

Diagnostyka białego szumu i metody dekompozycji

Analiza szeregów czasowych

Zuzanna Klamana

2026-02-17

Spis treści

1	Testowanie białoszumowości	2
1.1	Porównanie testu graficznego oraz formalnych testów statystycznych	2
1.2	Symulacyjne porównanie białoszumowości	10
2	Dekompozycja szeregów czasowych - eliminacja trendu i sezonowości	14
2.1	Krótki opis zagadnienia:	14
2.2	Opis eksperymentów/analiz	14
2.3	Wyniki	15

1 Testowanie białoszumowości

1.1 Porównanie testu graficznego oraz formalnych testów statystycznych

1.1.1 Krótki opis zagadnienia:

Analizujemy sztucznie wygenerowane szeregi czasowe, takie jak:

- Szum IID,
- Model ruchomej średniej rzędu 1,
- Błądzenie losowe,
- Szum IID z trendem liniowym,
- Biały szum o rozkładzie wykładniczym.

Celem jest zilustrowanie działania testu graficznego opartego na funkcji autokorelacji oraz wybranych formalnych testów statystycznych służących do weryfikacji hipotezy, że obserwowany szereg czasowy jest procesem białego szumu. Test przeprowadzamy na asymptotycznym poziomie istotności równym α .

1.1.2 Opis eksperymentów/analiz

Przebieg analizy można podzielić na dwa etapy:

- Przeprowadzenie testu graficznego opartego na funkcji autokorelacji, porównującego autokorelacje próbkowe z asymptotycznymi granicami istotności wynikającymi z własności białego szumu.
- Przeprowadzenie formalnych testów statystycznych przy wykorzystaniu statystyk testowych Boxa-Pierce'a czy Ljungiego-Boxa.

1.1.3 Wyniki

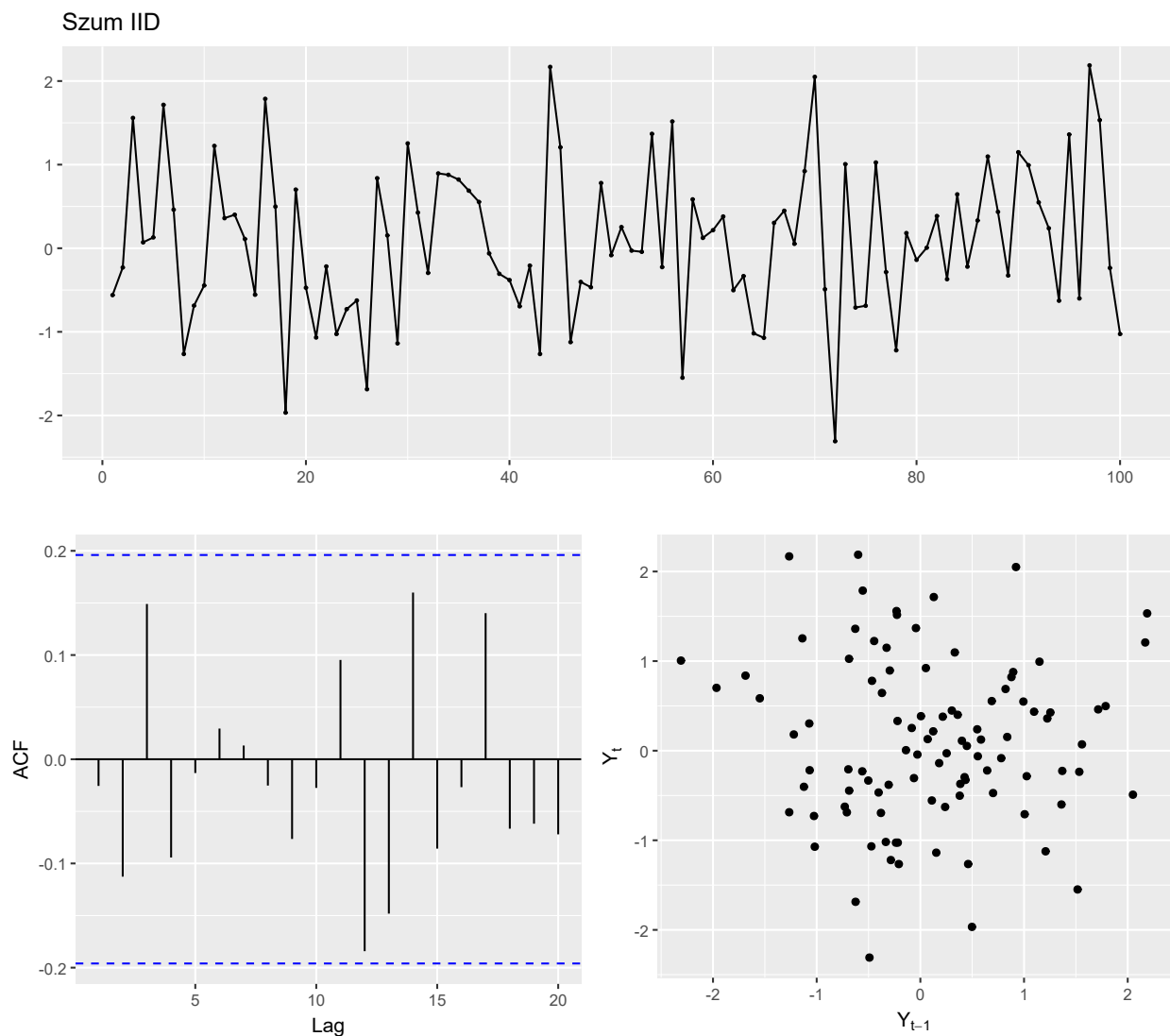
Test graficzny oparty na funkcji autokorelacji (ACF)

Reguła identyfikacji modelu nazywana również graficznym testem białoszumowości mówi, że:

Szereg możemy uznać za realizację białego szumu, jeżeli:

- Co najmniej 95 autokorelacji próbkowych znajduje się w przedziale $\pm \frac{1,96}{\sqrt{n}}$,
- Nie powinno być autokorelacji “istotnie” wychodzących poza przedziały ufności $\pm \frac{1,96}{\sqrt{n}}$.

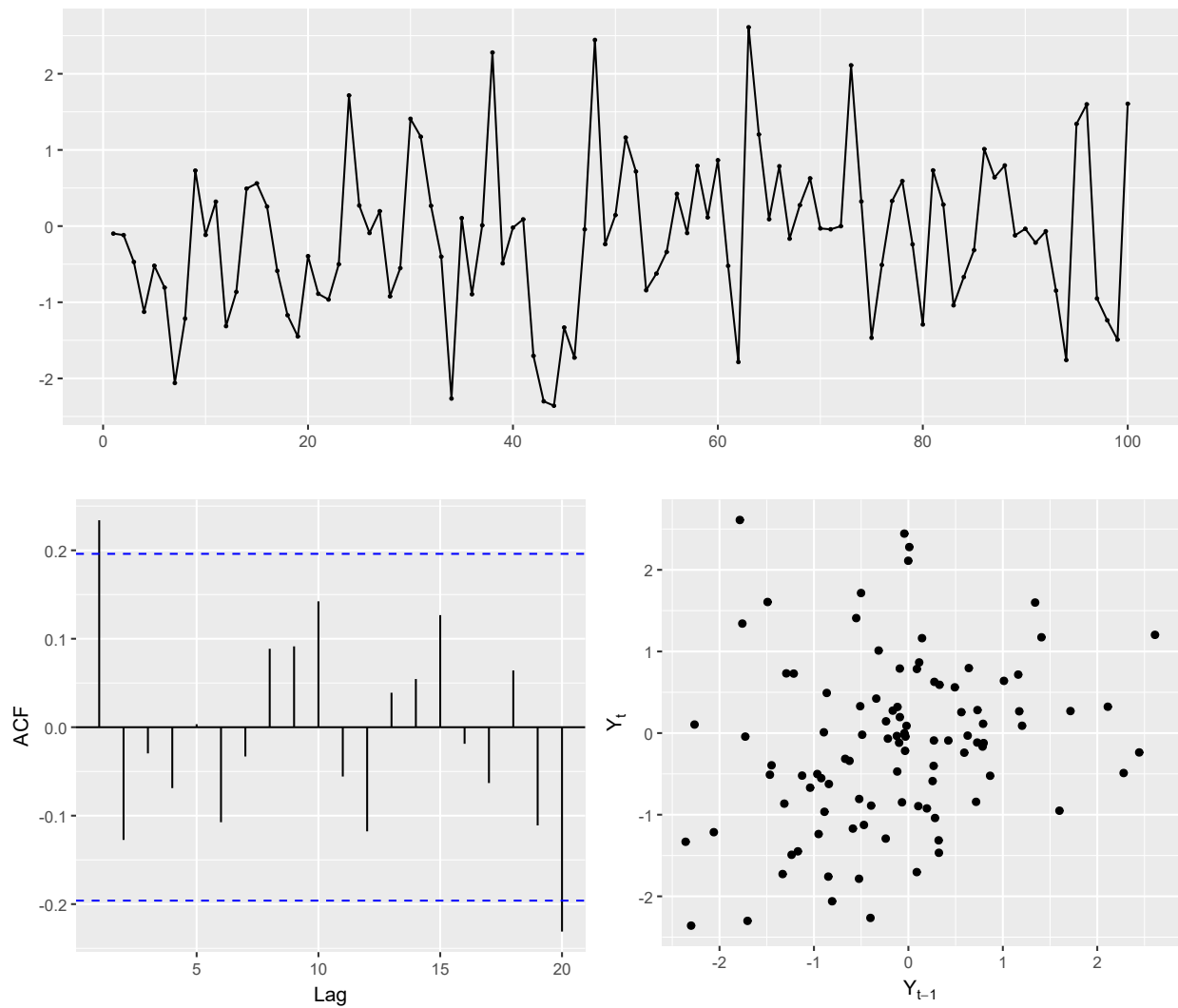
Do przeprowadzenia testu graficznego na sztucznie wygenerowanych szeregach czasowych, wykorzystamy funkcję `ggtsdisplay`, która wyświetla wykres realizacji szeregu, wykres ACF z przedziałami ufności oraz wykres rozrzutu między dwoma kolejnymi obserwacjami szeregu.



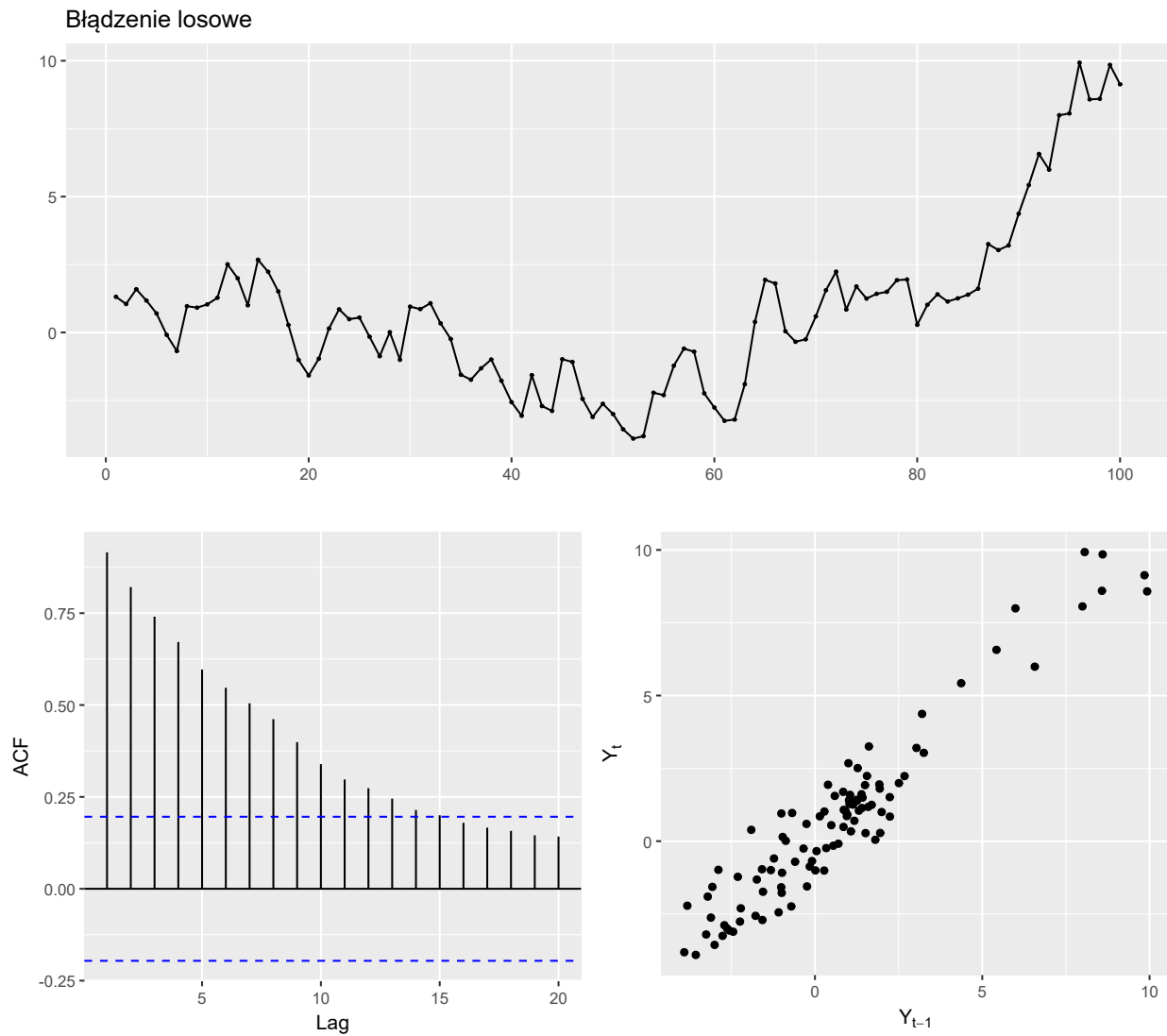
Na podstawie wykresu widać, że Szum IID jest białym szumem, wszystkie autokorelacje

mieszczą się w pasmach ufności, a ich wartości są bliskie zero.

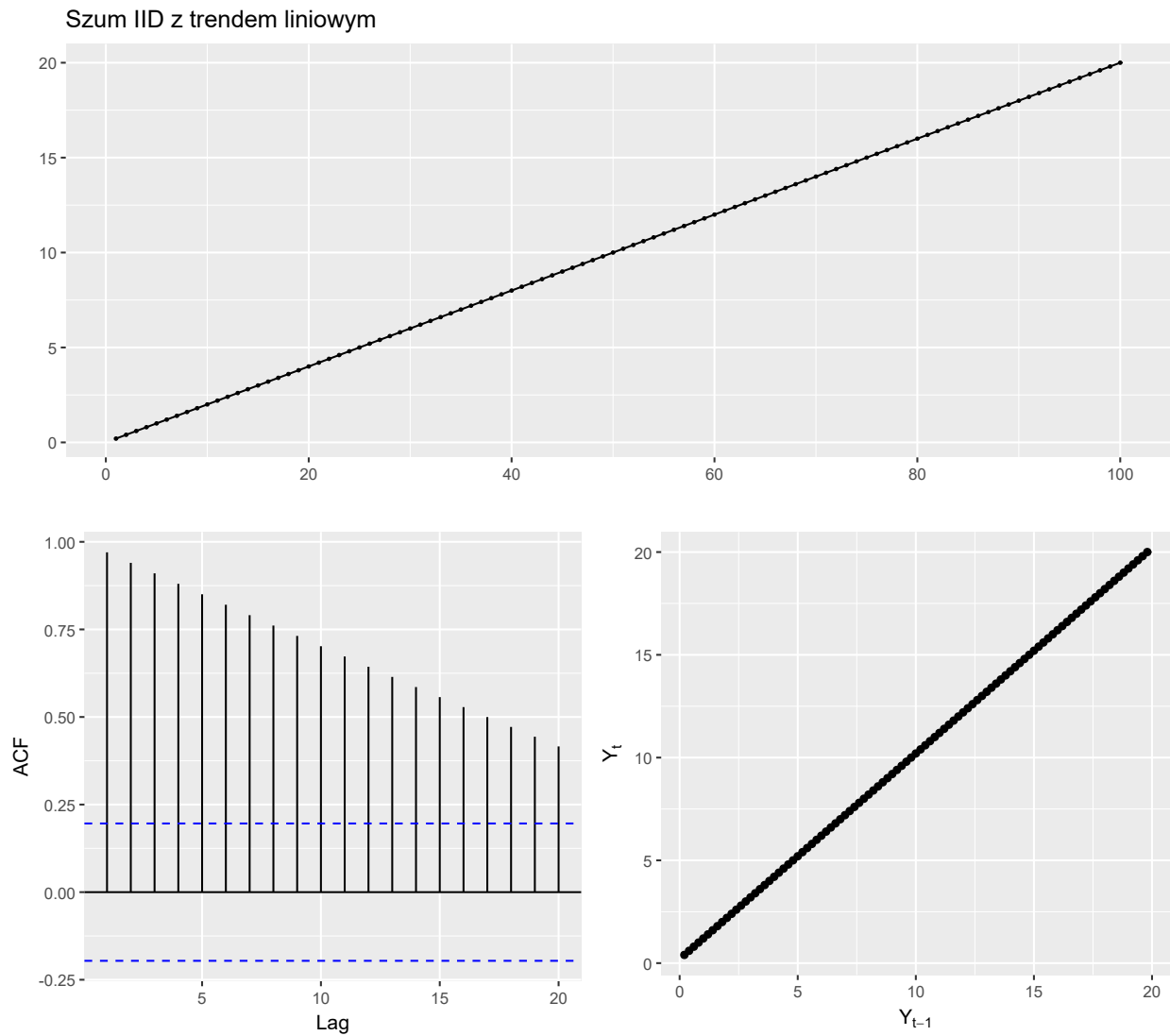
Model ruchomej średniej rzędu 1



Na podstawie wykresu widać, że model ruchomej średniej nie jest białym szumem, co prawda warunek dotyczący częstości wpadania autokorelacji można uznać za spełniony, jednak autokorelacja dla opóźnienia czasowego równego 1 znacznie przekracza pasmo ufności, wskazując na zależności czasowe.

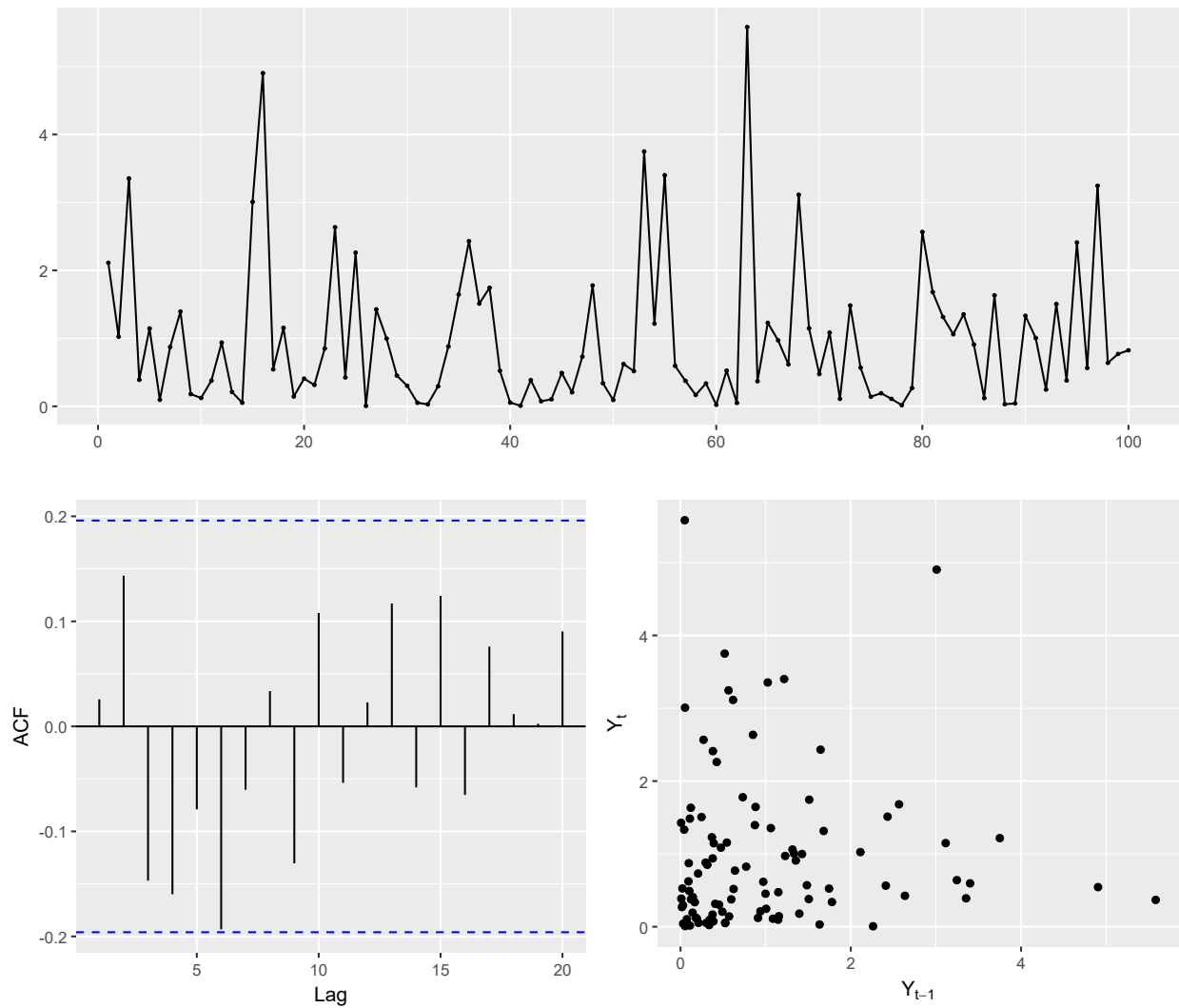


Na podstawie wykresu widać, że błądzenie losowe nie jest białym szumem, większość auto-korelacji przekracza pasmo ufności i jest malejąca, co świadczy o silnej zależności czasowej.



Na podstawie wykresu widać, że Szum IID z trendem liniowym wyraźnie pokazuje nielosową strukturę a wszystkie widoczne autokorelacje przekraczają pasmo ufności. Zatem można wywnioskować, że nie jest on białym szumem.

Biały szum o rozkładzie wykładniczym



Na wykresie nie widać żadnej regularnej struktury, a wszystkie autokorelacje mieszczą się w pasmach ufności. Z tego łatwo można wywnioskować, że analizowany szereg to realizacja białego szumu.

Formalne testy statystyczne

Podstawowymi testami opartymi na funkcji autokorelacji są:

- Test Boxa-Pierce’a,
- Test Ljungiego-Boxa.

Idea tych testów jest taka, że zamiast sprawdzać dla każdego h czy autokorelacja próbkowa znajduje się między przedziałami $\pm \frac{1.96}{\sqrt{n}}$, analizujemy pojedynczą wartość, tzn. statystykę testową opartą na ACF dla kilku/kilkudziesięciu początkowych opóźnień.

W przypadku testu Boxa-Pierce’a postać statystyki testowej wygląda następująco: $Q_{BP} = n \sum_{j=1}^h \hat{\rho}^2(j)$, gdzie h oznacza pewne maksymalne opóźnienie, a $\hat{\rho}(j)$ to próbkowa autokorelacja dla opóźnienia j .

W przypadku testu Ljungiego-Boxa postać statystyki testowej wygląda następująco: $Q_{LB} = n(n+2) \sum_{j=1}^h \frac{\hat{\rho}^2(j)}{n-j}$.

Duże wartości Q_{BP} i Q_{LB} będą oznaczały, że ACF są zbyt duże, aby uznać dane za realizację białego szumu. Aby stwierdzić czy te wartości są zbyt duże, wykorzystujemy wskaźnik p -value dla ustalonego poziomu istotności $\alpha = 0.05$. Do obliczenia tych statystyk zostanie wykorzystana funkcja `Box.test` z pakietu `stats`.

Testy Boxa-Pierce’a i Ljungiego-Boxa dla szumu IID

```
h.max <- floor(n/4)
BP_szum_iid <- Box.test(szum_iid, type="Box-Pierce", lag=h.max)
LB_szum_iid <- Box.test(szum_iid, type="Ljung-Box", lag=h.max)
```

Dla testu Boxa-Pierce’a wartość p -value jest równa 0.6327. Dla testu Ljungiego-Boxa wartość p -value jest równa 0.406, zatem w obydwu przypadkach wartość p -value znacznie przekracza poziom istotności $\alpha = 0.05$, czyli mamy do czynienia z szeregiem typu biały szum.

Testy Boxa-Pierce’a i Ljungiego-Boxa dla modelu ruchomej średniej

```
BP_ma1 <- Box.test(MA1, type="Box-Pierce", lag=h.max)
LB_ma1 <- Box.test(MA1, type="Ljung-Box", lag=h.max)
```

Dla testu Boxa-Pierce’a wartość p -value jest równa 0.3884. Dla testu Ljungiego-Boxa wartość p -value jest równa 0.1998, zatem w obu przypadkach wartość p -value znacznie przekracza poziom istotności $\alpha = 0.05$, czyli według testów mamy do czynienia z szeregiem typu biały szum, co nie jest prawdą.

Testy Boxa-Pierce'a i Ljung-Boxa dla błędzenia losowego

```
BP_rw <- Box.test(bladzenie_losowe, type="Box-Pierce", lag=h.max)
LB_rw <- Box.test(bladzenie_losowe, type="Ljung-Box", lag=h.max)
```

Dla testu Boxa-Pierce'a wartość *p-value* jest bliska 0. Dla testu Ljung-Boxa wartość *p-value* również jest bliska 0, zatem w obydwu przypadkach wartość *p-value* jest dużo mniejsza od wartości poziomu istotności $\alpha = 0.05$, czyli według testów ten szereg nie jest białym szumem.

Testy Boxa-Pierce'a i Ljung-Boxa dla szumu IID z trendem liniowym

```
BP_tl <- Box.test(trend.liniowy, type="Box-Pierce", lag=h.max)
LB_tl <- Box.test(trend.liniowy, type="Ljung-Box", lag=h.max)
```

Dla testu Boxa-Pierce'a wartość *p-value* jest bliska 0. Dla testu Ljung-Boxa wartość *p-value* również jest bliska 0, zatem w obydwu przypadkach wartość *p-value* jest dużo mniejsza od wartości poziomu istotności $\alpha = 0.05$, czyli według testów ten szereg nie jest białym szumem.

Testy Boxa-Pierce'a i Ljung-Boxa dla białego szumu o rozkładzie wykładniczym

```
BP_exp <- Box.test(szum_exp, type="Box-Pierce", lag=h.max)
LB_exp <- Box.test(szum_exp, type="Ljung-Box", lag=h.max)
```

Dla testu Boxa-Pierce'a wartość *p-value* jest równa 0.3829. Dla testu Ljung-Boxa wartość *p-value* jest równa 0.1944, zatem w obydwu przypadkach wartość *p-value* znacznie przekracza poziom istotności $\alpha = 0.05$, czyli mamy do czynienia z szeregiem typu biały szum.

1.1.4 Wnioski

- Dla powyższych szeregów testy graficzne w każdej sytuacji poprawnie weryfikowały hipotezę czy analizowany szereg jest szeregiem typu biały szum, natomiast formalne testy statystyczne mylnie zakwalifikowały model o ruchomej średniej do szeregów typu biały szum.
- Błąd w testach statystycznych jest efektem niskiej mocy testów przy dużej liczbie powtórzeń, w procesie MA(1) tylko pierwsza autokorelacja jest niezerowa, przez co jej wpływ zanika w sumie 25 opóźnień czasowych.

1.2 Symulacyjne porównanie białoszumowości

1.2.1 Krótki opis zagadnienia:

Celem tej części jest symulacyjne porównanie dwóch podejść do testowania białoszumowości: testu graficznego opartego na funkcji autokorelacji oraz formalnych testów Boxa-Pierce’a i Ljungiego-Boxa. Chcemy zweryfikować, jak często poszczególne testy poprawnie weryfikują hipotezę, że obserwowany szereg czasowy jest procesem białego szumu w różnych sytuacjach:

- dla różnych rozkładów,
- różnej długości szeregów,
- różnego wyboru maksymalnego opóźnienia,
- dla procesów, które nie są białym szumem.

1.2.2 Opis eksperymentów/analiz

W eksperymentach symulacyjnych wygenerujemy wielokrotne realizacje o liczbie N powtórzeń, gdzie $N \in \{100, 500, 1000\}$ szeregów czasowych długości $n \in \{25, 50, 100\}$. Rozważymy trzy różne rozkłady dla szumu IID:

- rozkład normalny $\mathcal{N}(0, 1)$,
- rozkład wykładniczy $\mathcal{E}(1)$,
- rozkład Bernoulliego $\mathcal{B}(0, 1)$.

Oprócz tego przeanalizujemy szeregi, które nie są białym szumem:

- Model ruchomej średniej rzędu 1,
- Błądzenie losowe,
- Szum IID z trendem liniowym.

Dla każdej kombinacji modelu, długości szeregu, liczby powtórzeń oraz maksymalnego opóźnienia obliczymy częstość odrzucenia hipotezy o białoszumowości przez test graficzny oraz testy Boxa-Pierce’a i Ljungiego-Boxa.

1.2.3 Wyniki

Dla każdego scenariusza wygenerujemy wiele realizacji danych, a następnie dla każdej realizacji ocenimy, czy dany test odrzuca hipotezę zerową o białoszumowości. Do weryfikacji hipotezy o białoszumowości zaimplementujemy funkcje:

- `test_graficzny` zwracającą wartość 0 lub 1 w zależności, czy test przyjął hipotezę o białoszumowości (0 - nie przyjął, 1 - przyjął)
- `testy_statystyczne` zwracającą wartość 0 lub 1 w zależności, czy testy Boxa-Pierce’a i Ljungiego-Boxa przyjęły hipotezę o białoszumowości (0 - nie przyjęły, 1 - przyjęły)

Dla każdej konfiguracji obliczymy częstość odrzucenia, które pozwolą nam wywnioskować, które parametry w którym modelu najbardziej wpływają na skuteczność poszczególnych testów.

Wpływ różnych rozkładów na częstość odrzucenia hipotezy o białoszumowości

Tabela 1: Wpływ różnych rozkładów na częstość odrzucenia hipotezy o białoszumowości ($N = 1000$, $n = 50$, $h.\max = 12$).

model	n	liczba_powtorzen	h.max	odrz_T_Graf	odrz_T_BP	odrz_T_LB
BS_ber	50	1000	12	0.308	0.035	0.070
BS_exp	50	1000	12	0.267	0.020	0.050
BS_norm	50	1000	12	0.298	0.018	0.064

Analiza wykazała niewielki wpływ rodzaju rozkładu na działanie testów. Test graficzny konsekwentnie wykazywał wyższy, lecz stabilny poziom odrzuceń w porównaniu do testów formalnych. Testy Boxa-Pierce'a i Ljungiego-Boxa utrzymywały niską częstość odrzuceń niezależnie od rozkładu, potwierdzając swoją odporność na jego kształt przy skończonej wariancji.

Wpływ długości szeregu n na częstość odrzucenia hipotezy o białoszumowości

Tabela 2: Wpływ długości szeregu n na częstość odrzucenia hipotezy o białoszumowości dla modelu BS_norm ($N = 1000$, $h.\max = h_{\text{graniczne}}$).

model	n	liczba_powtorzen	h.max	odrz_T_Graf	odrz_T_BP	odrz_T_LB
BS_norm	25	1000	12	0.193	0.017	0.094
BS_norm	50	1000	25	0.109	0.010	0.093
BS_norm	100	1000	50	0.143	0.012	0.111

Wzrost długości szeregu nie przełożył się wprost na wzrost mocy testu graficznego. Test Boxa-Pierce'a zanotował bardzo niską czułość wobec wszystkich analizowanych długości. Test Ljungiego-Boxa wykazał względnie stałą i nieco wyższą moc, sugerując mniejszą wrażliwość na krótkie szeregi.

Wpływ maksymalnego opóźnienia h_{max} na częstość odrzucenia hipotezy o białoszumowości

Tabela 3: Wpływ maksymalnego opóźnienia na częstość odrzucenia hipotezy o białoszumowości dla modelu BS_exp ($N = 1000$, $n = 100$).

model	n	liczba_powtorzen	h.max	odrz_T_Graf	odrz_T_BP	odrz_T_LB
BS_exp	100	1000	4	0.170	0.044	0.052
BS_exp	100	1000	25	0.187	0.020	0.056
BS_exp	100	1000	50	0.116	0.003	0.077

Wyniki demonstrują kluczowy wpływ parametru h_{max} na moc wszystkich badanych testów. W miarę jego zwiększania, czułość testów (poza testem Ljung-Boxa) spada, przy czym najbardziej radykalny spadek obserwuje się dla testu Boxa-Pierce’a. Co istotne, nawet test graficzny, uważany za odporniejszy, traci moc przy zbyt dużych opóźnieniach.

Wpływ różnych innych procesów nie będących białym szumem na częstość odrzucenia hipotezy o białoszumowości

Tabela 4: Wpływ różnych innych procesów nie będących białym szumem na częstość odrzucenia hipotezy o białoszumowości ($N = 1000$, $n = 100$, $h_{max} = 25$).

model	n	liczba_powtorzen	h.max	odrz_T_Graf	odrz_T_BP	odrz_T_LB
MA1	100	1000	25	0.943	0.546	0.684
RW	100	1000	25	1.000	1.000	1.000
SZ_trend_lin	100	1000	25	1.000	1.000	1.000

Jak można zauważyć wszystkie rozważane procesy, które nie są białym szumem, są prawidłowo rozpoznawane przez testy, częstości odrzucenia hipotezy o białoszumowości są wysokie. Jedynym wyjątkiem jest proces MA(1), dla którego testy formalne rzadziej odrzucają hipotezę o białoszumowości, co jest spowodowane tym, że MA(1) ma tylko jedną istotną autokorelację, a pozostałe są zerowe.

Wpływ ilości powtórzeń na częstość odrzucenia hipotezy o białoszumowości

Tabela 5: Wpływ ilości powtórzeń na częstość odrzucenia hipotezy o białoszumowości dla modelu BS_ber ($n = 50$, $h_{\max} = 25$).

model	n	liczba_powtorzen	h.max	odrz_T_Graf	odrz_T_BP	odrz_T_LB
BS_ber	50	100	25	0.150	0.010	0.100
BS_ber	50	500	25	0.122	0.024	0.120
BS_ber	50	1000	25	0.155	0.023	0.114

Zmiana liczby powtórzeń N wpływa jedynie na niewielkie wahania estymowanych częstości hipotezy o białoszumowości. Zarówno test graficzny, jak i formalne, osiągnęły stateczny poziom mocy już dla $N = 500$. Potwierdza to rzetelność procedury symulacyjnej i wiarygodność uzyskanych estymatorów mocy testów.

1.2.4 Podsumowanie:

Test graficzny okazał się najbardziej wrażliwy na sam fakt występowania jakiegokolwiek autokorelacji, wykrywając ją tam, gdzie testy formalne zawodziły. Z kolei testy formalne były wrażliwe na dobór parametru maksymalnego opóźnienia h_{\max} . Testy Boxa-Pierce’a i Ljunga-Boxa charakteryzowały się znacznie większą stabilnością, utrzymywały poziom odrzucenia bliski poziomowi istotności, niezależnie od parametrów symulacji. Wyniki te wskazują, że test graficzny jest użytecznym narzędziem wstępnej diagnostyki, jednak przy analizie danych o małej długości lub przy dużej liczbie opóźnień może być zawodny. Testy formalne są bardziej wiarygodne i powinny stanowić podstawową metodę weryfikacji białoszumowości.

2 Dekompozycja szeregów czasowych - eliminacja trendu i sezonowości

2.1 Krótki opis zagadnienia:

Analizować będziemy szereg czasowy `austourists` z pakietu `fpp2`. Są to dane pokazujące liczbę noclegów (w milionach) międzynarodowych turystów w Australii w latach 1999 – 2015. Głównym celem jest dekompozycja tych danych oraz eliminacja trendu i sezonowości. Przeanalizujemy także podstawowe własności szeregu oraz sprawdzimy w jakim stopniu wybór parametrów danej metody dekompozycji wpływa na otrzymane wyniki.

W celu wczytania danych, najpierw załadujemy pakiet `fpp2`. Udostępnia on szereg `austourists`, który będziemy analizować.

```
library(fpp2)
```

2.2 Opis eksperymentów/analiz

Przebieg analizy można podzielić na cztery główne etapy:

- Wstępna analiza graficzna - na początku konieczna jest ocena i analiza własności danych, w tym celu zastosujemy wykresy zwykłe i sezonowe.
- Zastosowanie i porównanie metod dekompozycji - następnie przetestujemy i porównamy trzy podstawowe metody dekompozycji: dekompozycje na podstawie ruchomej średniej, na podstawie modelu regresji oraz dekompozycje STL. W ramach tego etapu zbadamy również, jak zmiana kluczowych parametrów wpływa na otrzymane wyniki.
- Testowanie transformacji Boxa-Coxa - sprawdzimy, czy transformacja prowadzi do stabilizacji wariancji, a tym samym do poprawy jakości dopasowania modeli dekompozycji.
- Porównanie reszt i analiza stacjonarności - to kluczowy etap analizy. Porównamy szeregi reszt uzyskane ze wszystkich powyższych metod. Celem jest sprawdzenie, która technika najskuteczniej wyczyści dane z trendu i sezonowości.

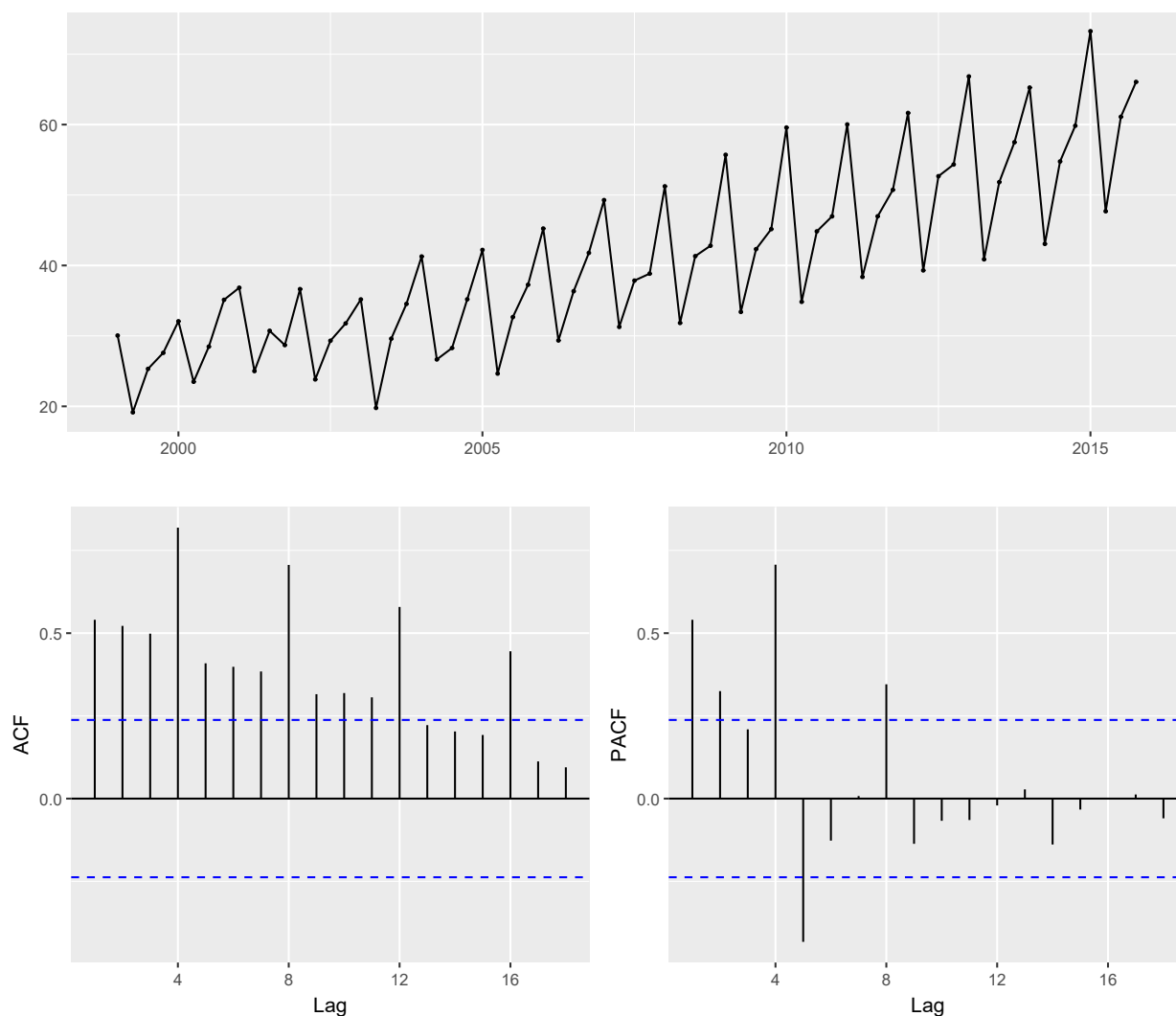
2.3 Wyniki

2.3.1 Analiza graficzna podstawowych własności szeregu

Zbiórcza diagnostyka szeregu

Analizę rozpoczynamy od wizualizacji szeregu w czasie. Pozwala to na wstępną identyfikację jego cech: obecność trendu, sezonowości oraz ewentualnych obserwacji odstających. Wykorzystamy do tego funkcję `ggtsdisplay`, która dodatkowo połączy zwykły wykres szeregu z wykresami funkcji autokorelacji (ACF) i częściowej autokorelacji (PACF).

```
ggtsdisplay(austourists)
```

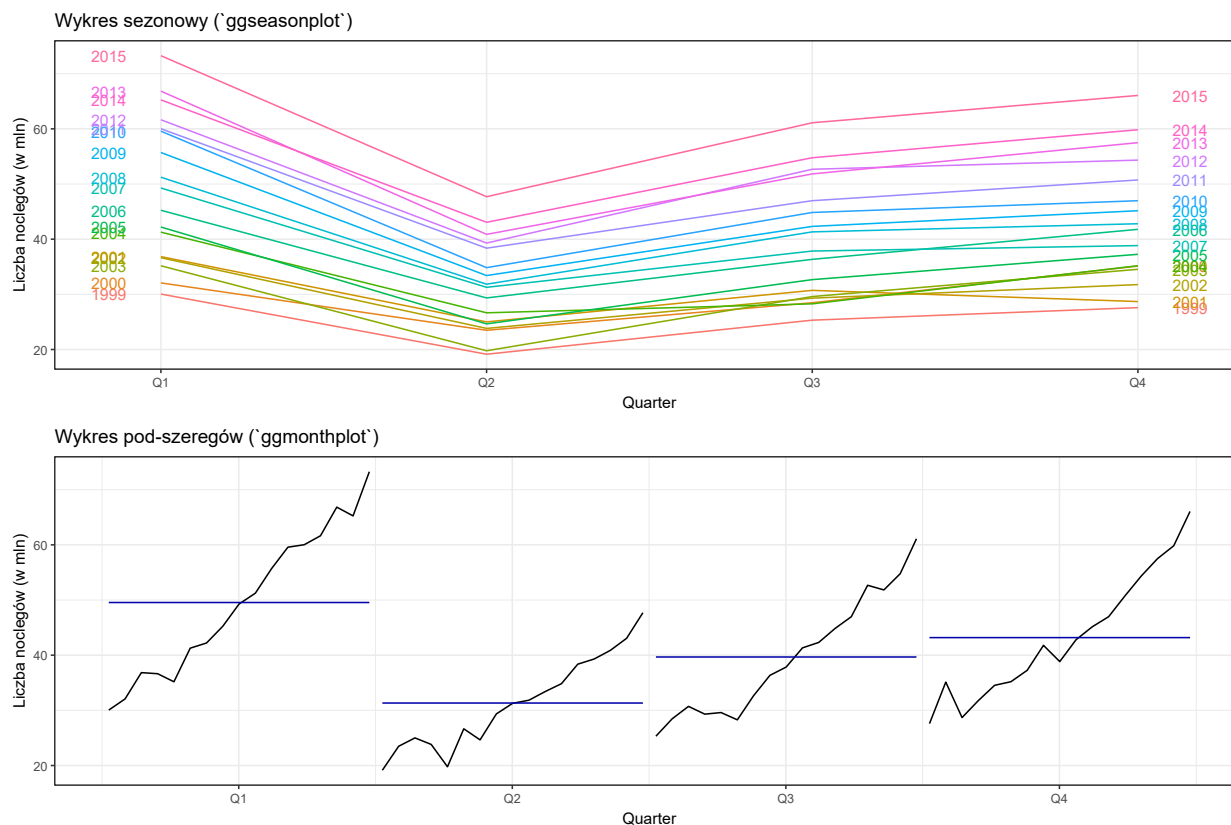


Na podstawie powyższych wykresów, możemy zaobserwować następujące cechy:

- Trend wzrostowy: Ogólna liczba noclegów turystycznych systematycznie rośnie w analizowanym okresie. Potwierdza to wykres **ACF**, na którym autokorelacje opadają bardzo powoli, jest to objaw niestacjonarności.
- Silna sezonowość: Obserwujemy regularne, powtarzające się co roku wahania. Ponadto wysokość wahań sezonowych rośnie wraz z poziomem trendu, więc mamy do czynienia z wariancją niejednorodną. Na wykresie **ACF** objawia się to wystającymi słupkami w punktach 4, 8 i 12. Wykres **PACF** również pokazuje mocną zależność w punkcie 4.
- Brak wartości odstających: Szereg jest bardzo regularny, zdarzają się lata z niższymi wartościami, ale żadna z obserwacji nie odbiega drastycznie od ogólnego wzorca.

Analiza sezonowości

Aby dokładniej zbadać charakter sezonowości, tworzymy dwa uzupełniające się wykresy za pomocą funkcji `ggseasonplot` oraz `ggmonthplot`.



- Wykres wygenerowany za pomocą funkcji `ggseasonplot`, potwierdza regularny wzorec sezonowy. Szczyt przyjazdów przypada na pierwszy kwartał, a największy spadek na kwartał drugi. Każda kolejna linia, symbolizująca następne lata jest położona wyżej od poprzedniej, co potwierdza istnienie trendu.
- Wykres, do generacji którego użyliśmy funkcji `ggmonthplot`, pokazuje to zjawisko jeszcze wyraźniej. Analizuje on każdy kwartał z osobna. Widać, że we wszystkich czterech kwartałach obserwacje rosną z roku na rok.

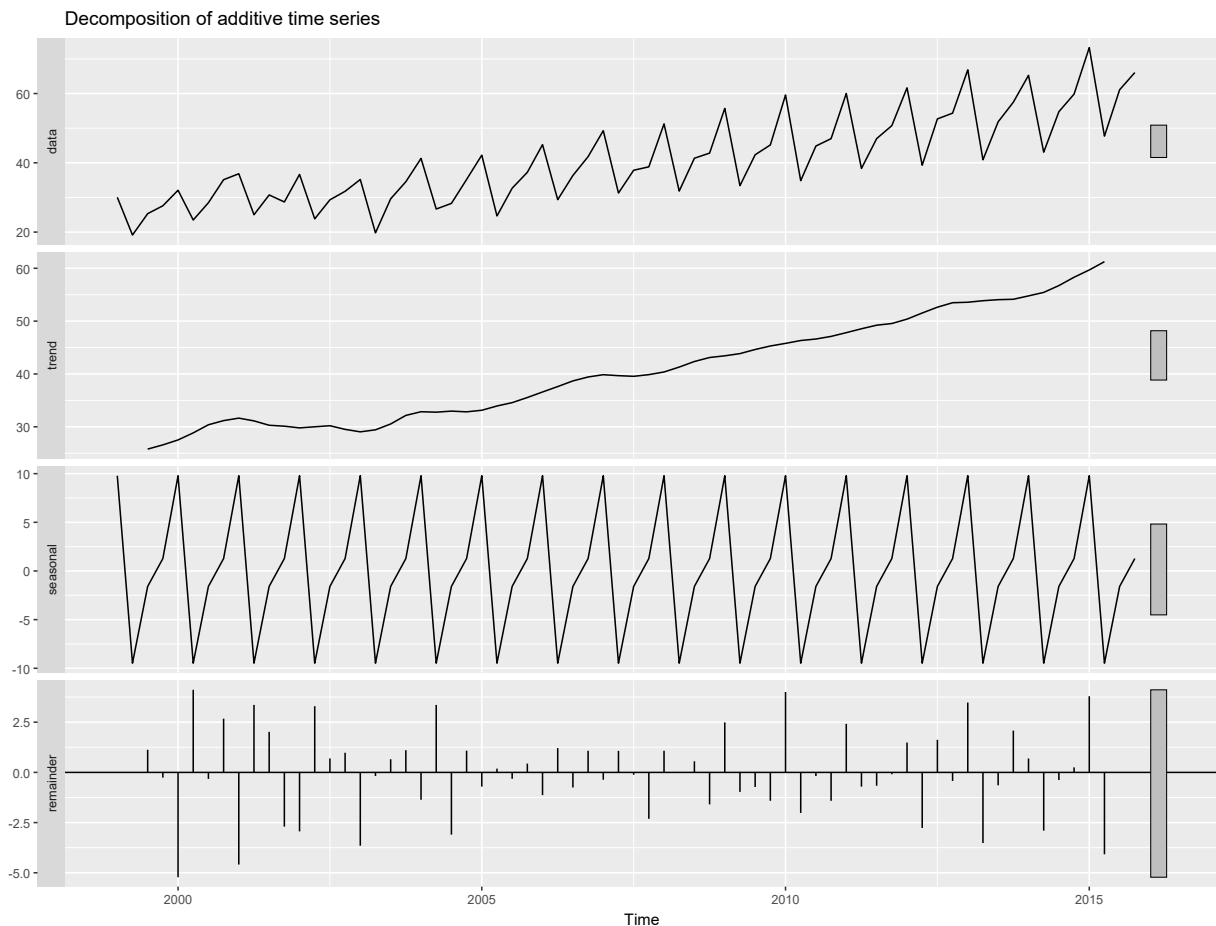
2.3.2 Dekompozycja szeregu czasowego z zastosowaniem różnych metod

Dekompozycja na podstawie ruchomej średniej

Rozpoczynamy od klasycznej dekompozycji opartej na średnich ruchomych (funkcja `decompose`). Aby sprawdzić, który model lepiej opisuje dane, przetestujemy oba warianty - addytywny i multiplikatywny.

```
# Dekompozycja addytywna
```

```
dekomp.add <- decompose(austourists, type = "additive")  
autoplot(dekomp.add)
```

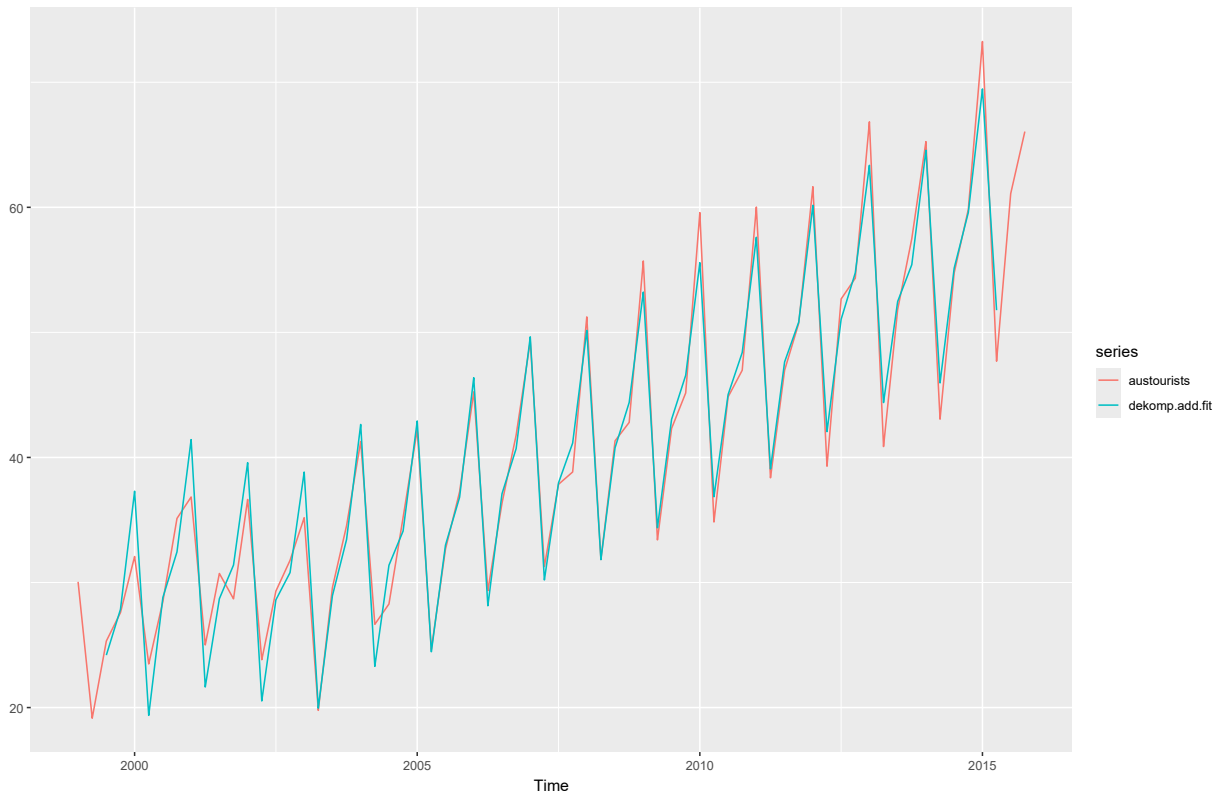


Dwa górne wykresy pokazują oryginalne dane oraz oszacowany, wygładzony trend. Trzeci wykres pokazuje, że wahania sezonowe mają stałą siłę przez cały czas. Kluczowy jest jednak dolny wykres, czyli wykres reszt. Nie są one losowe i nie przypominają białego szumu. Ich wariancja jest duża na początku szeregu (lata 1999–2002), następnie maleje w środkowym okresie (ok. 2003–2010), po czym ponownie rośnie pod koniec analizy. Taki nielosowy wzór pokazuje, że model addytywny jest źle dopasowany do tego szeregu.

Aby lepiej zwizualizować dopasowanie modelu addytywnego, sumujemy wartości oszacowania dla trendu i sezonowości, i porównamy je z oryginalnym szeregiem

```
dekomp.add.trend      <- dekomp.add$trend
dekomp.add.sezonowosc <- dekomp.add$seasonal
dekomp.add.ind.sezon  <- dekomp.add$figure
dekomp.add.reszty     <- dekomp.add$random

dekomp.add.fit <- dekomp.add.trend + dekomp.add.sezonowosc
autoplot(cbind(austourists, dekomp.add.fit), ylab="")
```



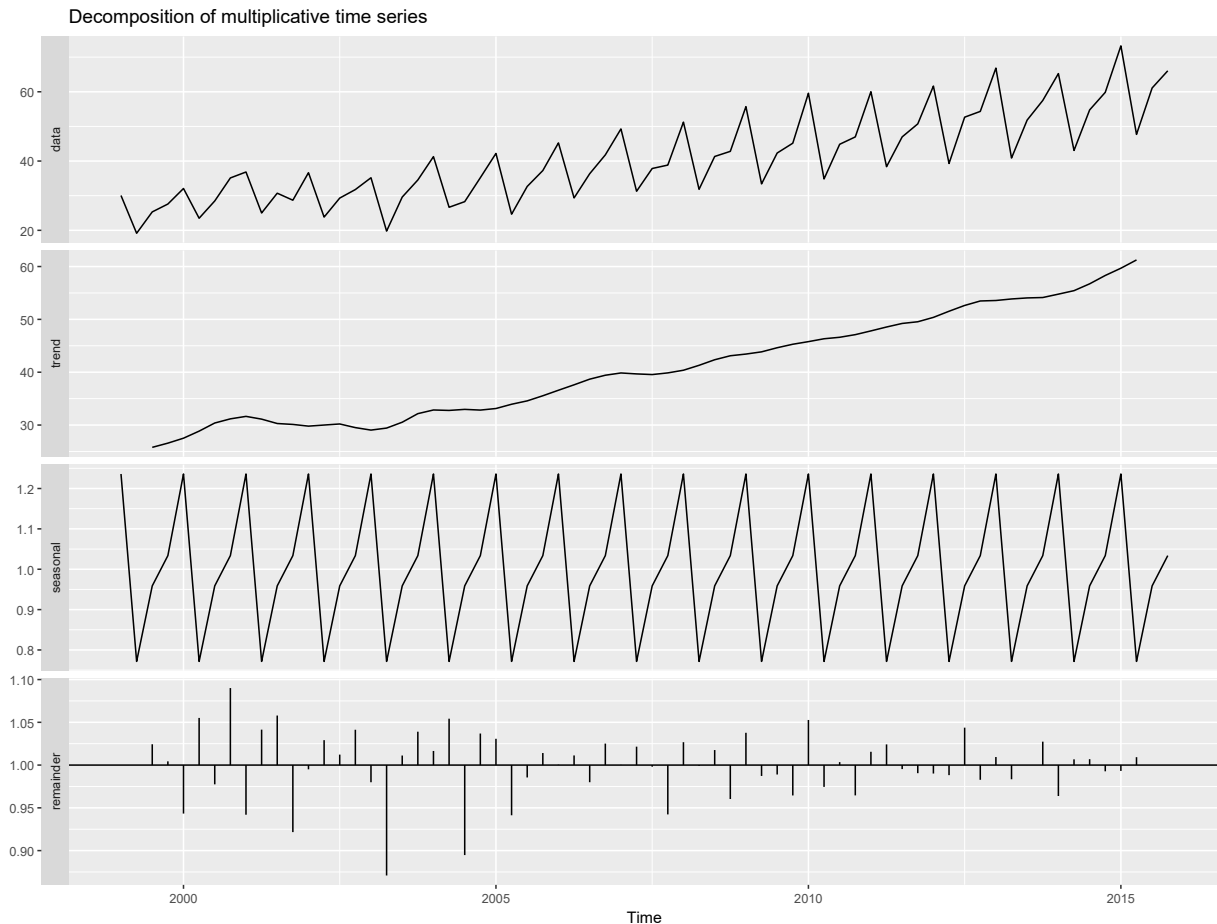
Na wykresie widać wyraźnie, że model addytywny nie pasuje dobrze do danych. Jego stała amplituda sezonowa (niebieska linia) nie odzwierciedla rosnących wahań w rzeczywistych danych (czerwona linia). W efekcie model zawyża wartości na początku okresu, a pod koniec je zaniża.

W celu poprawy jakości dopasowania wykorzystamy teraz dekompozycje multiplikatywną.

```
# Dekompozycja multiplikatywna
```

```
dekomp.mult <- decompose(austourists, type = "multiplicative")
```

```
autoplot(dekomp.mult)
```



Zastosowanie dekompozycji multiplikatywnej przyniosło poprawę dopasowania. Widać to przede wszystkim na dolnym wykresie, gdzie reszty są teraz bardziej stabilne. Ich wariancja jest bardziej stała w całym okresie, co eliminuje wzorec widoczny w modelu addytywnym.

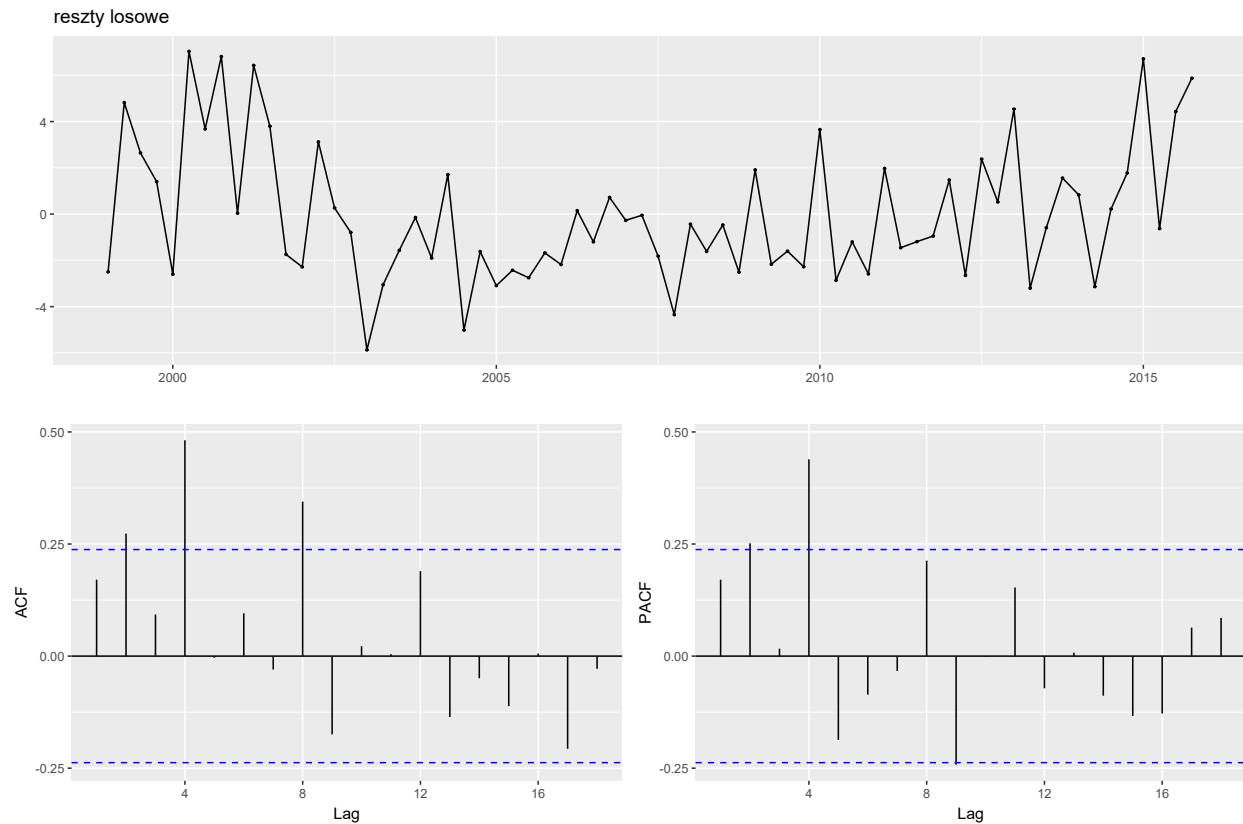
Wnioski:

Analiza za pomocą funkcji `decompose` pokazała, że szereg `austourists` ma naturę multiplikatywną. Model addytywny nie sprawdził się i nie był w stanie poprawnie opisać struktury tych danych. Dopiero zastosowanie dekompozycji multiplikatywnej dało znacznie lepsze wyniki.

Dekompozycja na podstawie modelu regresji

Zastosujemy teraz metodę dekompozycji na podstawie modelu regresji. Najpierw dopasujemy prosty model uwzględniający liniowy trend i stałą sezonowość.

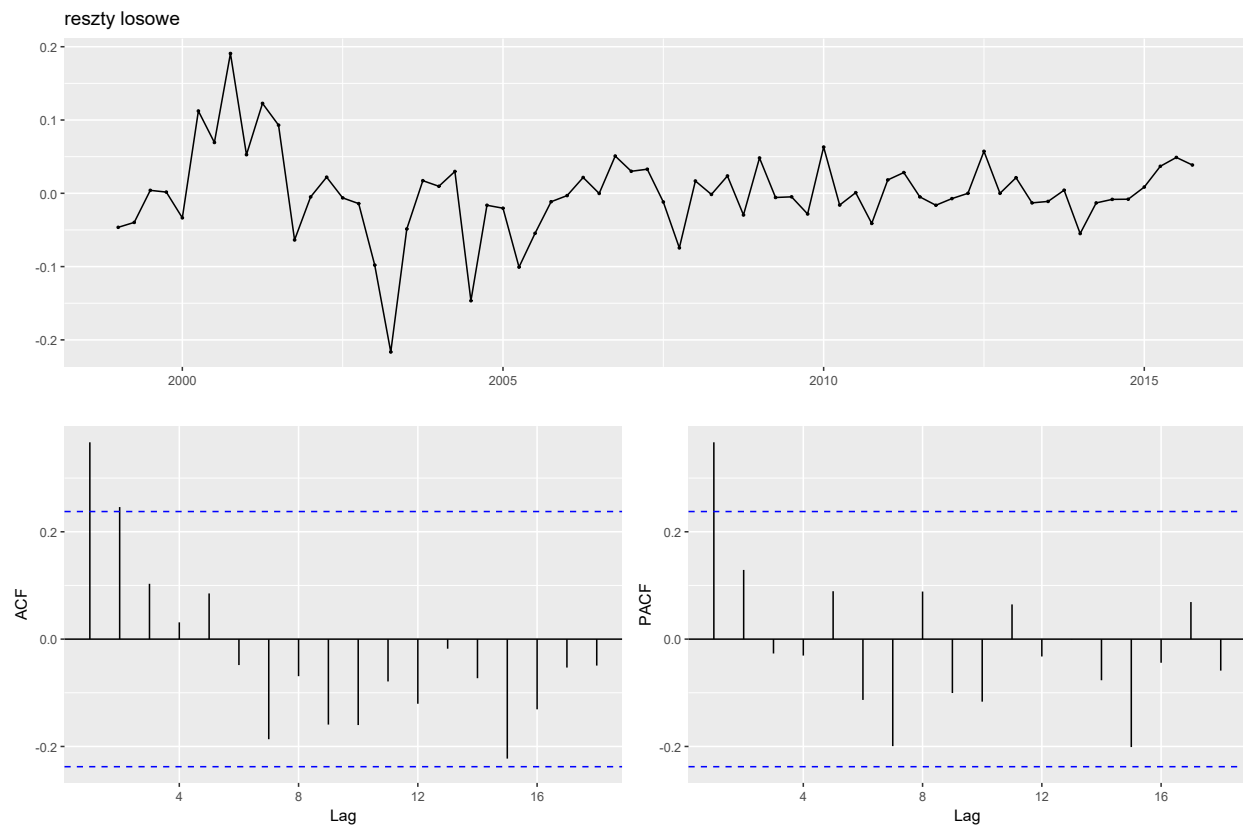
```
# Dekompozycja na podstawie modelu regresji: trend liniowy + sezonowość
tslm.1 <- tslm(austourists ~ trend + season)
summary(tslm.1)
ggtsdisplay(residuals(tslm.1), main = "reszty losowe")
```



Te wykresy pokazują, że model regresji źle dopasował się do danych. Górny wykres wskazuje na niestabilną wariancję, jest ona duża na początku i na końcu okresu, a wyraźnie mniejsza w środku. Ponadto, wykresy ACF i PACF pokazują zależności, szczególnie w punkcie 4, co dowodzi, że model nie wyeliminował w pełni wahań sezonowych.

Sprawdźmy, co stanie się gdy analizie poddamy dane zlogarytmowane.

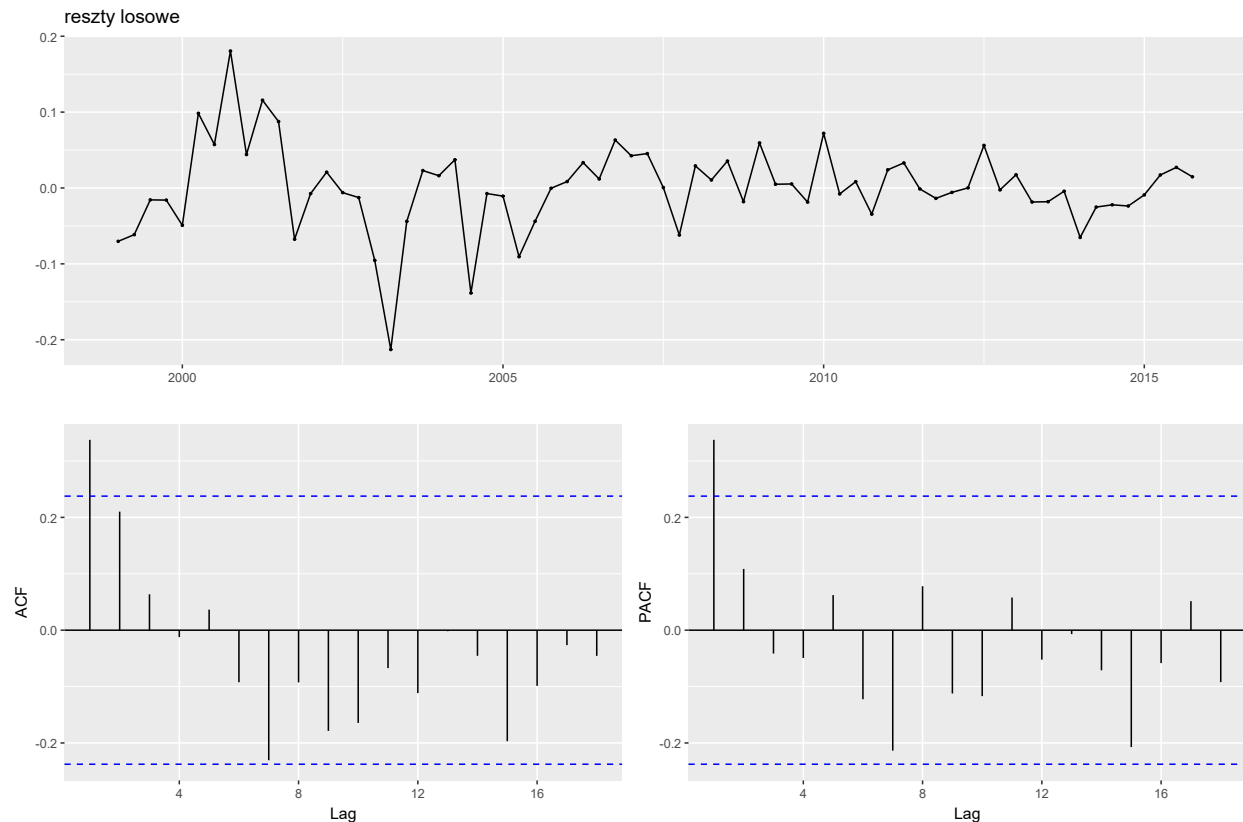
```
# Dekompozycja zlogarytmowanych danych na podstawie modelu regresji:  
# trend liniowy + sezonowość  
log.tslm.2 <- tslm(austourists ~ season + trend, lambda = 0)  
summary(log.tslm.2)  
ggtsdisplay(residuals(log.tslm.2), main = "reszty losowe")
```



Zastosowanie transformacji logarytmicznej nieco poprawiło model. Górny wykres pokazuje, że reszty mają teraz stabilniejszą wariancję. Dolne wykresy pokazują jednak, że reszty wciąż nie są losowe.

Do zlogarytmowanych danych zastosujemy teraz model opierający się na trendzie kwadratowym i sezonowości.

```
# Dekompozycja zlogarytmowanych danych na podstawie modelu regresji:  
# trend kwadratowy + sezonowość  
log.tslm.3 <- tslm(austourists ~ season + trend + I(trend ^ 2), lambda = 0)  
summary(log.tslm.3)  
ggtdisplay(residuals(log.tslm.3), main = "reszty losowe")
```



Dodanie trendu kwadratowego niewiele zmieniło w porównaniu do poprzedniego modelu liniowego, a reszty nadal nie są losowe.

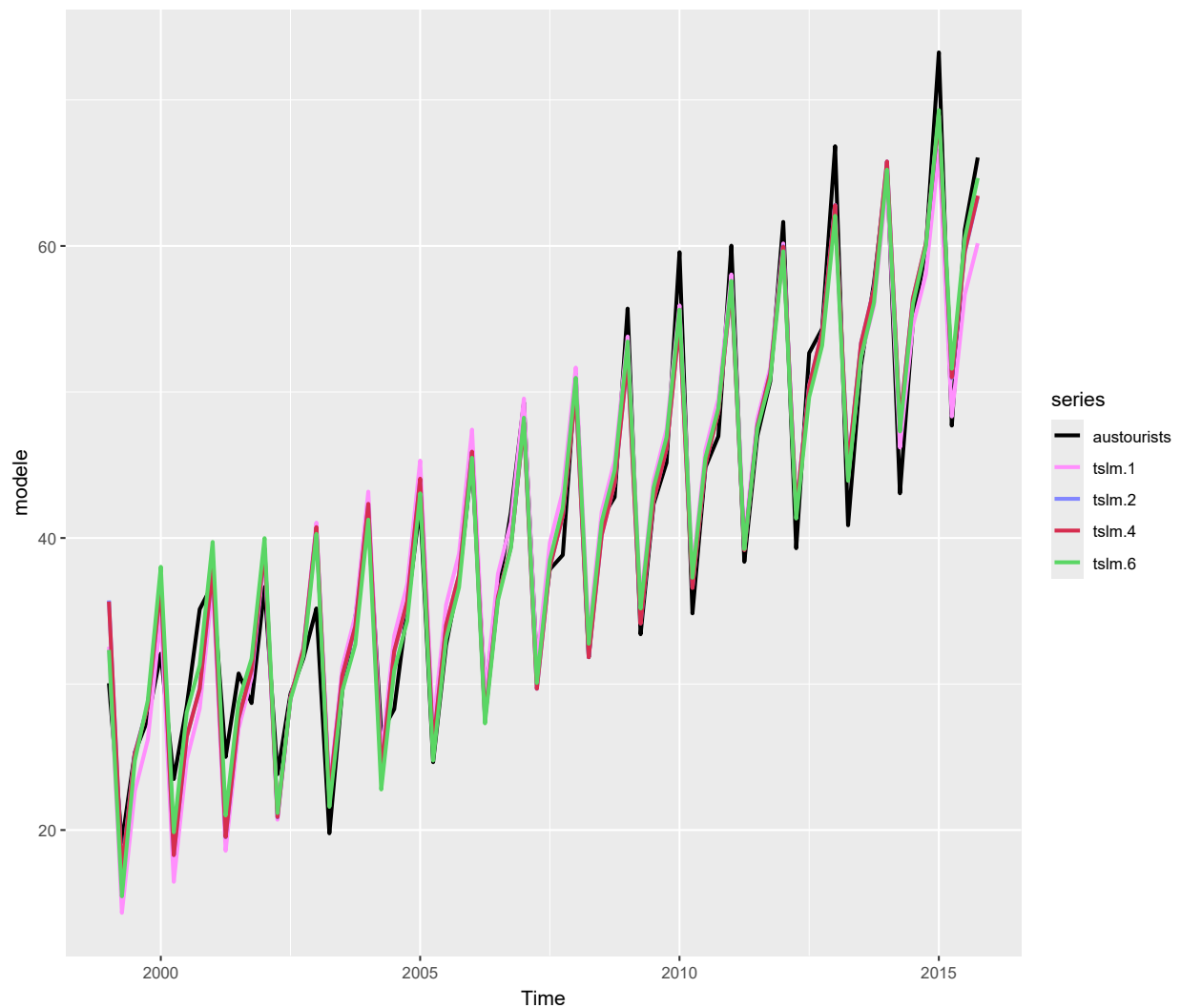
Dekompozycję funkcją `tslm()` przeprowadzimy teraz dla wielomianów różnych stopni, następnie porównamy otrzymane dopasowania.

```
pop.tslm1 <- tslm(austourists ~ trend + season)
pop.tslm2 <- tslm(austourists ~ poly(trend, 2, raw = TRUE) + season)
pop.tslm4 <- tslm(austourists ~ poly(trend, 4, raw = TRUE) + season)
pop.tslm6 <- tslm(austourists ~ poly(trend, 6, raw = TRUE) + season)
```

```
tslm.1 <- fitted(pop.tslm1)
tslm.2 <- fitted(pop.tslm2)
tslm.4 <- fitted(pop.tslm4)
tslm.6 <- fitted(pop.tslm6)
```

```
modele <- ts.union(austourists, tslm.1, tslm.2, tslm.4, tslm.6)
```

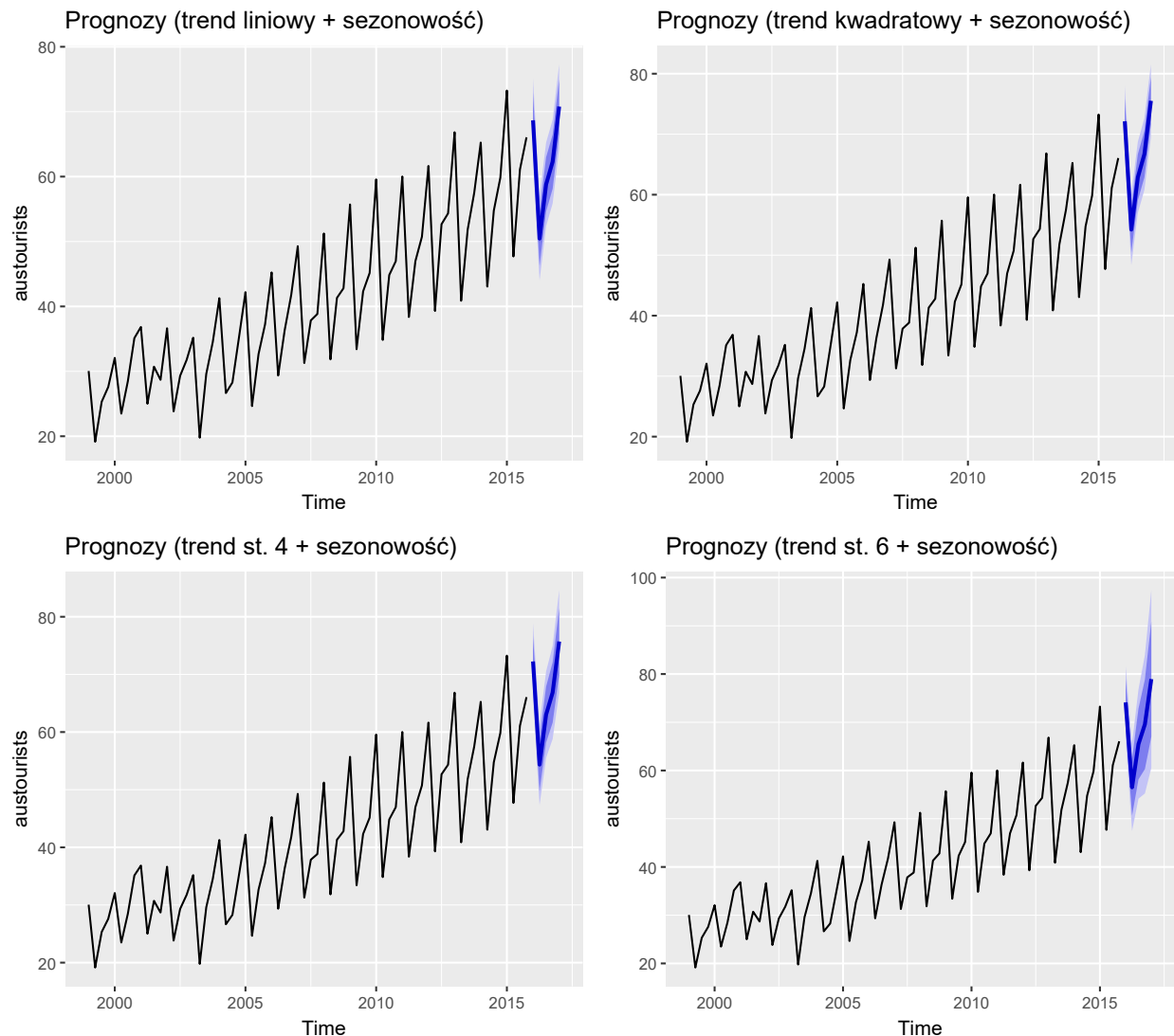
Dopasowanie modeli: trend wielomianowy + sezonowość



Modele o wyższym stopniu wielomianu, zwłaszcza **tslm.4** i **tslm.6**, stają się przeuczone. Zamiast wygładzać ogólny trend, zaczynają one dopasowywać się do wahań w danych. Widać to na początku i na końcu szeregu. Oznacza to, że te modele są niestabilne i nie nadają się do prognozowania. Prostsze modele (liniowy lub kwadratowy) są tu znacznie bezpieczniejsze.

Możemy teraz wyznaczyć prognozy dla poszczególnych parametrów.

```
prognozy.pop1 <- forecast(pop.tslm1, h=5)
prognozy.pop2 <- forecast(pop.tslm2, h=5)
prognozy.pop4 <- forecast(pop.tslm4, h=5)
prognozy.pop6 <- forecast(pop.tslm6, h=5)
```

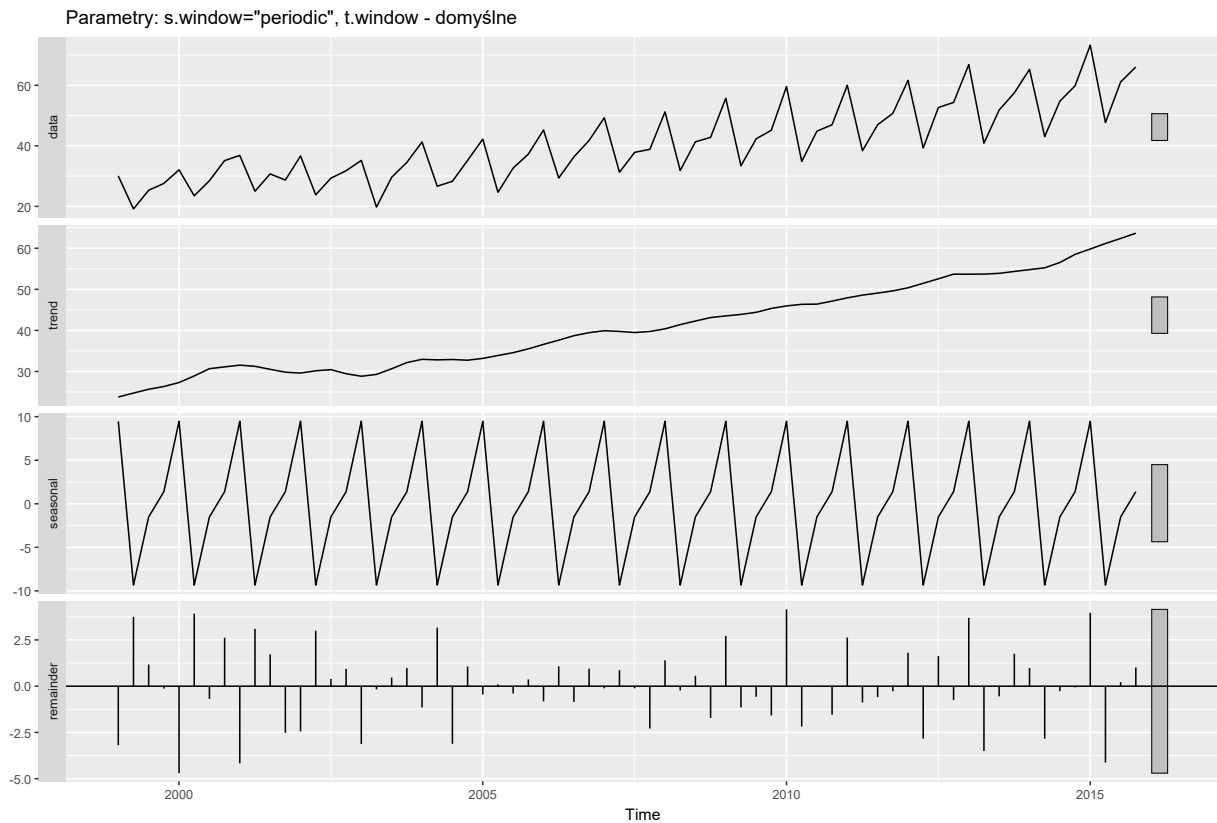


Im wyższy stopień wielomianu (szczególnie stopień 6), tym prognoza za bardzo wystrzeliwuje w górę i staje się nierealistyczna. Sugeruje to, że modele wyższych rzędów są przeuczone i nie nadają się do prognozowania.

Dekompozycja STL oparta na metodzie loess

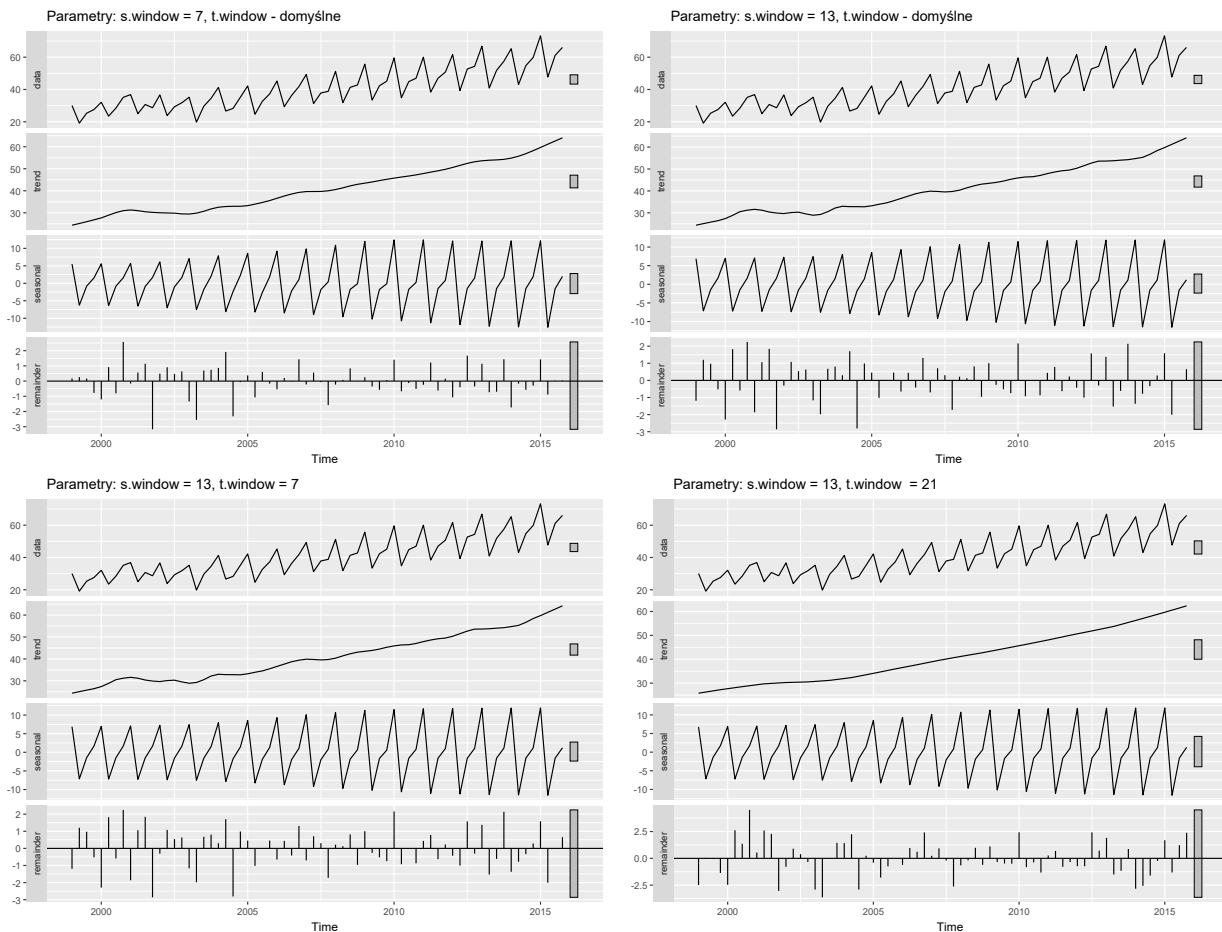
Teraz przetestujemy dekompozycję STL, która jest metodą opartą na wygładzaniu loess. Sprawdzimy, jak zmiana kluczowych parametrów wygładzających, czyli `s.window` dla sezonowości i `t.window` dla trendu, wpływa na kształt poszczególnych składowych.

```
# Dekompozycja STL: s.window = "periodic", t.window - domyślne'  
decomp.stl.1 <- stl(austourists, s.window="periodic")  
autoplot(decomp.stl.1) +  
  ggtitle('Parametry: s.window="periodic", t.window - domyślne')
```



W tym przypadku nasz model znalazł jeden, stały wzorec sezonowy, który powtarza się w każdym roku. Amplituda tych wahań jest niezmienna w czasie. Na wykresie reszt, możemy zauważyć, że nie są one losowe. Oznacza to, że ten konkretny model jest źle dopasowany do danych.

Sprawdźmy teraz, jaki wpływ na wyniki będą miały różne parametry wygładzające.



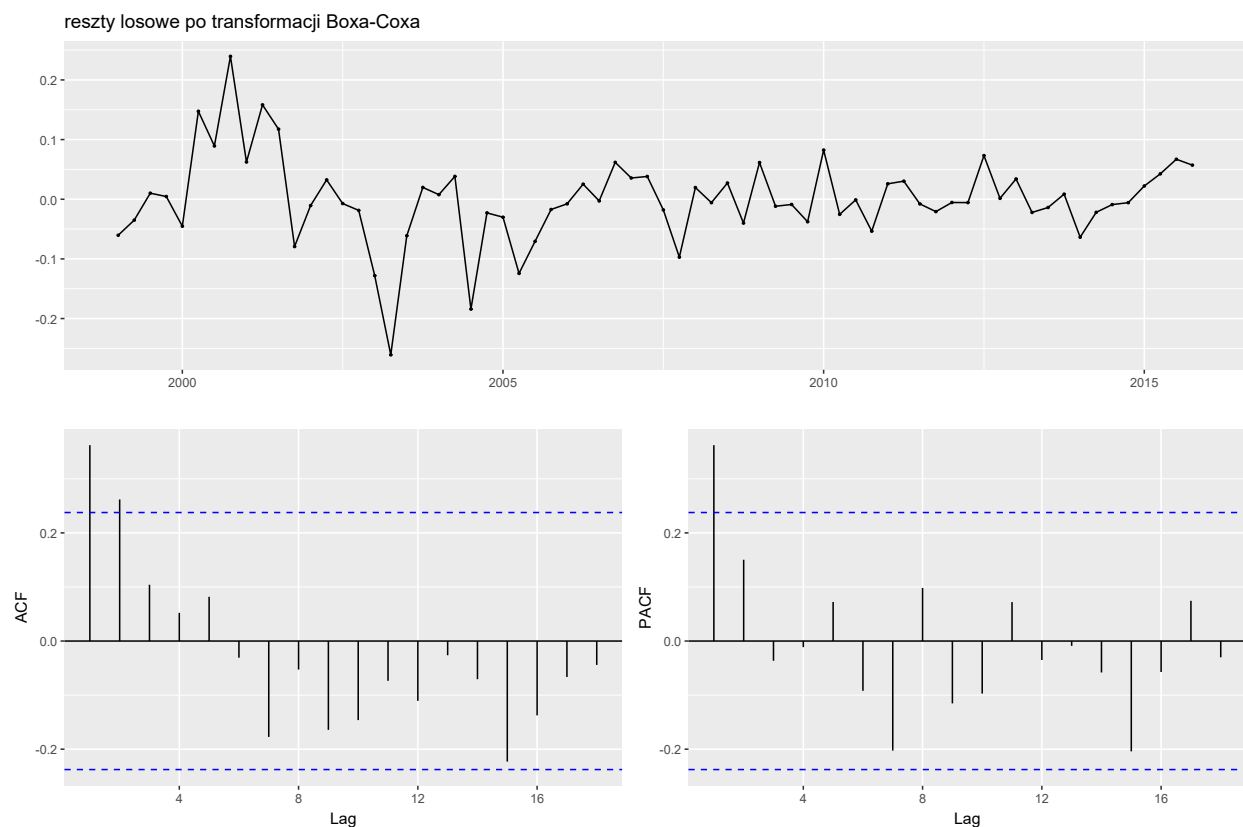
Zmiana parametrów `t.window` i `s.window` wpłynęła na wygląd składowych, ale nie rozwiązała głównego problemu. Zmiany były widoczne głównie w linii trendu: mniejsza wartość `t.window` (np. 7) dała trend bardziej pofalowany, podczas gdy większa (np. 21) mocno go wygładziła. We wszystkich czterech przypadkach panel reszt nie uległ poprawie. Wyniki te pokazują, że problemem nie są ustawienia wygładzania, ale niedopasowanie addytywnej metody STL do danych.

2.3.3 Transformacja Boxa-Coxa

Poprzednio testowane metody dekompozycji nie przyniosły oczekiwanych rezultatów. Sprawdźmy jednak, jak zachowują się dane, jeżeli najpierw zastosujemy transformację stabilizującą wariancję, a następnie wykorzystamy model dekompozycji addytywnej.

Zastosujemy transformację Boxa-Coxa do dekompozycji na podstawie modelu regresji.

```
tslm.1 <- tslm(austourists ~ trend + season, lambda = "auto")
summary(tslm.1)
ggtsdisplay(residuals(tslm.1),
            main = "reszty losowe po transformacji Boxa-Coxa")
```



Zmiana polegała na dodaniu argumentu `lambda = "auto"` do funkcji `tslm()`. To polecenie znalazło najlepszą wartość parametru λ transformacji Boxa-Coxa i dopasowało model regresji.

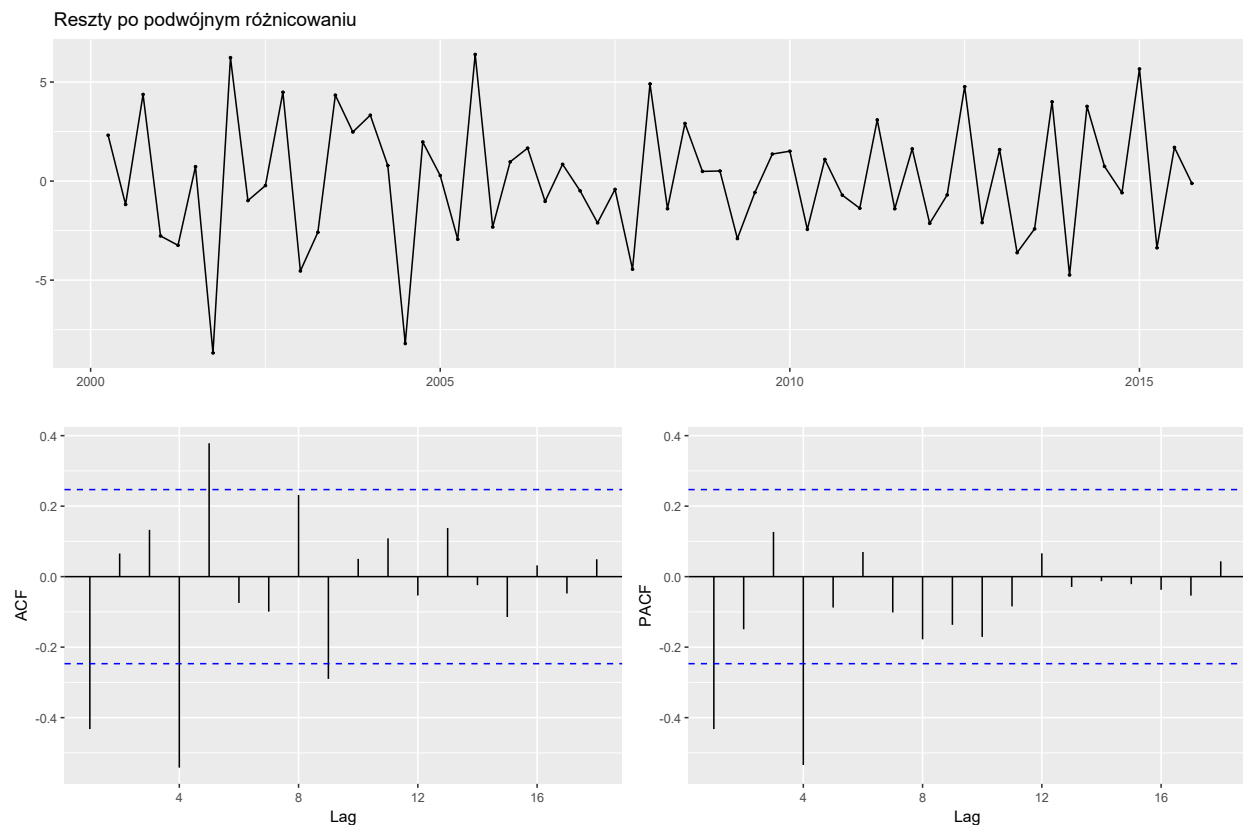
Zastosowanie transformacji przyniosło poprawę. Górny wykres, pokazuje, że wariancja reszt została ustabilizowana. Zniknął nielosowy wzorec, a reszty oscylują teraz wokół zera w miarę równomiernie.

2.3.4 Porównanie z metodą różnicowania

Jako ostatni etap analizy, porównamy wyniki uzyskane z dekompozycji z inną metodą eliminacji niestacjonarności - z różnicowaniem (funkcja `diff()`). Zamiast modelować trend i sezonowość, spróbujemy je po prostu usunąć, aby uzyskać szereg stacjonarny.

Ponieważ szereg `austourists` ma silny trend oraz wyraźną sezonowość, zastosujemy podwójne różnicowanie. Naszym celem jest sprawdzenie, czy otrzymany w ten sposób szereg reszt będzie można uznać za stacjonarny, tzn. czy jego wykresy ACF i PACF będą przypominać biały szum.

```
library(magrittr)
austourists %>%
  diff(lag = 4, differences = 1) %>%
  diff(lag = 1, differences = 1) %>%
  ggtsdisplay(main = "Reszty po podwójnym różnicowaniu")
```



Różnicowanie skutecznie usunęło niestacjonarność. Górny wykres pokazuje szereg, który oscyluje wokół stałego poziomu i ma stabilną wariancję. Oznacza to, że udało się usunąć zarówno trend, jak i sezonowość. Mimo, że szereg jest stacjonarny, to nie jest to jednak białym szumem. Dolne wykresy pokazują, że wciąż pozostała w nim pewna autokorelacja.

2.3.5 Wnioski

- Wstępna analiza graficzna wykazała, że szereg `austourists` jest niestacjonarny. Charakteryzuje się on wyraźnym trendem wzrostowym, silną sezonowością oraz niejednorodną wariancją.
- Porównanie metod dekompozycji pokazało, że standardowe modele addytywne były źle dopasowane. Wszystkie te metody pozostawiały nielosowe wzorce w resztach.
- Testowanie parametrów wewnątrz tych metod, jak zmiana trendu na kwadratowy w `tslm` czy zmiana okien parametrów w `stl`, nie poprawiły znacząco tego problemu niedopasowania.
- Jedynymi metodami dekompozycji, które dały zadowalające rezultaty, były dekompozycja multiplikatywna oraz model `tslm` z transformacją Boxa-Coxa.
- Porównanie dekompozycji z różnicowaniem wykazało, że podwójne różnicowanie również było skuteczną metodą eliminacji trendu i sezonowości.
- Ostatecznie, ani dekompozycja, ani różnicowanie nie dały w wyniku idealnego białego szumu. W obu przypadkach w resztach pozostała pewna autokorelacja widoczna na wykresach `ACF` i `PACF`, co sugeruje, że do pełnego opisu danych potrzebny byłby jeszcze bardziej złożony model dekompozycji.