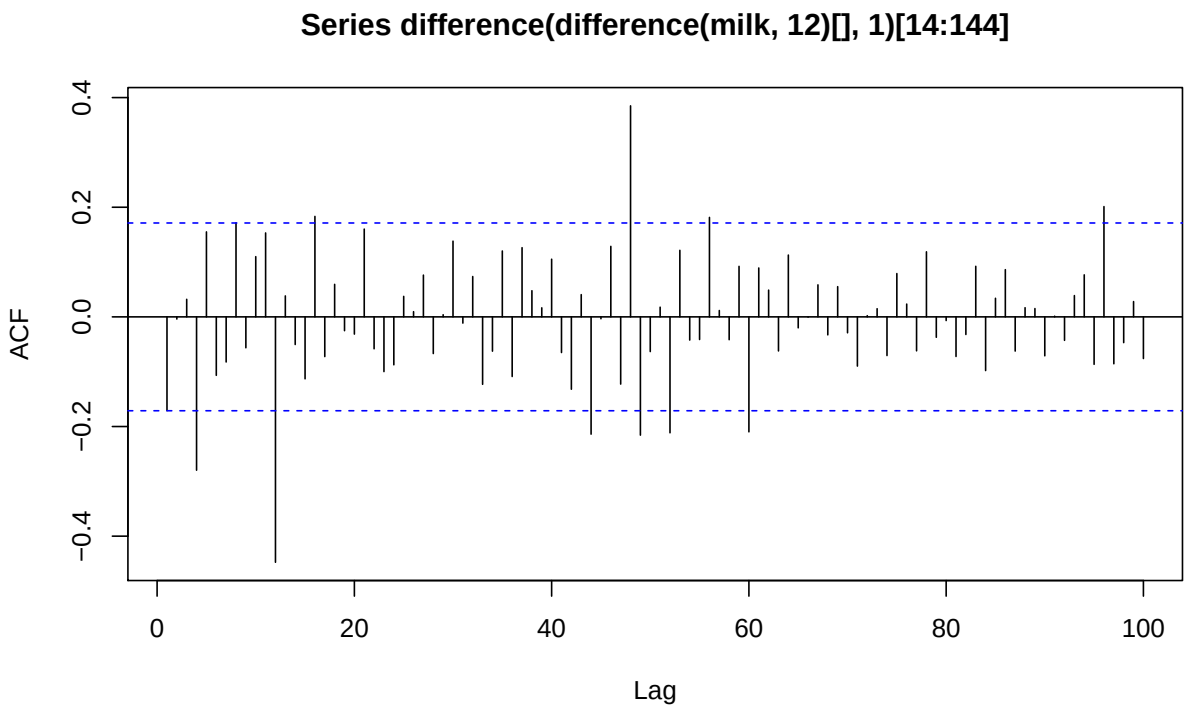

```
## Sukces! Udało się nam otrzymać szereg stacjonarny.
## Potwierdza to ustalony okres sezonowości równy 12.

## Dla porównania, poprzednio ustalony stacjonarny szereg residuów
```





Uwaga! Należy pamiętać, że zastosowanie większej liczby różnic niż wymagana spowoduje fałszywą dynamikę lub autokorelacje, które w rzeczywistości nie istnieją w szeregu czasowym. Dlatego wykonaj tak mało różnic, jak to konieczne, aby uzyskać szereg stacjonarny.






```
## [1] -0.003426289  0.128618154 -0.122636304  0.384970139 -0.215727655
## [6] -0.063299962
```

Istotnie, ponowne zróżnicowanie doprowadziło do wskazania autokorelacji na pozycji 48 równej 0.384970139.

Test KPSS

Test KPSS (od nazwisk Kwiatkowski–Phillips–Schmidt–Shin), wywoływany funkcją unitroot_kpss(), jest testem statystycznym sprawdzającym hipotezę zerową o stacjonarności szeregu czasowego (niestety statystyka testu KPSS ma złożoną konstrukcję oraz bardzo skomplikowany rozkład prawdopodobieostwa i nie jestem w stanie ich przywołać z pełnym zrozumieniem).

Możemy również wykorzystac ten test, aby określić potrzebną ilość różnicowania danych. Proces wykorzystania sekwencji testów KPSS do wyznaczenia odpowiedniej liczby dla pierwszych różnic realizowany jest za pomocą unitroot_ndiffs().

Możemy również określić, czy wymagane jest różnicowanie sezonowe. W tym celu możemy użyć funkcji unitroot_nsdiffs(). Wykorzystuje ona pomiar siły sezonowości zdefiniowanej wzorem

$$F_S = max\left(0, 1 - \frac{Var(\varepsilon_t)}{Var(S_t + \varepsilon_t)}\right) \quad , dla \;\; F_S \in [0, 1]$$

gdzie jeżeli $F_S < 0.64$, to nie jest wymagane różnicowanie sezonowe.

Przykładowe zastosowanie na zbiorze "milk":

```
## Test na stacjonarność szeregu milk
```

```
##   kpss_stat kpss_pvalue
##    2.615976    0.010000
```

```
## p-value = 0.010000 < 0.05 zatem odrzucamy hipotezę zerową o stacjonarności szeregu
```

```
## Test na stacjonarność szeregu res (milk)
```

```
##   kpss_stat kpss_pvalue
##   0.1040031   0.1000000
```

```
## p-value = 0.10000 > 0.05 zatem nie ma podstaw do odrzucenia hipotezy zerowej
```

```
## Liczba potrzebnych pierwszych różnic dla zbioru milk wynosi:
```

```
## ndiffs
##      1
```

```
## Liczba potrzebnych pierwszych różnic dla zbioru res wynosi:
```

```
## ndiffs
##      0
```

```
## Liczba potrzebnych różnic sezonowych dla zbioru milk wynosi:
```

```
## nsdiffs
##       0
```

```
## Liczba potrzebnych różnic sezonowych dla zbioru res wynosi:
```

```
## nsdiffs
##       0
```

Operator przesunięcia wstecznego

Operator przesunięcia wstecznego definiujemy

$$By_t = y_{t-1}$$

W przypadku podwójnego zastosowania mamy

$$B(By_t) = B^2 y_t = y_{t-2}$$

Przykładowo, jeżeli pragniemy wskazać obecny miesiąc dokładnie rok wcześniej, możemu użyć notacji $B^{12}y_t = y_{t-12}$.

Za pomocą operatora przesunięcia wstecznego, możemy zapisać szereg różnicowy. Różnice pierwszego stopnia możemy zapisać jako:

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1-B)y_t$$

czyli różnica może być reprezentowana przez $(1-B)$.

W przypadku różnicy drugiego stopnia mamy:

$$y''_t = y_t - 2y_{t-1} + y_{t-2} = (1-2B+B^2)y_t = (1-B)^2 y_t$$

W ogólnym przypadku możemy zapisać różnicę $d-tego$ stopnia jako $(1-B)^d y_t$.

Model Autoregresjii

Model autoregresjii rzędu p definiujemy następująco:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$

gdzie ε_t jest białym szumem. Taki model oznaczamy poprzez $AR(p)$.

Co warto zauważyć, zmiana parametrów $\phi_1, \ldots, \phi_p$ skutukje różnymi "wzorcami" szeregów czasowych. Natomiast wariancja błędu ε_t wpływa tylko na skalę szeregu, a nie jego wzorzec.

Niech $AR(1)$. Wtedy:

i) gdy $\phi_1 = 0$, y_t jest równoważne białemu szumowi;

ii) gdy $\phi_1 = 1$ oraz $c = 0$, y_t jest równoważne spacerowi losowemu;

iii) gdy $\phi_1 = 1$, oraz $c \neq 0$, y_t jest równoważne spacerowi losowemu o przesunięciu równym c;

iv) gdy $\phi_1 < 0$, y_t ma tendencję do oscylowania wokół średniej.

Zazwyczaj ograniczamy modele autoregresyjne do danych stacjonarnych, w którym to przypadku wymagane są pewne ograniczenia na wartości parametrów:

i) dla modelu $AR(1)$ $-1 < \phi_1 < 1$;

ii) dla modelu $AR(2)$ $-1 < \phi_2 < 1, \phi_1 + \phi_2 < 1, \phi_2 - \phi_1 < 1$.

Gdy rząd $p \geq 3$, to sprawa zaczyna się komplikować. Natomiast biblioteka "fable" radzi sobie z tymi komplikacjami podczas estymacji modelu.

Model Średniej Ruchomej

Model średniej ruchomej rzędu q wykorzystuje błędy prognoz z przeszłości w modelu podobnym do regresji, tj.:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}$$

gdzie ε_t jest białym szumem. Taki model oznaczamy poprzez $MA(q)$. Oczywiście, nie "obserwujemy" wartości ε_t (,ponieważ jest to biały szum- losowość), więc nie jest to tak naprawdę regresja w zwykłym sensie.

Uwaga! Nie należy mylić średniej modeli średniej ruchomej z wygładzaniem średniej ruchomej.Model średniej ruchomej jest wykorzystywany do prognozowania przyszłych wartości, natomiast wygładzanie średniej ruchomej jest wykorzystywane do szacowania cyklu trendu wartości przeszłych.

Analogicznie jak w przypadku modelu autoregresjii, zmiana parametrów $\theta_1, \ldots, \theta_q$ skutukje różnymi "wzorcami" szeregów czasowych. Natomiast wariancja błędu ε_t wpływa tylko na skalę szeregu, a nie jego wzorzec.

Możemy zapisać dowonly stacjonarny model $AR(p)$ w postaci modelu $MA(\infty)$. Przykładowo, dla $AR(1)$ mamy:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t = \phi_1(\phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t = \phi_1^2 y_{t-2} + \phi_1\varepsilon_{t-1} + \varepsilon_t = \phi_1^3 y_{t-3} + \phi_1^2\varepsilon_{t-2} + \phi_1\varepsilon_{t-1} + \varepsilon_t = \dots$$

Jeżeli $-1 < \phi_1 < 1$, wartości ϕ_1^k będą maleć wraz ze wzrostem k. Finalnie otrzymujemy proces $MA(\infty)$, tj.:

$$y_t = \varepsilon_t + \phi_1\varepsilon_{t-1} + \phi_1^2\varepsilon_{t-2} + \phi_1^3 y_{t-3} + \dots$$

Odwrotna zależność zachodzi, gdy na parametry $MA(q)$ nałożymy pewne ograniczenia. Wówczas model $MA(q)$ nazywamy odwracalnym. Oznacza to, że możemy zapisać dowolną odwracalną $MA(q)$ jako proces $AR(\infty)$. Przykładowo, rozważmy $MA(1)$. Wtedy $y_t = \varepsilon_t + \theta_1\varepsilon_{t-1}$. Korzystając z operatora przesunięcia wstecznego, tj.: $B\varepsilon_t = \varepsilon_{t-1}$, mamy:

$$y_t = \varepsilon_t + \theta_1\varepsilon_{t-1} = \varepsilon_t(1 + \theta_1 B) \iff \varepsilon_t = \frac{y_t}{1 + \theta_1 B} = \frac{y_t}{1 - (-\theta_1)B}$$

zauważmy, że przekształceniem jest wynik na sumę szeregu geometrycznego dla n=1 $\left(\sum_{n=1}^{\infty} aq^{n-1} = \frac{a}{1-q},\ dla\ |q| < 1\right)$.

Stąd, dla $a = y_{t-j}$ oraz $a = (-\theta_1)$ możemy zdefiniować "najnowszy" błąd jako

$$\varepsilon_t = \sum_{j=0}^{\infty} (-\theta_1)^j y_{t-j}$$

Wniosek:

i) jeżeli $|\theta_1| > 1$, to szereg jest rozbieżny oraz wagi θ_1 rosną wraz ze wzrostem opóźnienia j, więc im bardziej odległe obserwacje, tym większy ich wpływ na bieżący błąd;

ii) jeżeli $|\theta_1| = 1$, to szereg jest rozbieżny oraz wagi θ_1 mają stałą wielkość, a obserwacje odległe mają taki sam wpływ jak obserwacje niedawne;

iii) jeżeli $|\theta_1| < 1$, to szereg jest zbieżny oraz "nowsze" obserwacje mają większą wagę niż obserwacje z bardziej odległej przeszłości.

Zatem proces $MA(1)$ jest odwracalny dla $|\theta_1| < 1$. W przypadku $MA(2)$, aby model był odwracalny, wymaga się, aby $|\theta_2| < 1$, $\theta_1 + \theta_2 > -1$ oraz $\theta_1 - \theta_2 < 1$. W przypadku bardziej sklompikowanych ograniczeń dla $q \geq 3$, znowu z pomocą przychodzi biblioteka "fable", która rozwiązuje takie problemy w ramach odpowiedniej funkcji.

(niesezonowy) Model ARIMA()

Poprzez kombinację różnicowania wraz z autoregresją oraz srednią ruchomą, otrzymujemy (niesezonowy) model ARIMA(p,d,q):

$$y'_t = c + \phi_1 y'_{t-1} + \ldots + \phi_p y'_{t-p} + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q} + \varepsilon_t$$

gdzie p jest rzędem autoregresjii, d jest stopniem pierwszego zróżnicowania oraz q jest rzędem sredniej ruchomej. Przy czym zachowane są założenia o stacjonarności oraz odwracalności dla autoregresjii oraz średniej ruchomej.

Specjalne przypadki modelu ARIMA():

i) gdy ARIMA(0,0,0), to mamy do czynienia z białym szumem;

ii) gdy ARIMA(0,1,0) bez stałej c, to mamy do czynienia ze spacerem losowym;

iii) gdy ARIMA(0,1,0) wraz ze stałą c, to mamy do czynienia ze spacerem losowym z przesunięciem;

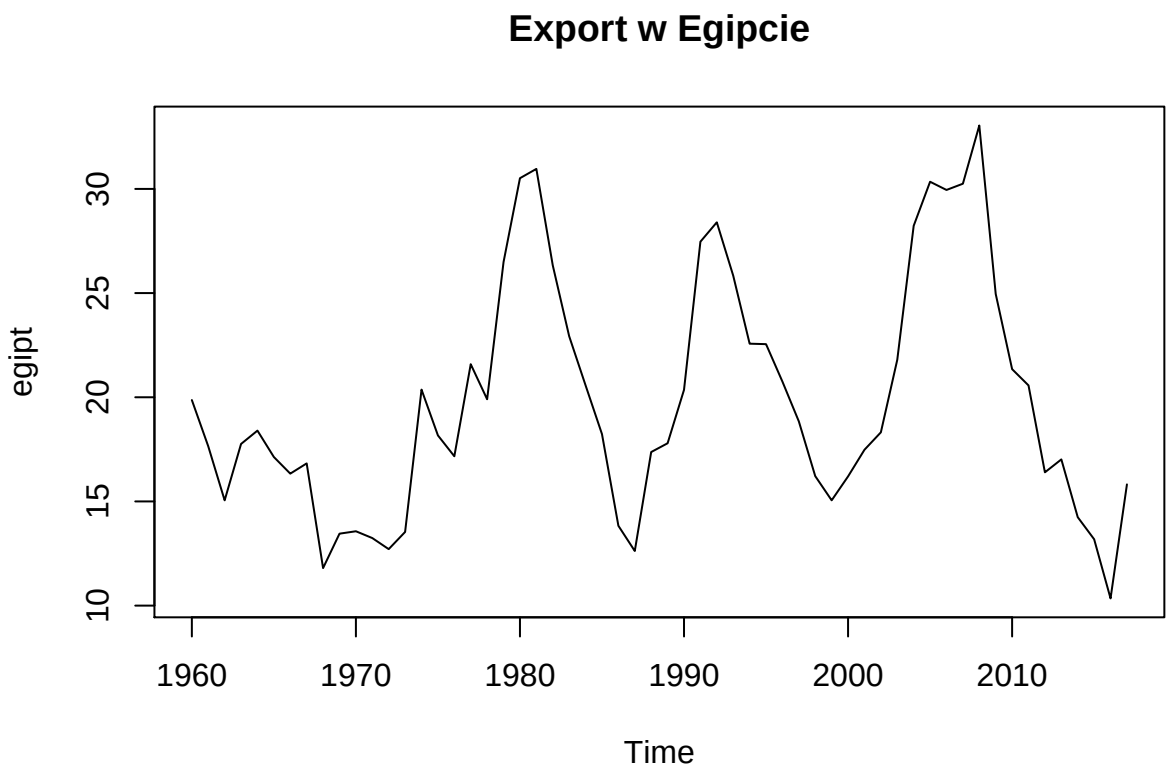
iv) gdy ARIMA(p,0,0), to mamy do czynienia z autoregresją;

v) gdy ARIMA(0,0,q), to mamy do czynienia ze średnią ruchomą.

Dla łatwiejszego tworzenia bardziej skomplikowanych modeli wykorzystujemy notacje przesunięcia wstecznego, aby przekształcić model (do postaci $AR(p)(d\ \ różnicowanie) = MA(q)$):

$$(1 - \phi_1 B - \ldots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \ldots + \theta_q B^q)\varepsilon_t$$

Jak się okazuję funkcja ARIMA() z biblioteki "fable" automatycznie dobiera odpowiednie parametry p, d, q.

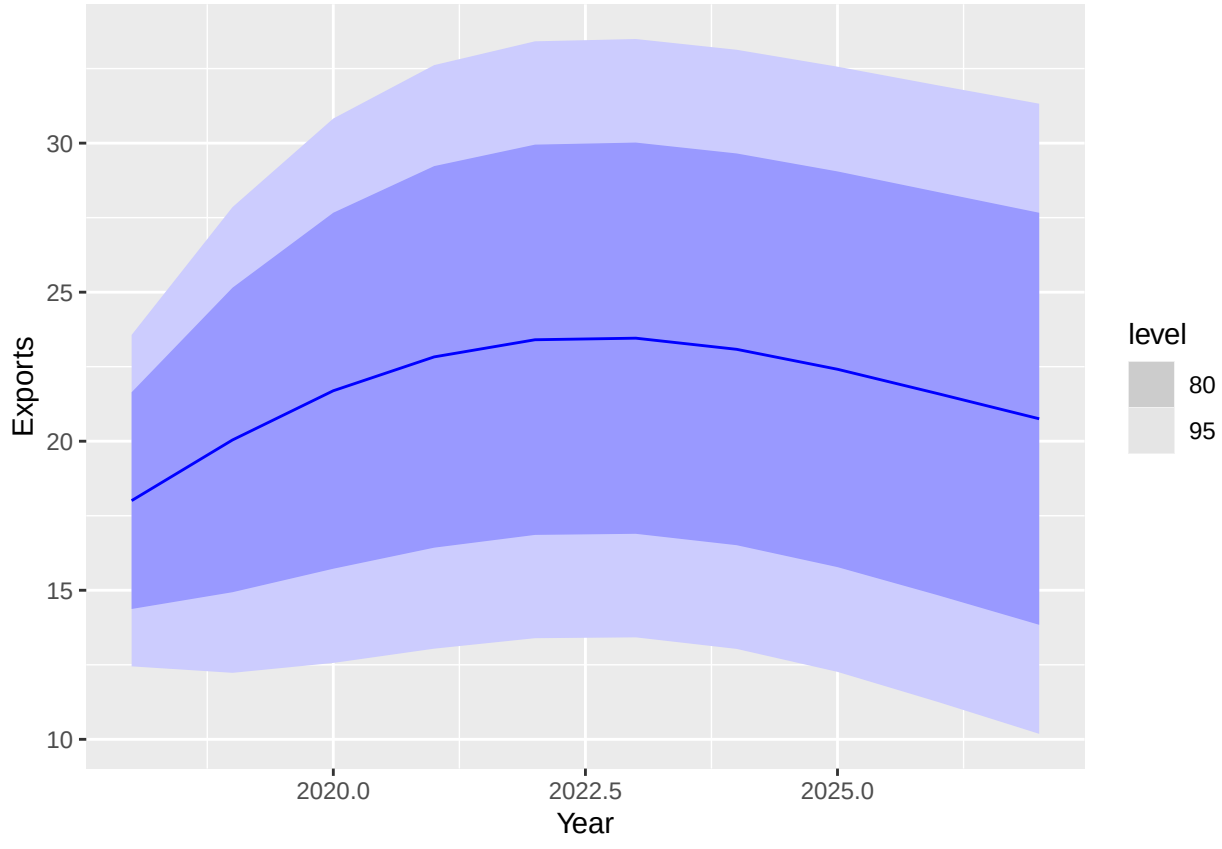


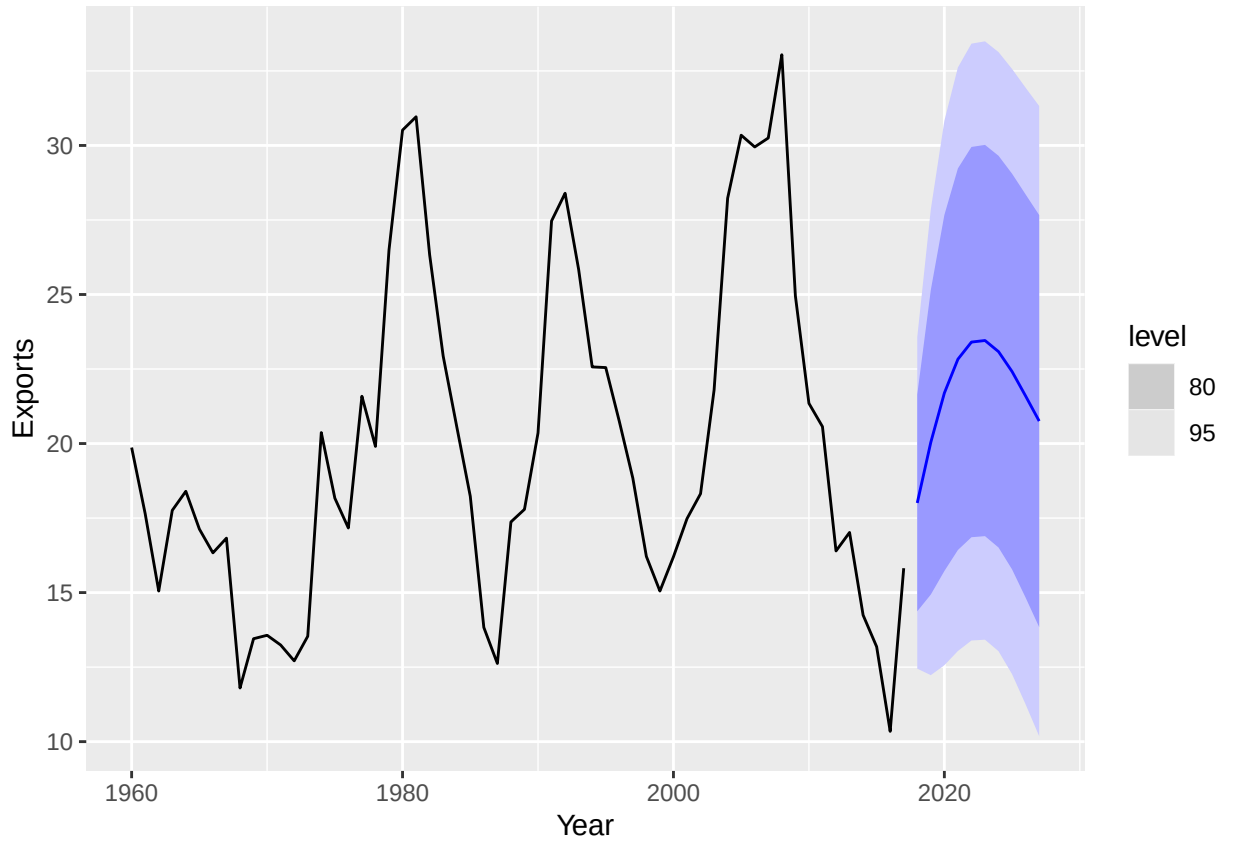

```
## Wytworzony raport za pomocą wbudowanej funkcji report():

## Series: Exports
## Model: ARIMA(2,0,1) w/ mean
##
## Coefficients:
##          ar1      ar2      ma1  constant
##       1.6764  -0.8034  -0.6896    2.5623
## s.e.  0.1111   0.0928   0.1492    0.1161
##
## sigma^2 estimated as 8.046:  log likelihood=-141.57
## AIC=293.13   AICc=294.29   BIC=303.43

## Wytworzona predykcja za pomocą wbudowanej funkcji forecast():

## # A fable: 10 x 5 [1Y]
## # Key:     Country, .model [1]
##    Country          .model          Year   Exports .mean
```

```
##    <fct>            <chr>          <dbl>    <dist> <dbl>
##  1 Egypt, Arab Rep. ARIMA(Exports)  2018  N(18, 8)  18.0
##  2 Egypt, Arab Rep. ARIMA(Exports)  2019 N(20, 16)  20.0
##  3 Egypt, Arab Rep. ARIMA(Exports)  2020 N(22, 22)  21.7
##  4 Egypt, Arab Rep. ARIMA(Exports)  2021 N(23, 25)  22.8
##  5 Egypt, Arab Rep. ARIMA(Exports)  2022 N(23, 26)  23.4
##  6 Egypt, Arab Rep. ARIMA(Exports)  2023 N(23, 26)  23.5
##  7 Egypt, Arab Rep. ARIMA(Exports)  2024 N(23, 26)  23.1
##  8 Egypt, Arab Rep. ARIMA(Exports)  2025 N(22, 27)  22.4
##  9 Egypt, Arab Rep. ARIMA(Exports)  2026 N(22, 28)  21.6
## 10 Egypt, Arab Rep. ARIMA(Exports)  2027 N(21, 29)  20.8
```

```
## Rows: 10
## Columns: 5
## Key: Country, .model [1]
## $ Country <fct> "Egypt, Arab Rep.", "Egypt, Arab Rep.", "Egypt, Arab Rep.", "E~
## $ .model  <chr> "ARIMA(Exports)", "ARIMA(Exports)", "ARIMA(Exports)", "ARIMA(E~
## $ Year    <dbl> 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027
## $ Exports <dist> [<dist_normal[1]>], [<dist_normal[1]>], [<dist_normal[1]>], [<~
## $ .mean   <dbl> 18.00745, 20.04187, 21.69376, 22.82856, 23.40384, 23.45654, 2~
```

```
## Dla $ Exports <dist> mamy do czynienia ze zmienną rozkładu
```





Warto podkreślić, że stała c ma istotny wpływ na prognozy długoterminowe uzyskane z modeli. Niech $p, q = 0$ oraz $\varepsilon_t = 0$. Wtedy model $(1 - \phi_1 B - \ldots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \ldots + \theta_q B^q)\varepsilon_t$ przyjmuje postać $(1 - B)^d y_t = c$. Stąd mamy interpretacje:

i) jeżeli $c = 0 \wedge d = 0$, to $(1 - B)^d y_t = c \implies (1 - B)^0 y_t = 0 \iff y_t = 0$ i długoterminowa prognoza będzie dążyć do zera;

ii) jeżeli $c = 0 \wedge d = 1$, to $(1 - B)^d y_t = c \implies (1 - B)^1 y_t = 0 \iff y_t - y_{t-1} = 0 \iff y_t = y_{t-1}$ i długoterminowa prognoza będzie dążyć stałej niezerowej;

iii) jeżeli $c = 0 \wedge d = 2$, to $(1 - B)^d y_t = c \implies (1 - B)^2 y_t = 0 \iff (1 - 2B + B^2) y_t = 0 \iff y_t - 2y_{t-1} + y_{t-2} = 0 \iff y_t = 2y_{t-1} - y_{t-2}$ i długoterminowa prognoza będzie dążyć do linii prostej;

iv) jeżeli $c \neq 0 \wedge d = 0$, to $(1 - B)^d y_t = c \implies (1 - B)^0 y_t = c \iff y_t = c$ i długoterminowa prognoza będzie dążyć średniej z danych;

v) jeżeli $c \neq 0 \wedge d = 1$, to $(1 - B)^d y_t = c \implies (1 - B)^1 y_t = c \iff y_t - y_{t-1} = c \iff y_t = c + y_{t-1}$ i długoterminowa prognoza będzie dążyć do linii prostej;

vi) jeżeli $c \neq 0 \wedge d = 2$, to $(1 - B)^d y_t = c \implies (1 - B)^2 y_t = c \iff (1 - 2B + B^2) y_t = c \iff y_t - 2y_{t-1} + y_{t-2} = c \iff y_t = c + 2y_{t-1} - y_{t-2}$ i długoterminowa prognoza będzie dążyć do trendu kwadratowego.

ACF (autocorrelations funtion), a PCF (partial autocorrelations function)- szukanie p,q

Wykres ACF pokazuje autokorelacje, które mierzą związek między y_t i y_{t-k} dla różnych wartości k. Jeżeli y_t i y_{t-1} są skorelowane, to y_{t-1} i y_{t-2} również będą skorelowane. Jednakże, wtedy również y_t i y_{t-2} mogą

być skorelowane, z uwagi na to, że oba są połączone z y_{t-1}, a nie z powodu jakichkolwiek nowych informacji zawartych w y_{t-2}, które mogłyby być wykorzystane do prognozowania y_t.

Wykres PACF pokazuje pomiar związku pomiędzy y_t i y_{t-k} po ussuniciu wpływu "lagów" $1, 2, 3, \ldots, k-1$. Każda autokorelacja cząstkowa może być oszacowana jako ostatni współczynnik w modelu autoregresyjnym.

Jeżeli mamy do czynienia z modelem ARIMA(p,d,0) lub ARIMA(0,d,q), to wykresy ACF i PACF mogą okazać się przydatne w określeniu wartości p lub q. Jednakże, jeżeli p,q > 0, to wtedy wykresy nie pomagają w znalezieniu odpowiednich wartości.

Jeżeli wykresy ACF i PACF wykazuja następujące zachowania:

i) ACF jest zanikający wykładniczo lub sinusoidalnie;

ii) występuje istotny statystycznie "peak" dla lag-u p w PACF, ale nigdzie poza nim;

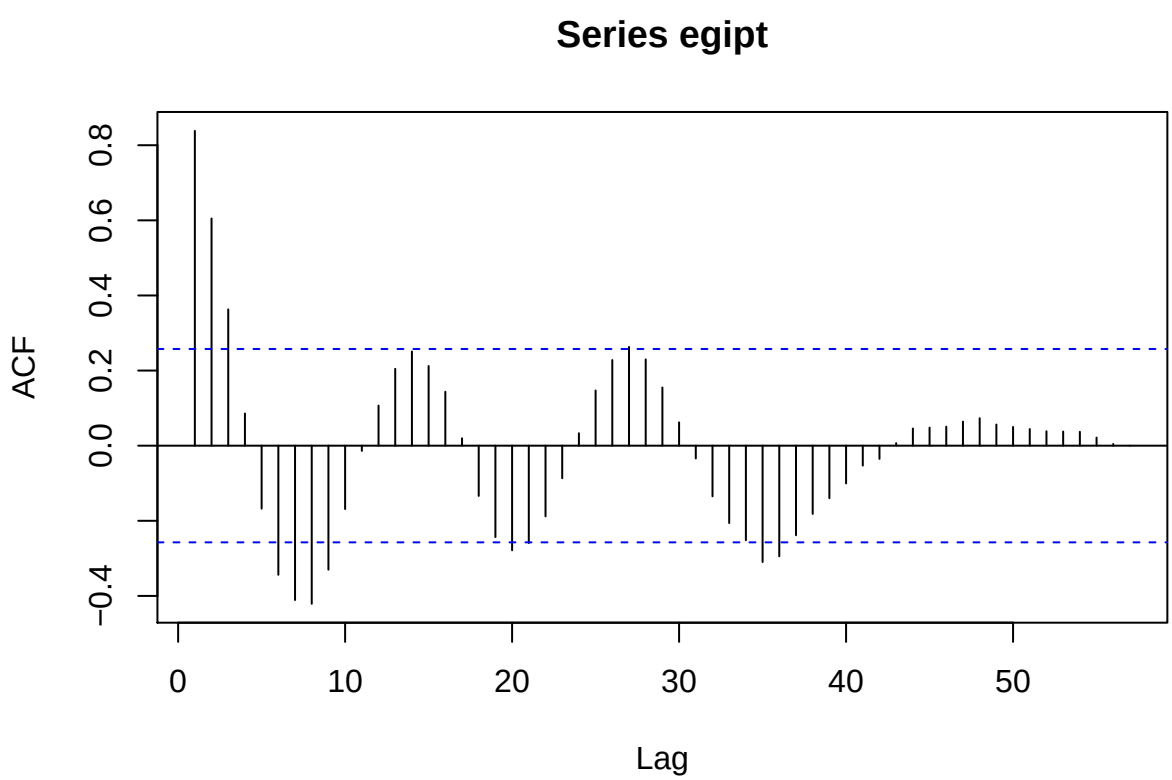
,to dane mogą dążyć do postaci modelu ARIMA(p,d,0),

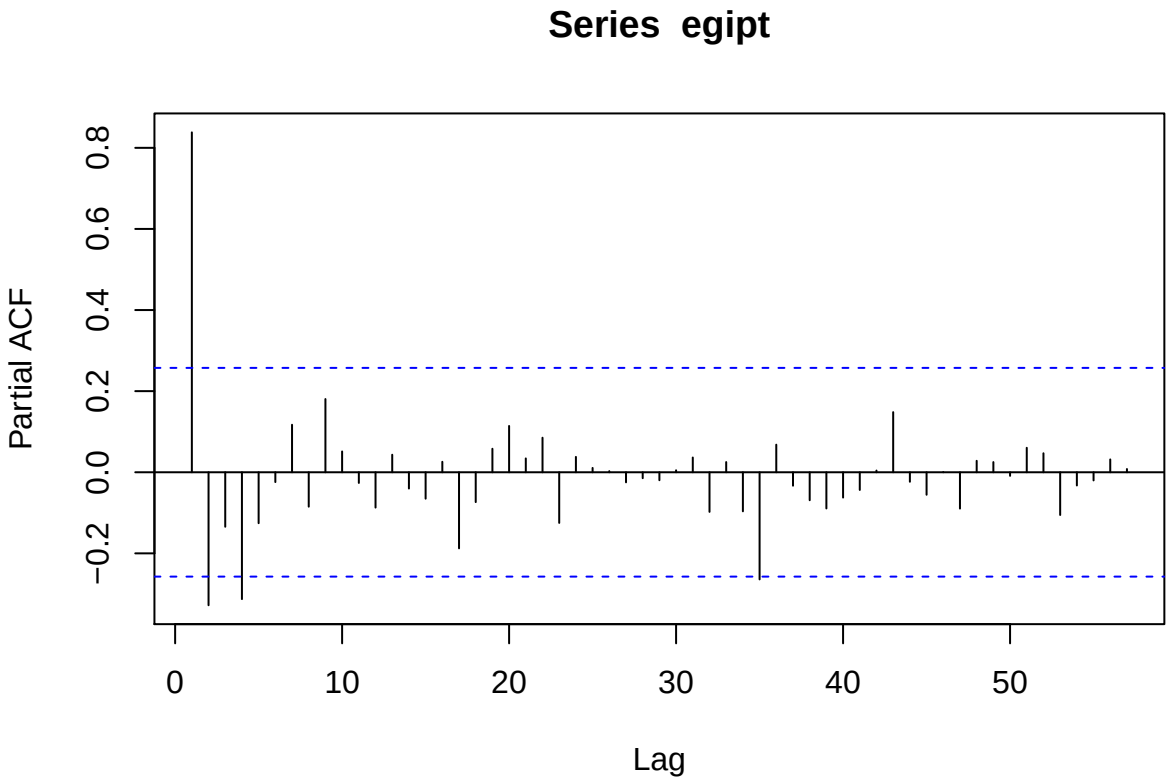
Analogicznie:

i) PACF jest zanikający wykładniczo lub sinusoidalnie;

ii) występuje istotny statystycznie "peak" dla lag-u q w ACF, ale nigdzie poza nim;

,to dane mogą dążyć do postaci modelu ARIMA(0,d,q),





Dla ACF możemy zaobserwować zanikający sinusoidalnie wzór. Natomiast PACF wykazuje istotny "peak" dla lag-u 4 (model dla lag-u 2 powstał wcześniej). Utwórzmy zatem model ARIMA(4,0,0)

```
## Series: Exports
## Model: ARIMA(4,0,0) w/ mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4  constant
##       0.9861  -0.1715  0.1807  -0.3283    6.6922
## s.e.  0.1247   0.1865  0.1865   0.1273    0.3562
##
## sigma^2 estimated as 7.885:  log likelihood=-140.53
## AIC=293.05   AICc=294.7   BIC=305.41
```

Ten model jest tylko trochę gorszy od modelu ARIMA(2,0,1) (,który wykazał AICc równy 294.29- im mniejszy wynik tym lepszy model).

Dodatkowo, korzystając z funkcji ARIMA(), możemy określić konkretne wartości pdq(), dla których chcemy utowrzyć model ARIMA().

```
## Series: Exports
## Model: ARIMA(2,0,1) w/ mean
##
## Coefficients:
##          ar1      ar2      ma1  constant
##       1.6764  -0.8034  -0.6896    2.5623
## s.e.  0.1111   0.0928   0.1492    0.1161
##
```

```
## sigma^2 estimated as 8.046:  log likelihood=-141.57
## AIC=293.13   AICc=294.29   BIC=303.43
```

MLE, AIC, AICc, BIC, czyli ocena skuteczności modelu

Po ustaleniu parametrów p, d, q, należy wyestymować zmienne $c, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$. Bilbioteka "fable", podczas estymacji tych zmiennych dla modelu ARIMA(), domyślnie korzysta z algorytmu MLE (maximum likelihood estimation) do maksymalizacji prawdopodobieństwa uzyskania danych, które zaobserwowaliśmy.

Należy pamiętać, że modele ARIMA są znacznie bardziej skomplikowane do oszacowania niż modele regresji, a różne programy dadzą nieco inne odpowiedzi, ponieważ używają różnych metod estymacji i różnych algorytmów optymalizacji.

Pakiet "fable" zawsze zwraca wartość z logarytmu wiarygodności danych $log(L)$, gdzie $L : \mathbb{R}^n \text{ x } \Theta \to [0, \infty)$ jest funkcją wiarygodności daną wzorem

$$L(x_1, \ldots, x_n; \theta) = p_\theta(x_1; \theta) \ldots p_\theta(x_n; \theta), \forall (x_1, \ldots, x_n) \in \mathbb{R}^n$$

Dla danych wartości p, d, q, funkcja ARIMA() będzie starać się maksymalizować $log(L)$ podczas estymowania wartości zmiennych $c, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$.

Trzema domyślnymi kryteriami informacyjnymi w ramach biblioteki "fable" są AIC (Akaike's Information Criterion), AICc (Akaike's Information Criterion corrected) oraz BIC (Bayesian Information Criterion). W celu uzyskania dobrego modelu, staramy się minimalizować powyższe kryteria (dla modelu ARIMA() głównie skupiamy się na AICc).

Uwaga! kryteria informacyjne służą głównie do porównania modelu dla danych wartości p, q (dążymy do minimalizacji wartości AIC, AICc, BIC, dla wybranych p, q). Dzieje się tak dlatego, że różnicowanie zmienia dane, na których obliczane jest prawdopodobieństwo, przez co wartości AIC pomiędzy modelami z różnymi rzędami różnicowania nie są porównywalne.

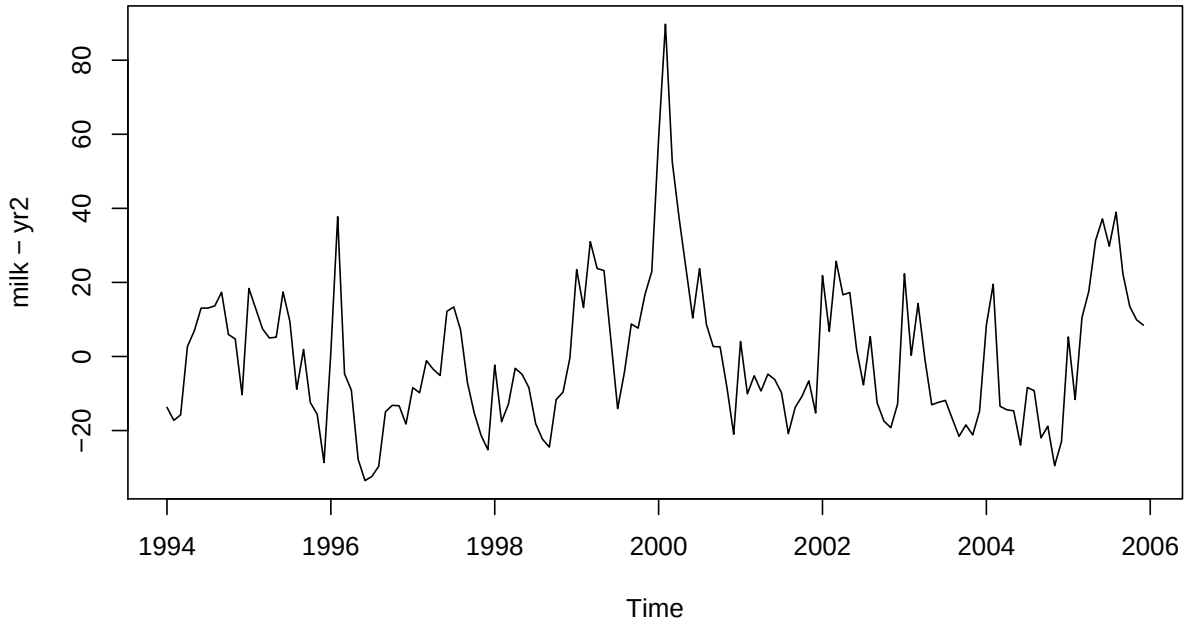
Działanie funkcji ARIMA()

Funkcja ARIMA() z biblioteki "fable" wykorzystuje wariacje algorytmu Hyndman-Khandakar, który łączy w sobie test KPSS wraz z minimalizacją AICc i MLE, aby uzyskać model ARIMA. Dla zainteresowanych, w książce Hyndman-a, w rozdziale 9.7 "ARIMA modelling in fable", autor graficznie przedstawił schemat działania tegoż algorytmu.

Podczas dopasowywania modelu ARIMA do (niesezonowego) szeregu czasowego, klasycznym podejściem jest:

1) Narysowanie wykresu i identyfikacja nietypowych obserwacji.
2) Jeżeli konieczna, transformacja danych dla ustabilizowania wariancji (transoframcja Box'a-Cox'a).
3) Jeżeli szereg jest niestacjonarny, to identyfikacja sezonowości lub pierwsze różnicowanie danych (oba do momentu uzyskania szeregu stacjonarnego).
4) Badanie wykresów ACF/PACF w celu stwierdzenia jaki model będzie odpowiedni (ARIMA(p,d,0) lub ARIMA(0,d,q)).
5) Próba dobrania najlepszego modelu pod względem minimalizacji AICc.
6) Sprawdzenie residuów z wybranego modelu poprzez narysowanie wykresu ACF.
7) Prognoza, pod warunkiem, że residua przypominają biały szum.

Algorytm Hyndman-Khandakar automatyzuje kroki 3-5. Dla przykładu posłużymy się zbiorem 'milk':



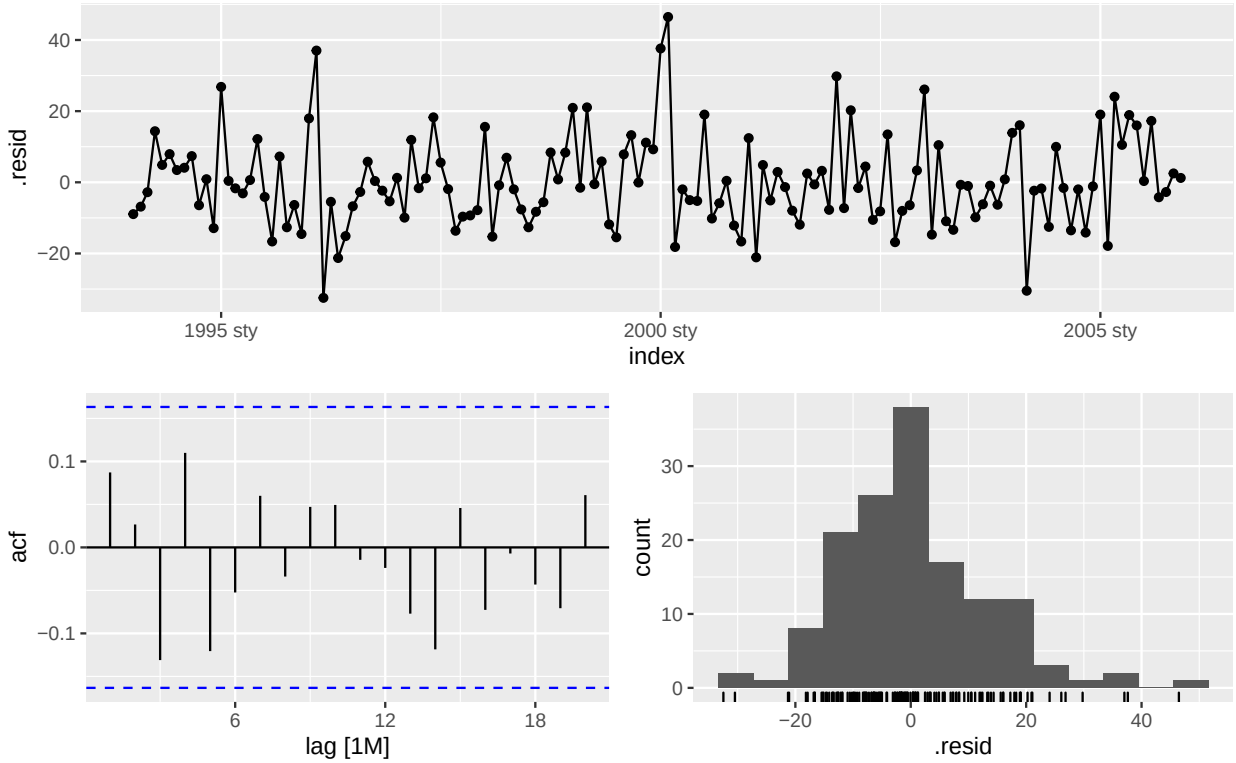

```
## Żadna transformacja nie jest wymagana

## Series: value
## Model: ARIMA(1,0,0)(1,0,0)[12]
##
## Coefficients:
##          ar1    sar1
##       0.7471  0.1748
## s.e.  0.0549  0.0846
##
## sigma^2 estimated as 165.5:  log likelihood=-571.77
## AIC=1149.53   AICc=1149.71   BIC=1158.44

## otrzymaliśmy model ARIMA(1,0,0), tj. Autoregresję dla p= 1

## Próba uzyskania lepszego modelu

## Warning in sqrt(diag(best$var.coef)): wyprodukowano wartości NaN

## Series: value
## Model: ARIMA(2,0,0)(2,0,1)[12]
##
## Coefficients:
##          ar1     ar2    sar1    sar2     sma1
##       0.7493  0.0418  0.4909  0.1660  -0.4275
## s.e.     NaN  0.0038     NaN  0.0038      NaN
##
## sigma^2 estimated as 160.3:  log likelihood=-568.42
## AIC=1148.84   AICc=1149.45   BIC=1166.66

## Okazuję się, że można również zrobić to krócej
```

```
## Warning in sqrt(diag(best$var.coef)): wyprodukowano wartości NaN
```

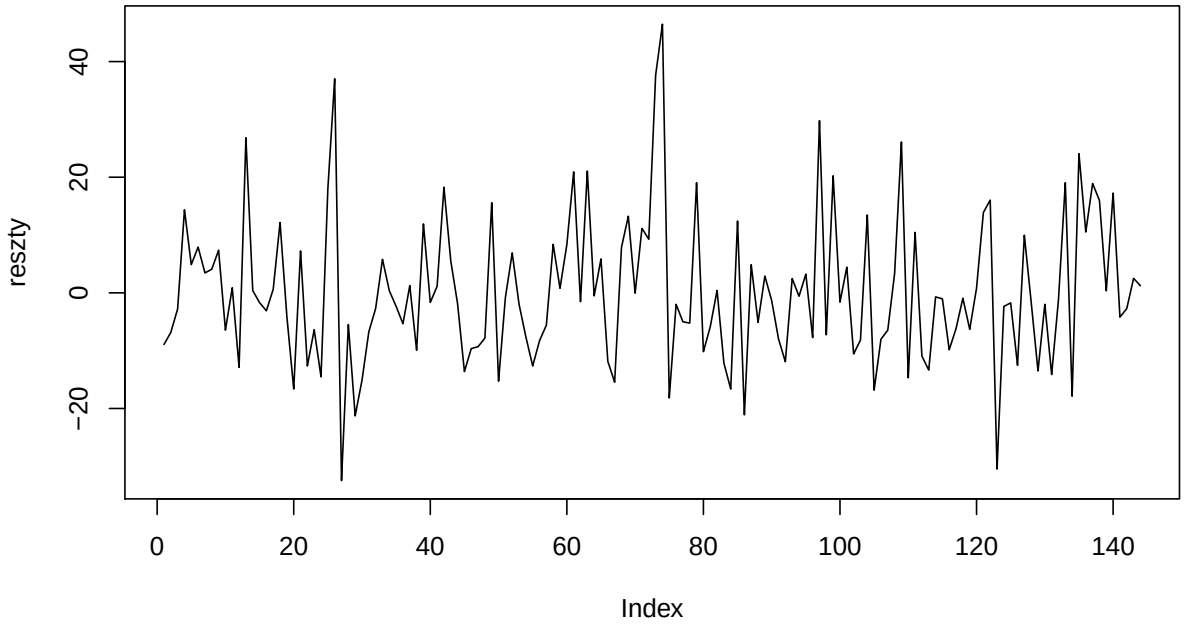
```
## # A tibble: 2 x 8
##   .model sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 res1     166.   -572. 1150. 1150. 1158. <cpl [13]> <cpl [0]>
## 2 res2     160.   -568. 1149. 1149. 1167. <cpl [26]> <cpl [12]>
```

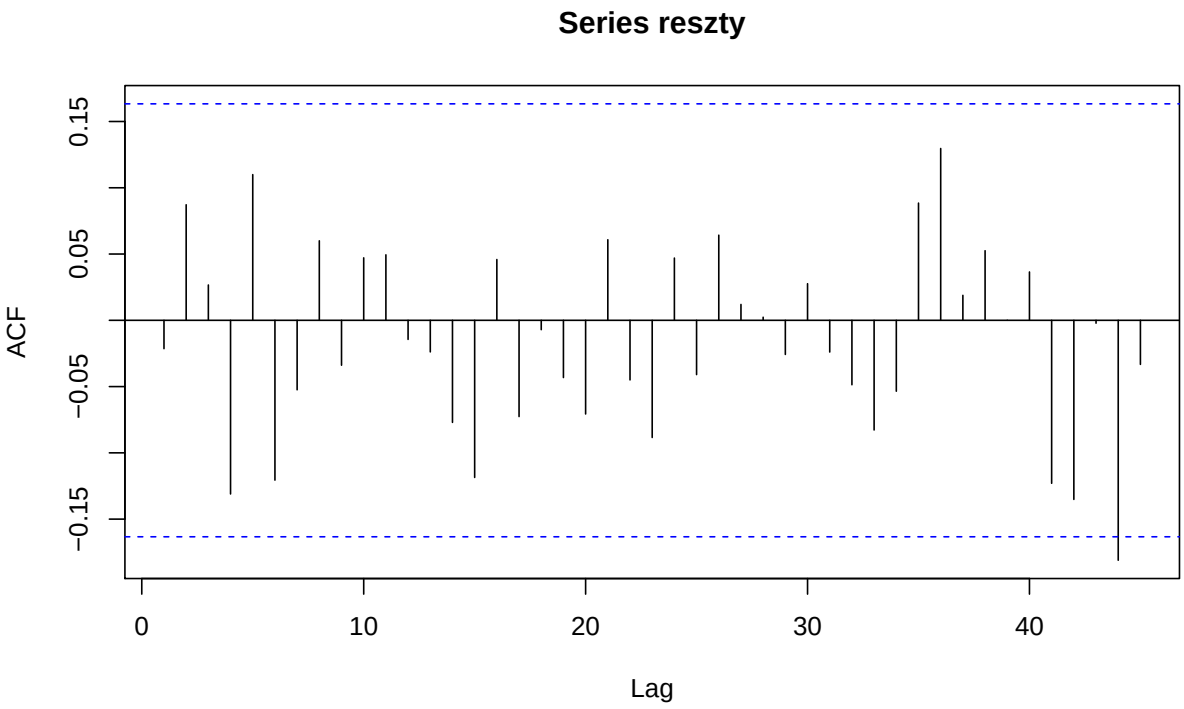
```
## Z powyższej tabeli można stwierdzić, że model ARIMA(1,0,0) był lepiej dopasowany
```

```
## Wykorzystanie funkcji gg_tsresiduals() do zbadania residuów
```

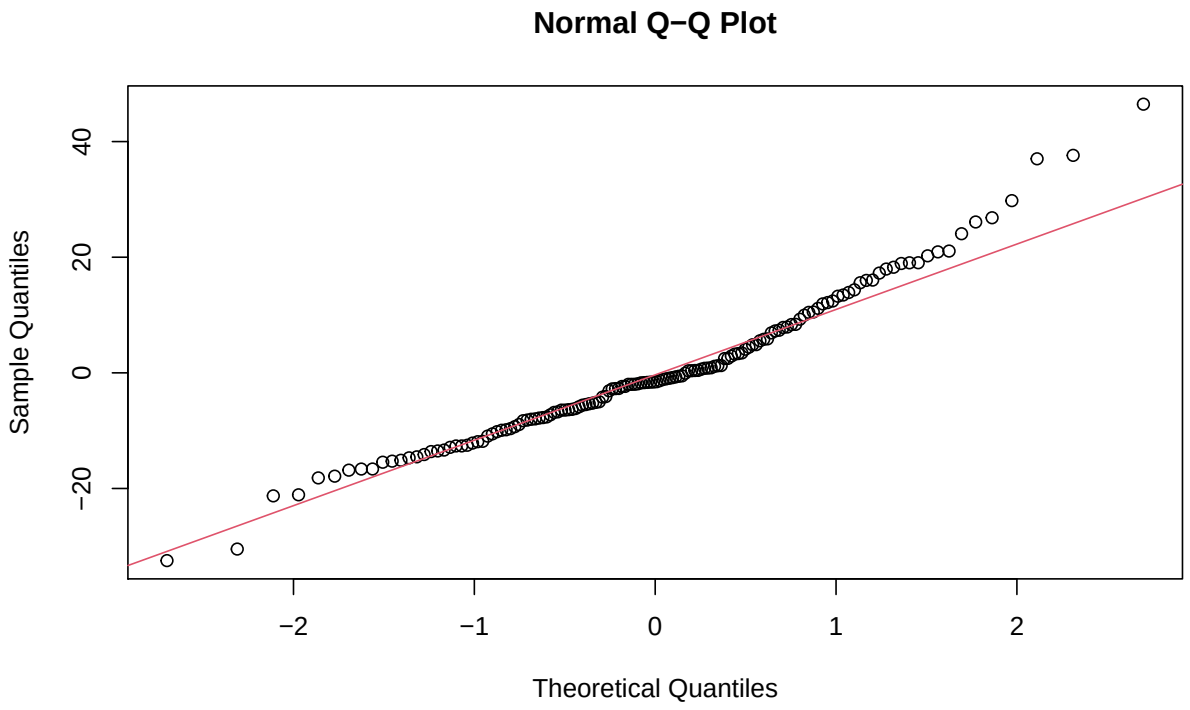



```
## Oczywiście możemy również wydobyć wartości residuów ręcznie poprzez funkcję stats::residuals()
```

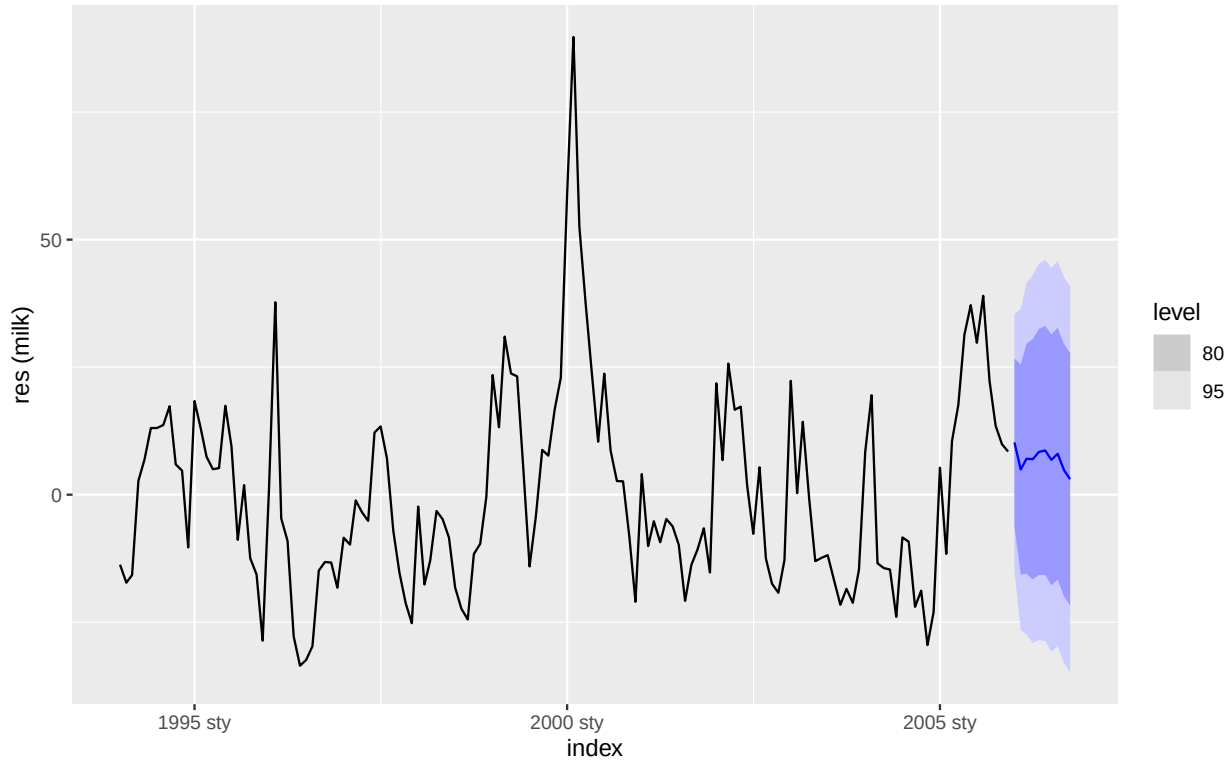


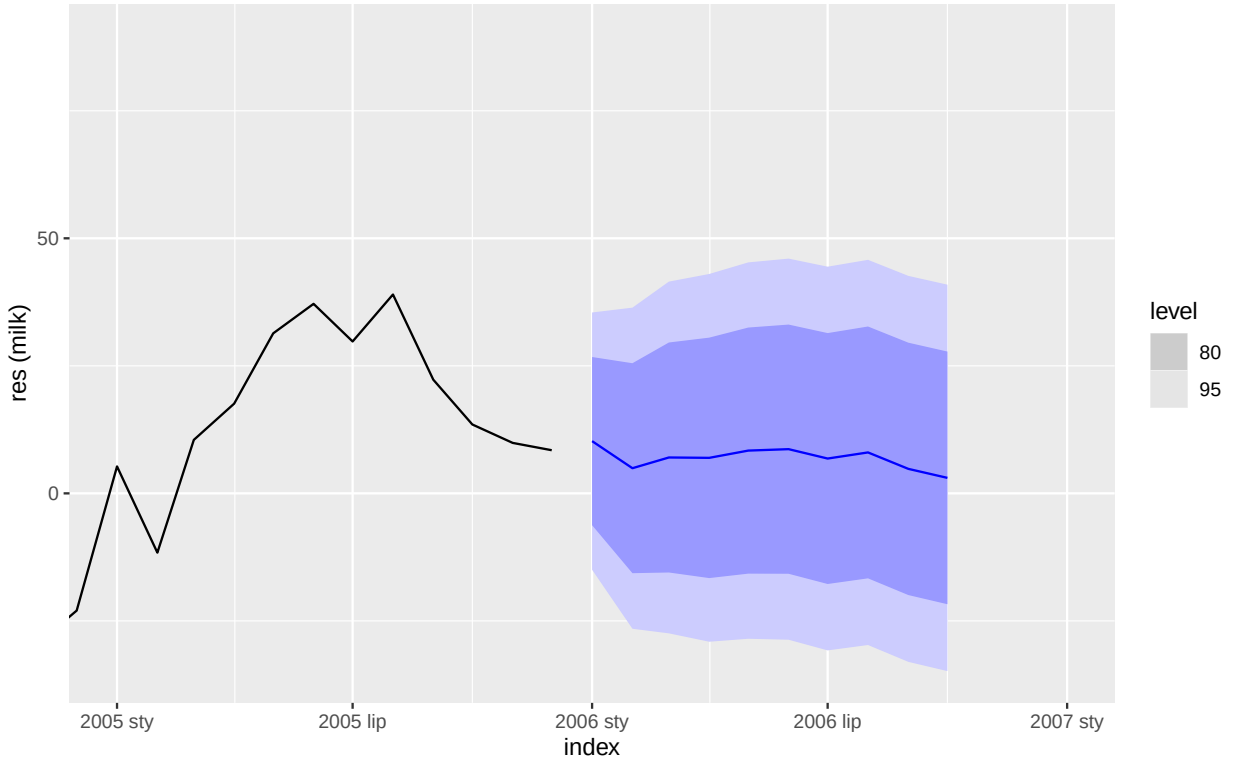



```
## Przypominają one biały szum
## Możemy dodatkowo sprawdzić czy reszty mają rozkład normalny
```




```
## Zatem możemy przejść do predykcji
```





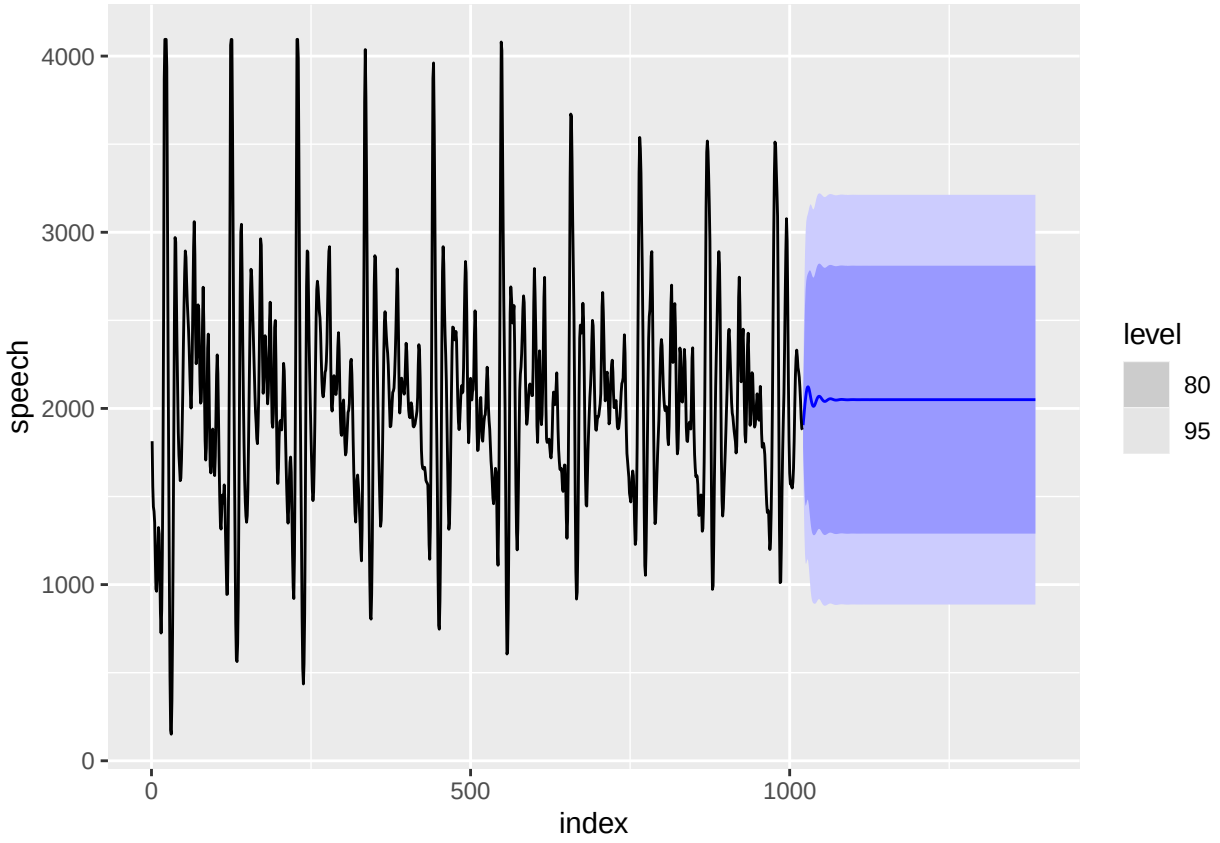
Domyślnie, funkcja ARIMA() automatycznie określam czy stała jest wymagana. Dla d= 0 lub d= 1, stała zostanie uwzględniona, jeśli poprawi to wartość AICc. Jeżeli $d > 1$ stała jest zawsze pomijana, ponieważ trend kwadratowy lub wyższego rzędu jest szczególnie niebezpieczny przy prognozowaniu. Oczywiście możemy zmusić model do uwzględnienia bądź zignorowania stałej.

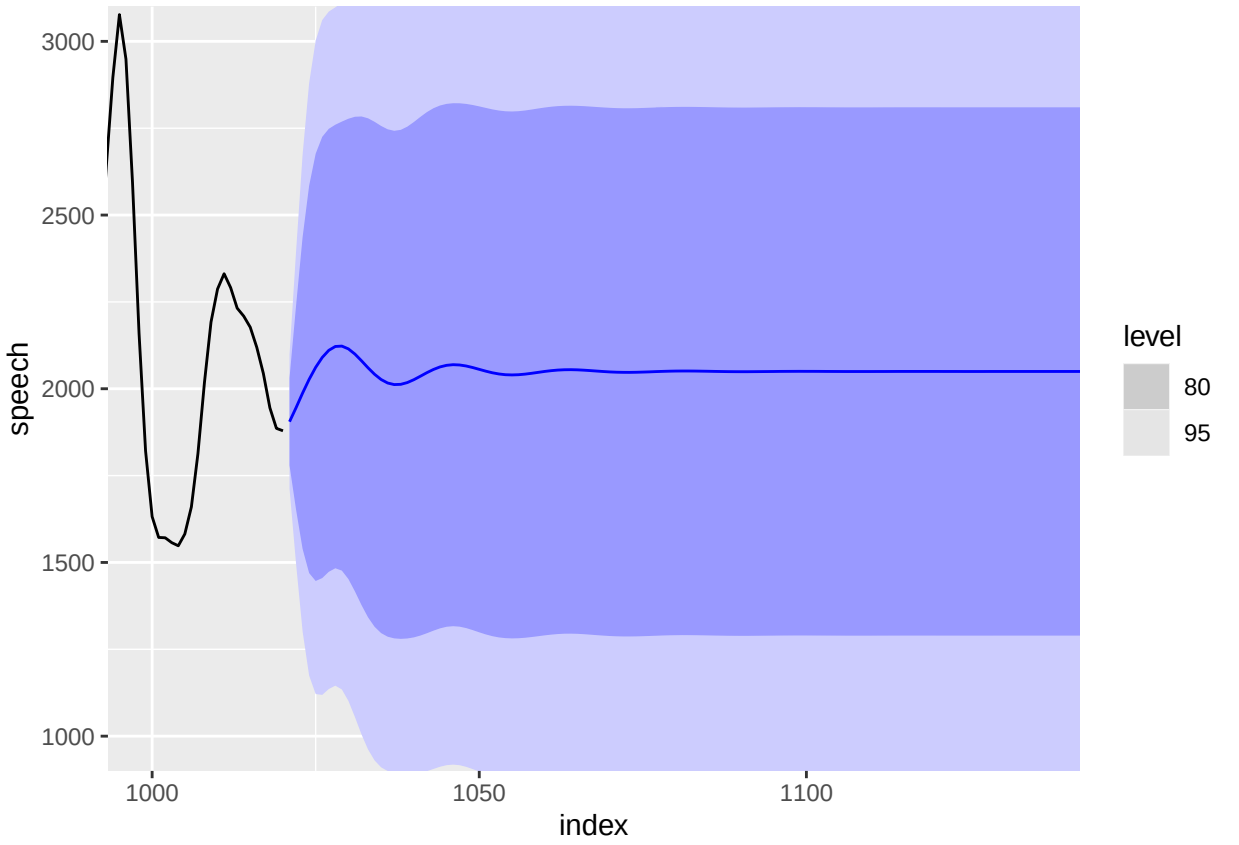
```
## ARIMA(value ~ 1 + ... ,gdzie 1 oznacza wymuszenie uwzględnienia stałej

## Series: value
## Model: ARIMA(1,0,0)(1,0,0)[12] w/ mean
##
## Coefficients:
##          ar1    sar1  constant
##       0.7471  0.1749    0.0457
## s.e.  0.0549  0.0846    1.0268
##
## sigma^2 estimated as 166.7:  log likelihood=-571.77
## AIC=1151.53   AICc=1151.82   BIC=1163.41

## ARIMA(value ~ 0 + ... ,gdzie 0 oznacza wymuszenie zignorowania stałej

## Series: value
## Model: ARIMA(1,0,0)(1,0,0)[12]
##
## Coefficients:
##          ar1    sar1
##       0.7471  0.1748
## s.e.  0.0549  0.0846
##
## sigma^2 estimated as 165.5:  log likelihood=-571.77
## AIC=1149.53   AICc=1149.71   BIC=1158.44
```

Uwaga! Przedziały ufności dla modeli ARIMA opierają się na założeniu, że reszty są nieskorelowane i mają rozkład normalny. Jeżeli którekolwiek z tych założeń nie jest spełnione, wówczas przedziały predykcji mogą być nieprawidłowe. Z tego powodu, zawsze należy sporządzić wykres ACF i histogram reszt w celu sprawdzenia założeń przed sporządzeniem przedziałów ufności.

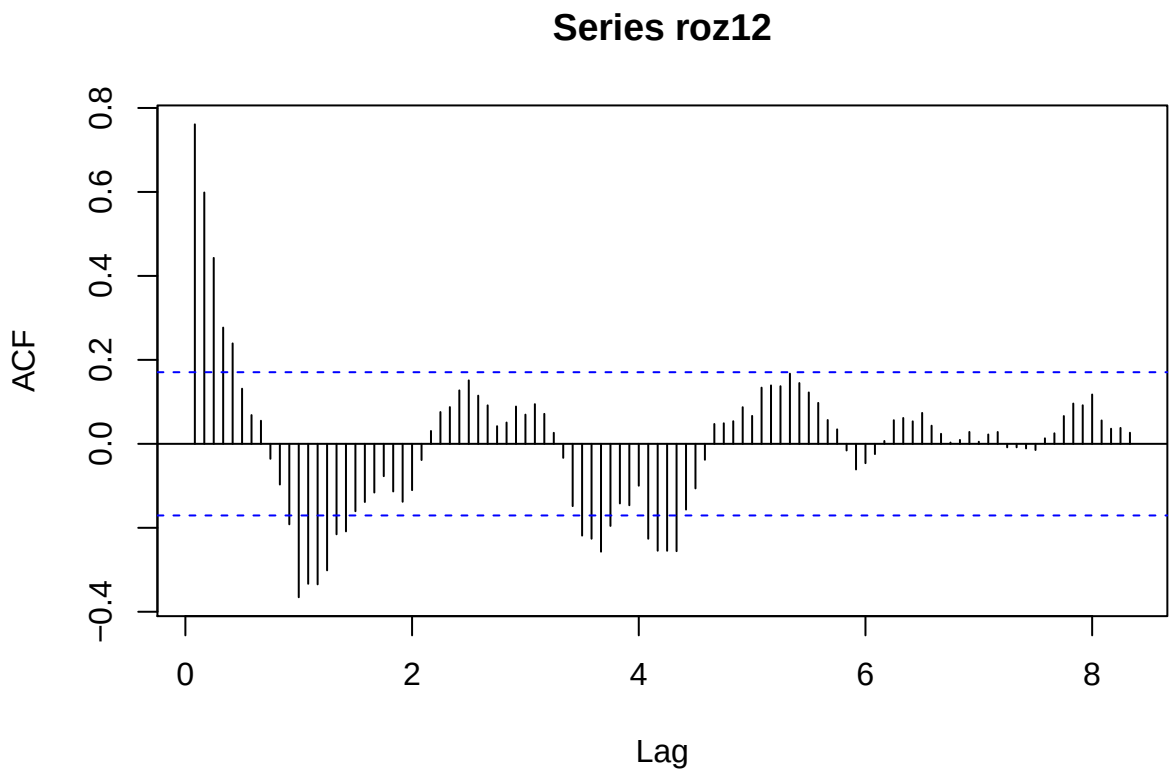
Jeżeli reszty są nieskorelowane, ale nie mają rozkładu normalnego, to zamiast tego można zastosować Bootstrap (wprowadzone przez Bradleya Efrona metody szacowania rozkładu błędów estymacji, za pomocą wielokrotnego losowania ze zwracaniem z próby. Są przydatne szczególnie, gdy nie jest znana postać rozkładu zmiennej w populacji). Wystarczy do funkcji forecast() dodać "bootstrap=TRUE".

```
## Series: value
## Model: ARIMA(4,0,1) w/ mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1  constant
##       2.6538  -2.8889  1.5474  -0.3758  -0.5468  130.1641
## s.e.  0.0742   0.1523  0.1162   0.0336   0.0755    1.3890
##
## sigma^2 estimated as 9620:  log likelihood=-6124.62
## AIC=12263.24   AICc=12263.35   BIC=12297.73
```

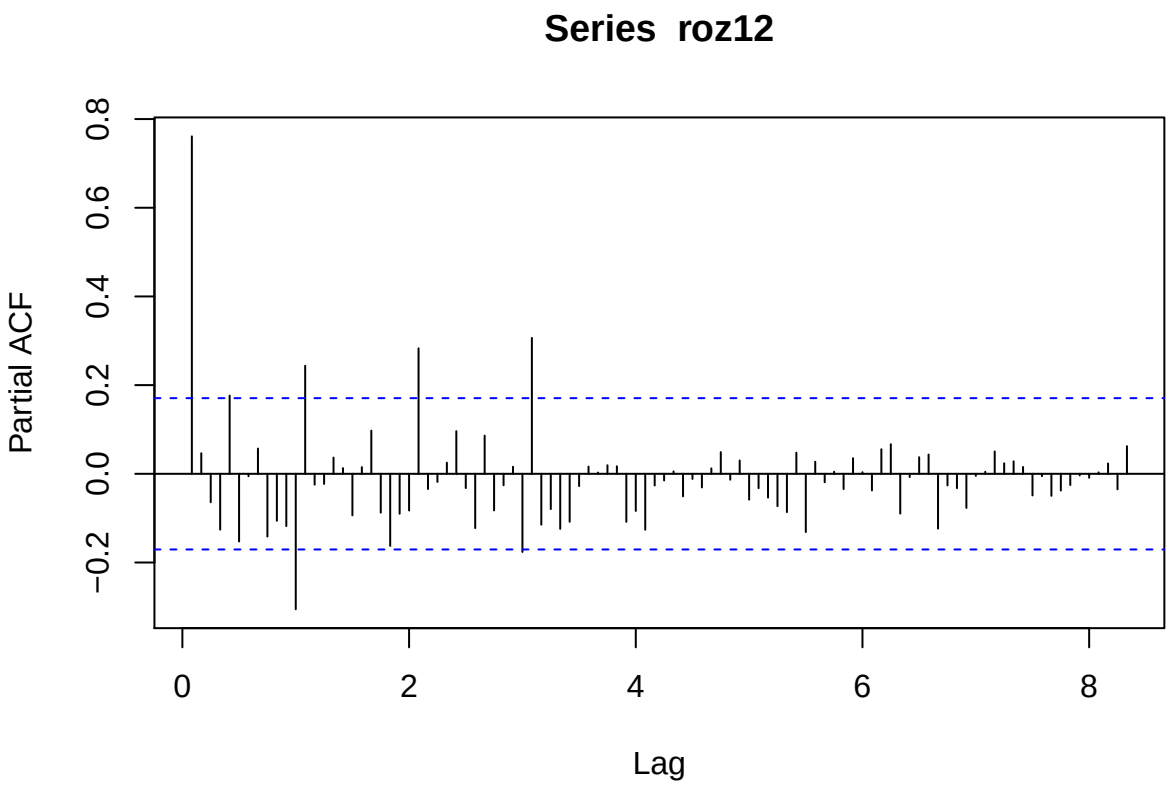





```
acf(roz12, 100)
```



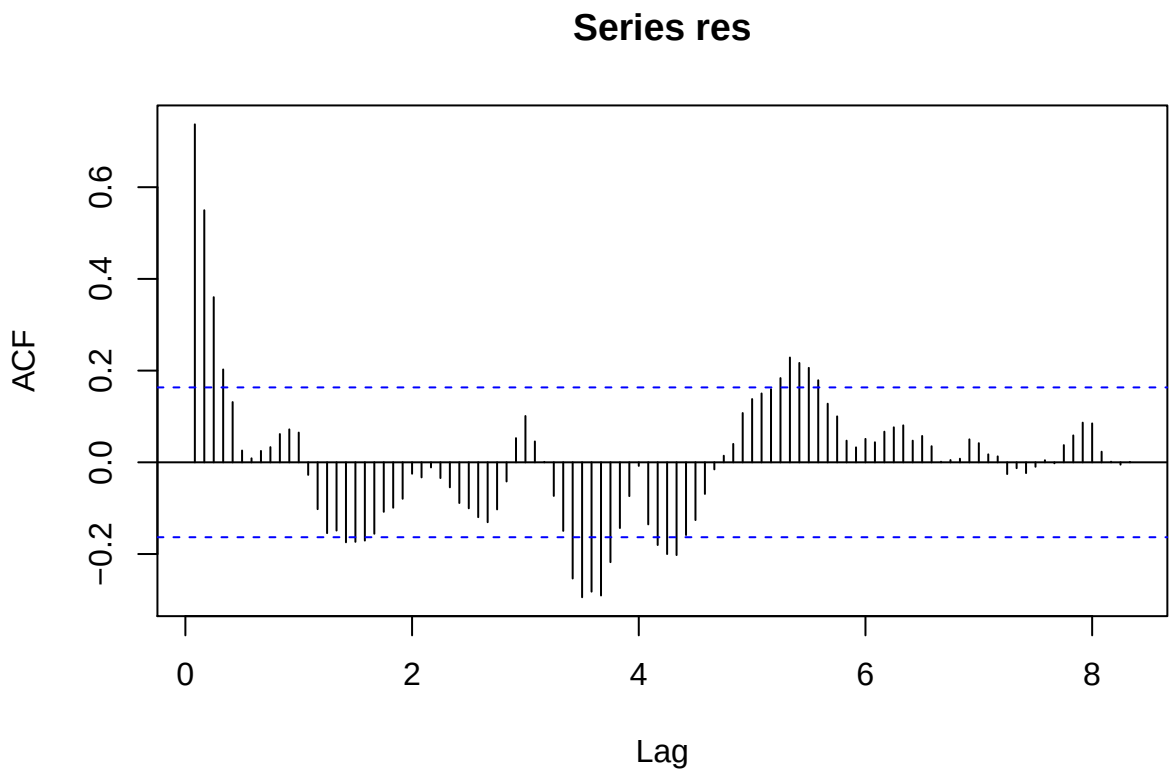

```
pacf(roz12, 100)
```



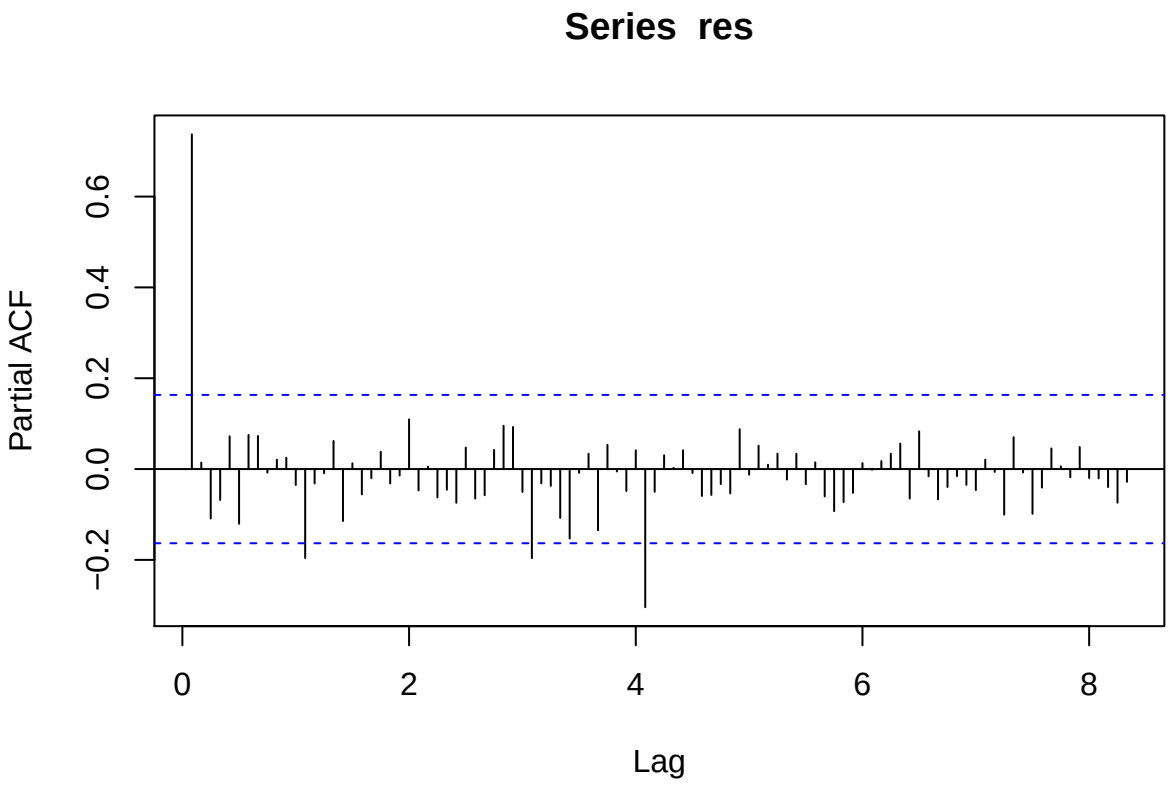

```
fit <- as_tsibble(milk) %>%  model(ARIMA(value ~ pdq(p=0:8, d=0:2, q=0:8)))
report(fit)
```

```
## Series: value
## Model: ARIMA(1,0,0)(2,1,2)[12] w/ drift
##
## Coefficients:
##          ar1    sar1     sar2     sma1    sma2  constant
##       0.8638  0.0607  -0.4074  -1.0121  0.4831    4.8169
## s.e.  0.0475  0.1862   0.1173   0.1994  0.1881    0.4785
##
## sigma^2 estimated as 137.9:  log likelihood=-518.84
## AIC=1051.67   AICc=1052.57   BIC=1071.85
```

```
acf(res, 100)
```




```
pacf(res, 100)
```


Series res
Partial ACF
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
Lag