Politechnika Wrocławska

Wydział Matematyki
Kierunek studiów: Matematyka i Statystyka
Specjalność: Statystyka i analiza danych

Praca dyplomowa – licencjacka

WYBRANE METODY IDENTYFIKACJI OBSERWACJI ODSTAJĄCYCH W SZEREGACH CZASOWYCH

Piotr Migdałek

słowa kluczowe:
szeregi czasowe, obserwacje odstające, modele ARIMA, dekompozycja STL, analiza interwencji, HOT-SAX, profil macierzowy

krótkie streszczenie:
Niniejsza praca przedstawia wybrane techniki detekcji oraz modelowania obserwacji odstających w szeregach czasowych. Jej istotą jest wykorzystanie narzędzi statystycznych oraz algorytmów do analizy realnych problemów, z którymi zmagają się na co dzień specjaliści z różnych dziedzin nauki, przemysłu i biznesu. Struktura pracy obejmuje niezbędne wprowadzenie teoretyczne oraz kolejne rozdziały, w których skupiono się na na konkretnych postaciach anomalii: interwencjach, punktowych obserwacjach odstających oraz sekwencjach.

Opiekun pracy dyplomowej	dr inż. Adam Zagdański	.....	.....
	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis

*Do celów archiwalnych pracę dyplomową zakwalifikowano do:**

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

** niepotrzebne skreślić*

pieczątka wydziałowa

Wrocław, rok 2022

Wrocław University
of Science and Technology

Faculty of Pure and Applied Mathematics
Field of study: Mathematics and Statistics
Specialty: Statistics and Data Analysis

Bachelor's Thesis

SELECTED OUTLIER DETECTION METHODS IN TIME SERIES DATA

Piotr Migdałek

keywords:
time series, outliers, ARIMA models, STL decomposition, intervention analysis, HOT-SAX, Matrix profile

short summary:
The thesis discusses selected outlier detection methods in time series data. The core of presented work consists of practical analysis of real-world problems from different fields of science, industry and business using statistical and algorithmic approaches. The thesis comprises of necessary theoretical background followed by chapters covering particular types of anomalies: interventions, point outliers and sequences.

Supervisor	dr inż. Adam Zagdański	.....	.....
	Title/degree/name and surname	grade	signature

*For the purposes of archival thesis qualified to:**

a) category A (perpetual files)

b) category BE 50 (subject to expertise after 50 years)

** delete as appropriate*

stamp of the faculty

Wrocław, 2022

Spis treści

Wstęp **3**

1 Podstawowe pojęcia związane z analizą szeregów czasowych **5**
1.1 Czym są szeregi czasowe? . . . 5
1.2 Funkcja autokorelacji (ACF) oraz cząstkowej autokorelacji (PACF) . . . 6
1.3 Modele z rodziny ARIMA . . . 8
1.3.1 Modele stacjonarne . . . 8
1.3.2 Modele ARIMA oraz SARIMA . . . 10
1.4 Testy losowości oraz normalności . . . 12
1.5 Dekompozycja STL . . . 13

2 Zastosowanie analizy interwencji w wykrywaniu obserwacji odstających **17**
2.1 Analiza interwencji a wykrywanie obserwacji odstających . . . 17
2.2 Taksonomia punktowych obserwacji odstających . . . 18
2.3 Metoda połączonej estymacji parametrów modelu ARIMA oraz efektów obserwacji odstających . . . 23
2.3.1 Estymowanie efektów obserwacji odstających . . . 23
2.3.2 Metoda estymacji parametrów modelu ARIMA w przypadku obecności wielu obserwacji odstających . . . 25
2.4 Studium przypadku: modelowanie ARIMA uwzględniające efekty obserwacji odstających . . . 26
2.4.1 Opis schematu analizy . . . 26
2.4.2 Opis danych . . . 27
2.4.3 Transformacja Boxa-Coxa . . . 28
2.4.4 Identyfikacja modeli w oparciu o funkcje PACF oraz ACF . . . 29
2.4.5 Automatyczna procedura wyboru optymalnego modelu ARIMA . . . 30
2.4.6 Automatyczna procedura wyboru optymalnego modelu ARIMA z uwzględnieniem efektów obserwacji odstających . . . 31
2.4.7 Przegląd zidentyfikowanych modeli . . . 36
2.4.8 Analiza reszt . . . 37
2.4.9 Testowanie istotności współczynników . . . 40
2.4.10 Analiza dokładności prognoz . . . 40

3 Metody wykrywania punktowych obserwacji odstających **43**
3.1 Jednostronna oraz dwustronna metoda ruchomego okna . . . 43
3.1.1 Metoda jednostronnej oraz dwustronnej mediany . . . 43
3.1.2 Metoda ruchomego okna . . . 44
3.2 Dekompozycja MSTL oraz reguła Tukeya . . . 44
3.3 S-H-ESD . . . 45
3.3.1 Test Grubbsa oraz test Rosnera . . . 45
3.3.2 Modyfikacje testu Rosnera – S-ESD i S-H-ESD . . . 46
3.4 Model mieszanin gaussowskich oraz klasteryzacja . . . 47
3.5 Porównanie skuteczności metod wykrywania anomalii punktowych . . . 49
3.5.1 Zestawienie porównywanych algorytmów . . . 49
3.5.2 Miary wykorzystane do oceny skuteczności i porównania metod . . . 50

3.5.3 Opis danych . . . 51
3.5.4 Dane produkcyjne . . . 52
3.5.5 Dane symulowane . . . 53
3.5.6 Wnioski dotyczącego analizy przedstawionych algorytmów detekcji oraz propozycje dalszych badań . . . 55

4 Metody wykrywania anomalnych sekwencji 59
4.1 Brute force . . . 60
4.2 HOT-SAX . . . 60
4.2.1 Reguły heurystyczne . . . 61
4.2.2 SAX . . . 61
4.2.3 Aproksymacja magicznej reguły heurystycznej (*magic heuristic*) . . . 62
4.3 Profil macierzowy . . . 64
4.3.1 Algorytmy stosowane do obliczania profilu macierzowego profil macierzowy . . . 65
4.4 Studium przypadku: wykrywanie anomalnych sekwencji dla danych medycznych . . . 66
4.4.1 Cel analizy . . . 66
4.4.2 Porównywane algorytmy . . . 66
4.4.3 Wykrywania anomalnych uderzeń serca w EKG . . . 67
4.4.4 Podsumowanie analizy . . . 72

Podsumowanie 73

Dodatek: Wykorzystane oprogramowanie 75
One-sided window method . . . 76
Two-sided window method . . . 77

Bibliografia 77

Wstęp

Detekcja obserwacji odstających w szeregach czasowych jest współcześnie jednym z najistotniejszych zadań analizy danych. Anomalie występujące w procesach produkcyjnych, transakcjach bankowych czy rynkach finansowych mogą nieść za sobą poważne konsekwencje, gdy nie zostaną odpowiednio wcześniej wykryte oraz przeanalizowane.

Motywacją do powstania niniejszej pracy była chęć przedstawienia szerokiego spektrum technik detekcji i modelowania obserwacji odstających różnego typu oraz prezentacja wyników autorskich badań przedstawiających wykorzystanie omawianych narzędzi do analizy zagadnień związanych m.in. z medycyną oraz ochroną środowiska. Ważnym celem przeprowadzonych badań była także szczegółowa analiza porównawcza rozważanych metod, która pozwoliła na praktyczną weryfikację skuteczności poszczególnych algorytmów oraz zidentyfikowanie ich najważniejszych zalet i ograniczeń.

Niniejsza praca składa się z czterech rozdziałów. W Rozdziale 1 wprowadzone są podstawowe pojęcia oraz techniki związane z analizą szeregów czasowych, ze szczególnym uwzględnieniem modelowania autokorelacyjnego.

Rozdział 2 poświęcony został w całości problematyce związanej z użyciem modeli wykorzystywanych w analizie interwencji w celu sparametryzowania różnych efektów obserwacji odstających, by finalnie zaproponować procedurę połączonej estymacji parametrów modelu oraz efektów obserwacji odstających. Kończące ten rozdział studium przypadku szczegółowo ilustruje rozszerzoną procedurę modelowania szeregów czasowych z uwzględnieniem wpływu obserwacji odstających, wykorzystując w tym celu dane dotyczące emisji dwutlenku węgla w Polsce na przestrzeni ostaniach 150 lat.

Treścią Rozdziału 3 jest wykrywanie punktowych obserwacji odstających. Zawiera on obszerną analizę porównawczą różnorodnych metodologicznie algorytmów detekcji pod kątem ich efektywności oraz czasu działania.

W ostatnim, 4 Rozdziale pracy omówiono zagadnienia związane z wykrywaniem anomalnych sekwencji. Dokładność oraz czas detekcji wprowadzonych metod identyfikacji anomalnych podciągów jest analizowana na przykładzie danych z badania elektrokardiografem.

Wszystkie wyniki oraz koncepcje analiz wraz z ilustracjami oraz tabelami zamieszczonymi w niniejszej pracy zostały w całości przygotowane przez autora. Wszystkie analizy przeprowadzono na bazie pakietu statystycznego `R`. Więcej informacji nt. wykorzystanych narzędzi można znaleźć w dodatku umieszczonym na końcu niniejszej pracy.

Rozdział 1

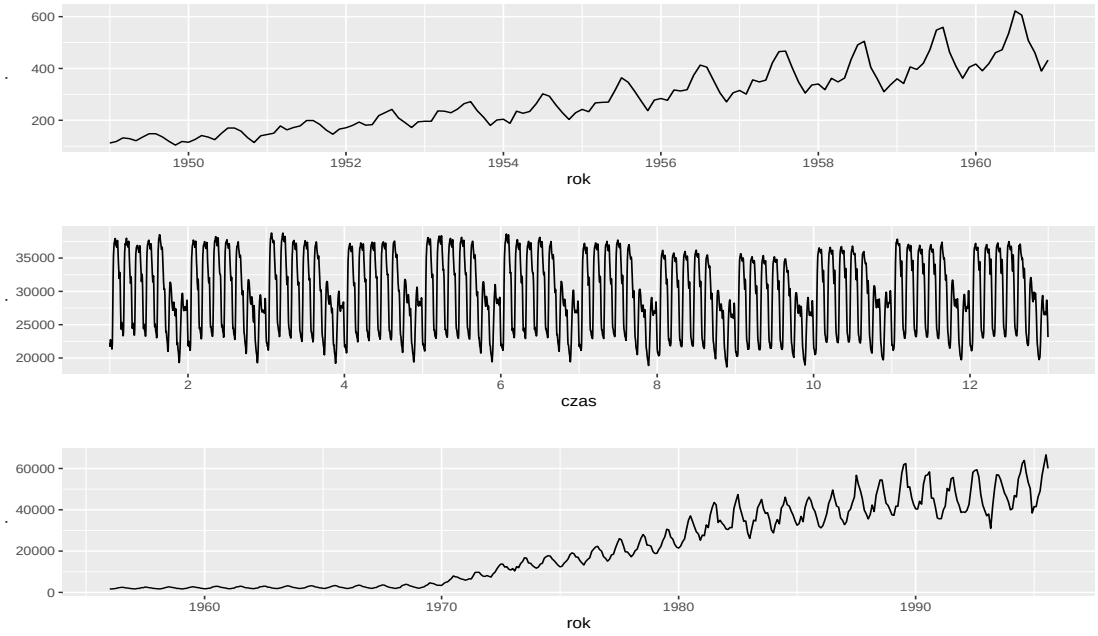
Podstawowe pojęcia związane z analizą szeregów czasowych

Współcześnie niemal w każdym sektorze występuje potrzeba analizy danych, które są indeksowane czasem. W poniższym rozdziale omówione zostaną podstawowe metody analizy oraz modelowania danych tego typu nazywanych inaczej szeregami czasowymi. Zaprezentowane podejście będzie w szczególności skupiać się na modelowaniu autokorelacyjnym, czyli modelowaniu potencjalnego wpływu zaobserwowanych historycznych wartości na obecne obserwacje szeregu. Dokładniejszy opis poruszonych zagadnień oraz tematyki związanej z analizą szeregów czasowych można znaleźć w monografiach [9] oraz [39].

1.1 Czym są szeregi czasowe?

Definicja 1.1. Szeregiem czasowym (ang. *time series*) Y_t ($t \in I$, gdzie I jest zbiorem indeksującym; w praktyce najczęściej jest to zbiór liczb naturalnych) nazywamy realizację procesu stochastycznego, którego dziedziną jest czas. Mniej formalnie to obserwacje pewnej wielkości zarejestrowane w kolejnych (zazwyczaj regularnych) odstępach czasu.

Przykłady typowych szeregów czasowych przedstawia rysunek 1.1.



Rysunek 1.1: Przykłady szeregów czasowych; (od góry) miesięczna liczba zagranicznych pasażerów linii lotniczych w USA w latach 1949-1960, zapotrzebowanie na energię elektryczną w Anglii oraz Walii od 5 czerwca 2000 do 27 sierpnia 2000 (dane zbierane co pół godziny), miesięczna produkcja paliwa w Australii w latach 1956-1995.

Powyższe przykłady szeregów czasowych ilustrują ważne charakterystyki, których identyfikacja jest istotna na etapie właściwego modelowania. Jedną z najważniejszych własności szeregu jest sezonowość bądź cykliczność, którą możemy zaobserwować dla każdego z trzech szeregów przedstawionych na rysunku 1.1. Szeregi dotyczące liczby pasażerów oraz produkcji paliwa zawierają w sobie również komponent trendu deterministycznego, który dla obydwu przykładów jest rosnący, jeśli chodzi o długoterminową tendencję.

1.2 Funkcja autokorelacji (ACF) oraz cząstkowej autokorelacji (PACF)

Analizując szeregi czasowe, jedną z najistotniejszych kwestii jest poprawne zidentyfikowanie oraz modelowanie występujących zależności, tzn. (potencjalnego) wpływu historycznych obserwacji szeregu czasowego na jego obecną wartość. Bardziej formalnie, staramy się rozstrzygnąć, czy występuje i jak silna jest autokorelacja – korelacja czasowa. Często taka zależność jest konsekwencją występowania w danych trendu czy sezonowości, które można łatwo zidentyfikować za pomocą podstawowych narzędzi graficznych. Natomiast, by uchwycić nieco mniej oczywiste zależności potrzebne są bardziej formalne narzędzia – funkcje autokorelacji (ACF) oraz autokorelacji cząstkowej (PACF).

Definicja 1.2. Funkcję autokorelacji definiujemy jako

$$ACF(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}, \quad h = 0, 1, 2, \ldots, n-1, \tag{1.1}$$

gdzie $\hat{\gamma}(h)$ jest próbkową funkcją autokowariancji

$$\hat{\gamma}(h) = \frac{1}{n}\sum_{j=1}^{n-h}(Y_{j+h} - \overline{Y})(Y_j - \overline{Y}), \quad h = 0, 1, 2, \ldots, n-1, \tag{1.2}$$

gdzie $\overline{Y}$ to średnia próbkowa.

ACF(h) (ang. *AutoCorrelation Function*) jest miarą korelacji liniowej pomiędzy obserwacjami szeregu czasowego odległymi od siebie o h jednostek czasowych; h jest często nazywany parametrem opóźnienia (ang. *lag*). Funkcja ACF jest uogólnieniem współczynnika korelacji próbkowej oraz oszacowaniem teoretycznej funkcji autokorelacji $\rho(h) = Corr(Y_{t+h}, Y_t)$.

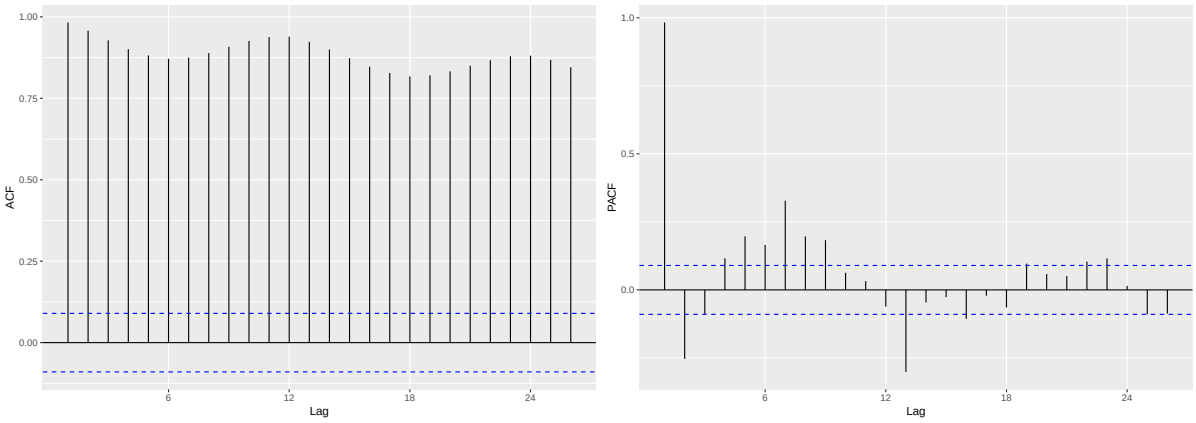
Funkcją silnie związaną z ACF jest PACF (ang. *Partial AutoCorrelation Function*), która również określa zależność poszczególnych obserwacji, lecz robi to w sposób bezwzględny – pomijając korelację pochodzącą z obserwacji znajdujących się pomiędzy badanymi wartościami.

Definicja 1.3. Funkcję próbkowej autokorelacji cząstkowej definiujemy jako oszacowanie teoretycznej funkcji autokorelacji cząstkowej

$$\begin{aligned} &\alpha(h) = Corr(Y_{h+1} - P_{\overline{sp}(1,Y_2,\ldots,Y_h)}Y_{h+1},\ Y_1 - P_{\overline{sp}(1,Y_2,\ldots,Y_h)}Y_1), \quad h \geq 2 \\ &\alpha(1) = Corr(Y_2, Y_1) = \rho(1), \end{aligned} \tag{1.3}$$

gdzie $P_{\overline{sp}(1,Y_2,\ldots,Y_h)}$ jest rzutem ortogonalnym na podprzestrzeń liniową wyznaczoną przez ciąg obserwacji $1, Y_2, \ldots, Y_h$.

ACF oraz PACF są podstawowymi narzędziami, które pozwalają na opisanie podstawowych własności badanego szeregu czasowego. W praktyce często wyznacza się je dla kolejnych opóźnień, tworząc wykresy zwane korelogramami. Przykładowy korelogram dla szeregu czasowego przedstawiającego miesięczną produkcję paliwa w Australii w latach 1956-1995 przedstawiono na rysunku 1.2.



Rysunek 1.2: Wartości funkcji ACF oraz PACF dla kolejnych opóźnień szeregu czasowego miesięcznej produkcji paliwa w Australii w latach 1956-1995.

Z powyższych wykresów można wyciągnąć istotne wnioski dotyczące własności badanego szeregu:

- Dodatnia, wolno zanikająca autokorelacja wskazuje na obecność trendu deterministycznego.
- Natomiast cykliczne zanikanie, które możemy zaobserwować na wykresie ACF wskazuje na obecność sezonowości.
- Wartość PACF dla pierwszego opóźnienia bliska 1 sugeruje, że w danych występuje silny trend deterministyczny.
- Szybkość zanikania funkcji ACF oraz PACF pozwala na wstępną identyfikację modeli stacjonarnych (odpowiednio MA i AR) lub niestacjonarnych (po wykonanej operacji różnicowania), o czym więcej w kolejnym podrozdziale.

Zaznaczone na wykresie przerywaną niebieską linią 95% asymptotyczne przedziały ufności są postaci:

$$\left[-\frac{z(97.5)}{\sqrt{n}}, \frac{z(97.5)}{\sqrt{n}}\right], \tag{1.4}$$

gdzie $z(97.5)$ kwantyl rzędu 97.5 standardowego rozkładu normalnego. Przedziały te pozwalają na określenie czy autokorelacja oraz autokorelacja cząstkowa dla wybranego h może być uznana za statystycznie istotną; a zostaje uznana za taką, gdy wykracza poza wyznaczony przedział.

1.3 Modele z rodziny ARIMA

Najpopularniejszymi modelami statystycznymi wykorzystywanymi w analizie szeregów czasowych są modele stacjonarne ARMA oraz niestacjonarne modele ARIMA (odpowiadające modelom ARMA dla odpowiednio przekształconych danych). Mimo, że zostały wprowadzone już niemal pół wieku temu, w dalszym ciągu ta rodzina modeli liniowych cieszy się sporą popularnością, co może zaskakiwać z uwagi na rozwój w ostatnim czasie skomplikowanych i wysoce skutecznych technik uczenia maszynowego. Niemniej klarowna struktura matematyczna, przejrzystość procedury identyfikacji modelu oraz estymacji parametrów jest ogromnym atutem modeli ARIMA. Łatwość interpretacji w połączeniu z dużą skutecznością sprawia, że modele te często wykorzystywane są również w zastosowaniach biznesowych.

1.3.1 Modele stacjonarne

Podstawową, lecz dalej szeroko używaną klasą modeli jest klasa modeli stacjonarnych. Szereg nazwiemy stacjonarnym, gdy:

- $Corr(Y_p, Y_k) = Corr(Y_{p+t}, Y_{k+t}) = \rho(k-p) = \rho(h)$ – korelacja zależy tylko od odstępu czasowego h.
- Zarówno wartość oczekiwana EY_t, jak i wariancja $VarY_t$ są stałe (nie zależą od t).

Zauważmy, że dla poprawności powyższej definicji koniecznie jest także, aby Y_t był szeregiem drugiego rzędu, to znaczy $EY_t^2 < \infty$. Zatem szeregi przedstawione na rysunku 1.1 nie są stacjonarne, gdyż funkcja średniej dla wszystkich wykresów zmienia się wraz z t. Ponadto, wariancja nie jest jednorodna w czasie za sprawą efektów cyklicznych oraz sezonowych. W praktyce, gdy na korelogramie dla funkcji ACF oraz PACF wartości dla kolejnych opóźnień szybko zanikają, ostatecznie nie wychodząc poza przedziały ufności, wtedy można przypuszczać, że badany szereg jest stacjonarny; na rysunku 1.2 wartości ACF dla kolejnych opóźnień są istotne oraz bardzo wolno zanikają – szereg jest zatem niestacjonarny.

Powyższe własności opisują formalnie stacjonarność w sensie słabym (kowariancyjnym), którą przujmuje się ze względów praktycznych, gdyż jest bardziej ogólnym pojęciem. Natomiast stacjonarność w sensie ścisłym zakłada równość rozkładów wielowymiarowych dla dowolnego opóźnienia h, czyli $(Y_1, \ldots, Y_n) \stackrel{d}{=} (Y_{1+h}, \ldots, Y_{n+h})$.

Oczywiście istnieją bardziej formalne narzędzia, które pozwalają na zbadanie, czy dany szereg jest stacjonarny. Najpopularniejszym testem stacjonarności jest test ADF (ang. *Augmented Dickey-Fuller test*) [12], gdzie tak naprawdę weryfikuje się obecność pierwiastka jednostkowego (niestacjonarność).

Najbardziej popularne modele stacjonarne to:

- Biały szum – $WN(\sigma^2)$,
- $AR(p)$ – model autoregresji rzędu p,
- $MA(q)$ – model ruchomej średniej rzędu q,
- $ARMA(p, q)$ – model mieszany łączący segment autoregresyjny oraz ruchomej średniej,
- $ARMA(P, Q)[s]$ – sezonowy model $ARMA$,

- $ARMA(p,q)(P,Q)[s]$ – pełny model $ARMA$, który również uwzględnia zależności dla opóźnień o okresie s.

Najbardziej podstawowym modelem jest $WN(\sigma^2) = ARMA(0,0)$, czyli biały szum, który będziemy dalej nazywali zakłóceniem losowym lub składnikiem resztowym i oznaczali najczęściej jako Z_t.

Definicja 1.4. Białym szumem nazywamy ciąg zmiennych nieskorelowanych o jednakowym rozkładzie, o średniej 0 oraz wariancji σ^2.

Dla białego szumu korelacja względem dowolnej chwili i dowolnego odstępu czasowego jest równa 0. W praktyce spotykamy się jednak najczęściej ze statystycznie istotną (niezerową) autokorelacją, którą chcemy odpowiednio modelować. Podstawowymi modelami, którymi można się wtedy posłużyć są modele autoregresji oraz ruchomej średniej.

Definicja 1.5. Modelem autoregresji rzędu p nazwiemy stacjonarny szereg czasowy Y_t, który jest postaci:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + Z_t, \tag{1.5}$$

gdzie Z_t jest wyżej zdefiniowanym białym szumem.

Model $AR(p)$ można skojarzyć z klasycznym modelem regresji liniowej, lecz rolę zmiennych objaśniających w tym przypadku pełnią opóźnione wartości szeregu, natomiast objaśniana przy ich pomocy jest wartość w kolejnej chwili t. W modelu $MA(q)$ zachodzi podobne liniowe równanie różnicowe, ale dla szeregu reszt.

Definicja 1.6. Modelem ruchomej średniej rzędu q nazwiemy stacjonarny szereg czasowy Y_t, który jest postaci:

$$Y_t = \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \ldots + \theta_q Z_{t-q} + Z_t. \tag{1.6}$$

W praktyce, odpowiednio dobre dopasowanie modeli AR(p) oraz MA(q) często uzyskuje się wybierając wysoki rząd modelu (odpowiednio p lub q). By wybrać odpowiedni rząd (dla szeregu stacjonarnego) należy przeanalizować wyznaczone korelogramy dla funkcji ACF oraz PACF; rząd modelu $MA(q)$ wskazuje wartość opóźnienia dla ostatniej istotnej (wykraczająca poza przedziały ufności) autokorelacji, analogicznie dla modelu $AR(p)$ można określić rząd p w ten sam sposób używając korelogramu funkcji PACF. Użycie modelu mieszanego $ARMA(p,q)$ pozwala na uzyskanie adekwatnego opisu występujących zależności przy jednoczesnym zmniejszeniu liczby współczynników (zredukowanie rzędów p oraz q).

Definicja 1.7. Modelem autoregresji ruchomej średniej $ARMA(p,q)$ nazwiemy stacjonarny szereg czasowy Y_t, który jest postaci:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \ldots + \theta_q Z_{t-q} + Z_t, \tag{1.7}$$

gdzie $Z_t \sim WN(\sigma^2)$.

Używając operatora przesunięcia wstecz (ang. *backward shift*): $BY_t = Y_{t-1}$, można zapisać model $ARMA(p,q)$ w skróconej postaci:

$$\phi(B) Y_t = \theta(B) Z_t, \tag{1.8}$$

gdzie $\phi(z)$ oraz $\theta(z)$ to wielomiany autoregresyjny oraz ruchomej średniej zdefiniowane tak jak poniżej:

$$\begin{aligned}\phi(z) &= 1 - \phi_1 z - \ldots - \phi_p z^p, \\ \theta(z) &= 1 + \theta_1 z + \ldots + \theta_q z^q.\end{aligned} \tag{1.9}$$

Aby w Definicji 1.5 i 1.7 spełnione było założenie o stacjonarności, konieczne są dodatkowe ograniczenia dla współczynników autoregresyjnych, tzn. wielomian $\phi(z)$ nie może mieć pierwiastków na okręgu jednostkowym ($\phi(z) \neq 0$ dla $|z| = 1$).

Często istotne korelacje występują dla wielokrotności pewnego opóźnienia s, wtedy by modelować ten rodzaj zależności najlepiej posłużyć się sezonowym modelem $ARMA(P, Q)[s]$.

Definicja 1.8. Sezonowym modelem autoregresji ruchomej średniej $ARMA(P, Q)[s]$ nazwiemy stacjonarny szereg czasowy Y_t, który spełnia równanie:

$$\Phi(B^s)Y_t = \Theta(B^s)Z_t, \tag{1.10}$$

gdzie

$$\begin{aligned}\Phi(B^s) &= 1 - \Phi_1 B^s - \ldots - \Phi_P B^{Ps}, \\ \Theta(B^s) &= 1 + \Theta_1 B^s + \ldots + \Theta_Q B^{Qs}.\end{aligned} \tag{1.11}$$

Nie należy mylić modelowania autokorelacji dla opóźnień sezonowych, które stosuje powyższy model, z trwałymi wzorcami sezonowymi, które świadczą o niestacjonarności szeregu, co uniemożliwia dopasowanie modeli stacjonarnych.

Naturalnym uogólnieniem modeli $ARMA(p, q)$ oraz $ARMA(P, Q)[s]$ jest pełny model $ARMA(p, q)(P, Q)[s]$, który obejmuje jako odpowiednie przypadki wszystkie wcześniej omówione modele stacjonarne.

Definicja 1.9. Modelem autoregresji ruchomej średniej $ARMA(p, q)(P, Q)[s]$ nazwiemy stacjonarny szereg czasowy Y_t, który spełnia zależność:

$$\Phi(B^s)\phi(B)Y_t = \Theta(B^s)\theta(B)Z_t. \tag{1.12}$$

Warto nadmienić, że notacyjnie zachodzą równości $ARMA(P, Q)[s] = ARMA(P \cdot s, Q \cdot s)$ oraz $ARMA(p, q)(P, Q)[s] = ARMA(p + P \cdot s, q + Q \cdot s)$. Bogatą taksonomię testów stacjonarności oraz szerszy opis modeli stacjonarnych oraz ich własności można znaleźć w monografii [9].

1.3.2 Modele ARIMA oraz SARIMA

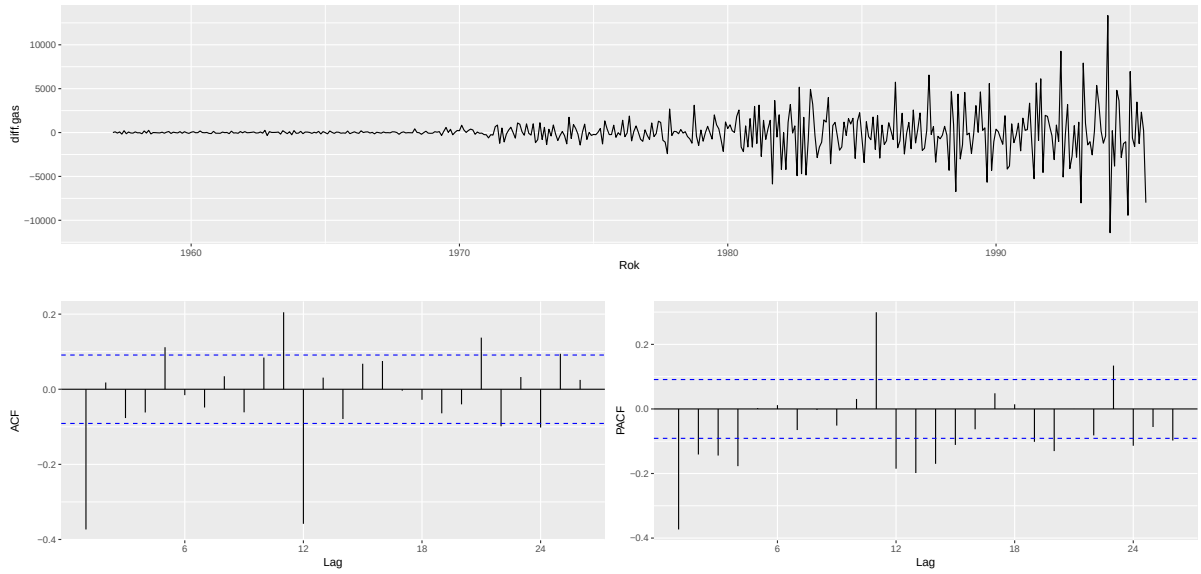
Spotykane w praktyce szeregi czasowe są najczęściej szeregami niestacjonarnymi, w szczególności charakteryzują się obecnością sezonowości lub deterministycznego trendu. W konsekwencji, zastosowanie modeli stacjonarnych nie jest wówczas możliwe i konieczne jest zastosowanie odpowiednich transformacji wyjściowych danych. Kluczową transformacją, która sprowadza dane do postaci stacjonarnej jest operacja różnicowania.

Definicja 1.10. Operator różnicowania z opóźnieniem h definiuje się jako ∇_h, gdzie zróżnicowany szereg Y_t jest postaci:

$$\nabla_h Y_t = (1 - B^h)Y_t = Y_t - Y_{t-h}, \tag{1.13}$$

gdzie B jest wcześniej zdefiniowanym operatorem przesunięcia wstecz.

W kontekście praktycznym stosujemy najczęściej $h = 1$ (redukowanie trendu deterministycznego) oraz $h = s$ (redukowanie sezonowości o okresie s). Dla przykładu analizując dane przedstawiające miesięczną produkcje paliwa w Australii w latach 1956-1995 (przedstawione na rysunku 1.1), przy pomocy funkcji ACF oraz PACF (rysunek 1.2) można jednoznacznie stwierdzić obecność wzorców sezonowych oraz deterministycznej tendencji wzrostowej. Na rysunku 1.3 przedstawione są te same dane oraz funkcje ACF oraz PACF po wykonaniu operacji różnicowania z opóźnieniem 12 oraz następnie z opóźnieniem 1. Analizując poniższy rysunek, widać, że w danych w dalszym ciągu występują istotne korelacje dla kilku opóźnień. Wzorce sezonowe oraz trend nie są obecne, niemniej w danych można zauważyć efekty heteroskedastyczne.



Rysunek 1.3: Dwukrotnie zróżnicowany szereg (najpierw z opóźnieniem 12, następnie z opóźnieniem 1) miesięcznej produkcji paliwa w Australii w latach 1956-1995 oraz korelogramy funkcji ACF oraz PACF dla kolejnych opóźnień przekształconego szeregu.

Wprowadzenie operacji różnicowania pozwala na zdefiniowane klasy modeli niestacjonarnych $ARIMA(p, d, q)$ (ang. *AutoRegressive Integrated Moving Average*), dla których modyfikacją względem modelu $ARMA(p, q)$ będzie uwzględnienie d-krotnego różnicowania z opóźnieniem 1.

Definicja 1.11. Modelem $ARIMA(p, d, q)$ nazwiemy szereg czasowy, który spełnia równanie:

$$\phi(B)\nabla^d Y_t = \theta(B)Z_t, \tag{1.14}$$

gdzie ∇^d oznacza d-krotne różnicowanie z opóźnieniem 1; wielomiany $\phi(B)$ oraz $\theta(B)$ są określone jak w (1.9).

W pełnej ogólności można mówić o modelu $SARIMA$ (ang. *Seasonal ARIMA*), który jest naturalnym rozszerzeniem pełnego modelu $ARMA$ (Definicja 1.7) o odpowiednie różnicowanie (zwykłe i sezonowe) uwzględniające niestacjonarność.

Definicja 1.12. Modelem $SARIMA(p, d, q)(P, D, Q)[s]$ nazwiemy szereg czasowy, który spełnia równanie:

$$\Phi(B^s)\phi(B)\nabla^d\nabla_s^D Y_t = \Theta(B^s)\theta(B)Z_t, \tag{1.15}$$

gdzie ∇^d oznacza d-krotne różnicowanie z opóźnieniem 1, ∇_s^D oznacza D-krotne różnicowanie z opóźnieniem s; wielomiany $\phi(B)$ oraz $\theta(B)$ są określone jak w (1.9), natomiast wielomiany $\Theta(B^s)$ oraz $\Phi(B^s)$ jak w (1.11).

Warto dodać, że model $SARIMA(p,0,q)(P,0,Q)[s]$ jest równoważny modelowi stacjonarnemu $ARMA(p,q)(P,Q)[s]$ oraz przez pełny model $SARIMA$ można wyrazić każdy z wcześniej przedstawionych modeli stacjonarnych oraz niestacjonarnych. Więcej szczegółów na temat własności teoretycznych modeli niestacjonarnych można znaleźć w monografii [9].

1.4 Testy losowości oraz normalności

Ważnym elementem weryfikacji poprawności dopasowania modelu jest analiza jego reszt. Podstawowymi narzędziami graficznym, które można w tym celu wykorzystać są wykres funkcji ACF, histogram reszt lub estymator jądrowy gęstości czy najzwyklejszy wykres szeregu reszt. Graficzne metody pozwalają na wstępne zbadanie hipotez o losowości oraz normalności reszt.

Bardziej formalnymi narzędziami diagnostycznymi używanymi do diagnostyki reszt są odpowiednie testy statystyczne. Najważniejszą własnością, którą koniecznie muszą cechować się reszty poprawnie dopasowanego modelu jest brak autokorelacji. W diagnostyce (oprócz wykresu funkcji ACF) przydatnymi narzędziami jeśli chodzi o ten aspekt będą klasyczne testy losowości – test Boxa-Pierce'a (B-P) oraz test Ljungi-Boxa (L-B) (często nazywane w literaturze zbiorczo testami portmanteau). Statystyka testowa testu B-P oparta jest na funkcji ACF dla kilku/kilkudziesięciu początkowych opóźnień. Ma ona postać:

$$Q_{B-P} = n \sum_{i=1}^{h} \hat{\rho}^2(i), \tag{1.16}$$

gdzie h to maksymalne opóźnienie, natomiast $\hat{\rho}$ – próbkowa ACF dla i-tego opóźnienia. Zbyt duże wartości Q_{B-P} oznaczają, że wartości ACF są zbyt duże, by uznać reszty szeregu za realizację nieskorelowanej sekwencji.

W praktyce częściej używanym testem jest zmodyfikowana wersja testu B-P – test L-B. Jest tak ze względu na lepsze przybliżenie statystyki testowej Q_{L-B} rozkładem asymptotycznym. Jest ona postaci:

$$Q_{L-B} = n(n+2) \sum_{i=1}^{h} \hat{\rho}^2(i)/(n-i). \tag{1.17}$$

Hipotezą H_0 dla powyższych testów jest losowość reszt, natomiast H_1 mówi o obecności istotnej korelacji w szeregu reszt. Wiedząc, że statystyki Q_{L-B} oraz Q_{B-P} mają asymptotyczny rozkład chi-kwadrat z h stopniami swobody, decyzje o odrzuceniu bądź przyjęciu H_0 podejmujemy na podstawie wyznaczonych p-wartości.

Test McLeod-Li (M-L) jest testem L-B dla kwadratów reszt. Pozwala on zbadać, czy w szeregu reszt występują efekty heteroskedastyczne – dokładniej, czy dla kwadratów reszt obserwujemy istotną autokorelację. Podobnie jak poprzednio, rozkład statystyki testowej aproksymujemy asymptotycznym rozkładem chi-kwadrat o h stopniach swobody.

Ważnym elementem analizy reszt przy użyciu tych trzech testów jest korekcja liczby stopni swobody (domyślnie h). Należy skorygować liczbę stopni swobody odejmując liczbę estymowanych parametrów modelu (k) od wartości ustalonego maksymalnego opóźnienia h.

Na kolejną grupę testów losowości składają się test punktów zwrotnych (TP), test rangowy (R) oraz test znaków (DS). Pierwszy z wymienionych opiera się na idei, że ciąg i.i.d. nie może mieć ani zbyt dużej, ani zbyt małej liczby punktów zwrotnych. Punktem zwrotnym nazwiemy obserwacje, która spełnia zależność: $Y_{i-1} > Y_i$ i $Y_i < Y_{i+1}$ lub $Y_{i-1} < Y_i$ i $Y_i > Y_{i+1}$. Duża wartość $T - E[T]$ (T – liczba punktów zwrotnych) wskazuje na to, że fluktuacje w szeregu pojawiają się częściej niż jest to oczekiwane dla szeregu losowego. Z drugiej strony, dużo mniejsza od zera wartość $T - E[T]$ wskazuje na korelację między sąsiednimi obserwacjami szeregu reszt.

Test rangowy opiera się na statystyce testowej P będącej liczbą par (i, j), takich że: $Y_j > Y_i$ oraz $j > i$, dla $i = 1, \ldots n - 1$. Duża dodatnia (lub ujemna) wartość $P - E[P]$ wskazuje na obecność rosnącego (lub malejącego) trendu w danych.

Test znaków opiera się na statystyce testowej S – liczba obserwacji, dla których ∇Y_t (∇ – operator różnicowania z opóźnieniem 1) ma dodatni znak. Analogicznie do testu rangowego, duża dodatnia (lub ujemna) wartość $S - E[S]$ wskazuje na obecność rosnącego (lub malejącego) trendu w danych.

Pożądaną własnością szeregów reszt jest ich normalność, gdyż konstrukcja przedziałów predykcyjnych czy testu istotności współczynników modelu opiera się na założeniu normalności szeregu reszt (gdy założenie to nie jest spełnione uzyskuje się gorszą aproksymację rozkładem asymptotycznym). Testem statystycznym często używanym do weryfikacji normalności reszt jest test Jarque-Bera, który jest również popularnym wśród praktyków narzędziem wykrywania odstępstw od normalności dla szeregów finansowych. Test opiera się na standaryzowanych momentach rozkładu normalnego i fakcie, że w teorii przyjmują one odpowiednio wartości 0 (skośność) oraz 3 (kurtoza) – hipoteza zerowa. Statystyka testowa testu J-B ma postać:

$$N_{J-B} = \frac{n}{6}(S^2 + \frac{1}{4}(K-3)^2), \tag{1.18}$$

gdzie S to próbkowy współczynnik asymetrii, natomiast K - próbkowy współczynnik ekscesu. Innymi popularnymi testami normalności są test Shapiro-Wilka czy test Kołmogorowa-Smirnowa.

Dokładniejszy opis teoretyczny dotyczący powyższych oraz alternatywnych testów losowości oraz normalności można znaleźć w monografii [9].

1.5 Dekompozycja STL

Ideą metod dekompozycji jest rozkład szeregu czasowego na poszczególne składniki (struktura addytywna) bądź czynniki (struktura multiplikatywna), odpowiadające za regularne tendencje występujące w danych, uzupełniając je o element zakłócenia losowego.

Jedną z najbardziej uniwersalnych metod wykorzystywanych do dekompozycji szeregów czasowych jest algorytm STL (*Seasonal and Trend decomposition using Loess*) [13]. Do głównych zalet tego podejścia można zaliczyć m.in. odporność na obserwacje odstające, łatwą kontrolę stopnia wygładzania komponentów oraz obsługę sezonowości o dowolnych okresach.

Używając dekompozycji STL chcemy przedstawić szereg czasowy Y_t w postaci:

$$Y_t = \hat{S}_t + \hat{T}_t + \hat{Z}_t, \tag{1.19}$$

gdzie $\hat{S}_t$ oraz $\hat{T}_t$ to estymowana sezonowość oraz trend deterministyczny, natomiast reszty $\hat{Z}_t$ wyznaczamy jako $\hat{Z}_t = Y_t - \hat{S}_t - \hat{T}_t$. Do estymacji trendu oraz sezonowości – $\hat{S}_t$ oraz $\hat{T}_t$

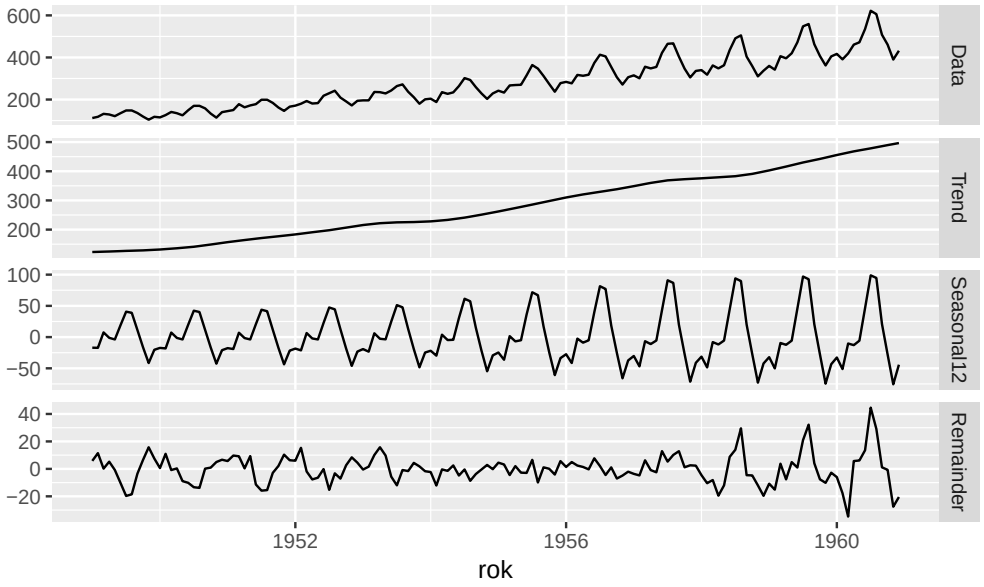
– STL wykorzystuje metodę LOESS (*LOcally Estimated Scatterplot Smoothing*) [14], która jest często stosowanym algorytmem nieparametrycznej estymacji funkcji regresji.

Algorytm STL składa się z dwóch pętli. W pętli zewnętrzna obliczane są wagi obserwacji (ang. *robustness weights*), wykorzystywane do zredukowania wpływu obserwacji odstających, które są używane w kolejnym kroku. Natomiast w pętli wewnętrznej uzyskiwane są coraz dokładniejsze aproksymacje komponentów trendu oraz sezonowości.

Szereg zostaje podzielony na sekwencje (z uwagi na występujące w danych cykle sezonowe), ich liczbę określa ustalony parametr. Procedura estymująca $\hat{S}_t$ oraz $\hat{T}_t$ (pętla wewnętrzna STL) wygląda następująco:

- Usunięcie trendu (z poprzedniej iteracji $\hat{T}_t$, a dla pierwszej iteracji zakładamy, że wynosi 0).
- Wygładzanie poszczególnych sekwencji z wykorzystaniem algorytmu LOESS.
- Zastosowanie filtra dolnoprzepustowego, aby wyestymować pozostały trend: na wcześniej wygładzonych cyklach zostaje zaaplikowana trzykrotnie ruchoma średnia, po czym kolejny raz metoda LOESS.
- Sezonowość w obecnej iteracji określana jest jako różnica wygładzonych sekwencji oraz wyestymowanego w poprzednim kroku trendu..
- Wyznaczona sezonowość zostaje usunięta.
- Trend zostaje wygładzony znów wykorzystując metodę LOESS.

W zewnętrznej pętli algorytmu wyznaczane są wagi obserwacji, które są wykorzystane następnie w algorytmie LOESS (w pętli wewnętrznej), po to by zmniejszyć wpływ obserwacji odstających na estymowany trend oraz sezonowość. Na rysunku 1.4 przedstawiony został przykładowy wynik dekompozycji STL (otrzymane składowe) dla wcześniej wykorzystywanych już danych, zawierających informacje nt. miesięcznej liczby pasażerów linii lotniczych (podrozdział 1.1).



Rysunek 1.4: Składowe uzyskane w wyniku zastosowania dekompozycji STL dla szeregu miesięcznej liczby zagranicznych pasażerów linii lotniczych w USA w latach 1949-1960.

Ze względu na odporność metody STL na obserwacje odstające, wszystkie algorytmy wykrywania punktowych obserwacji odstających wprowadzone w Rozdziale 3 wykorzystują tę metodę do eliminacji trendu czy sezonowości, które mogłyby zakłócać detekcję – same obserwacje odstające są obecne z reguły w komponencie $\hat{Z}_T$.

Chcąc uwzględnić wiele komponentów sezonowych (np. dzienny oraz tygodniowy – jak w danych dotyczących zapotrzebowania na energię elektryczną w Anglii, rys. 1.1), naturalnym uogólnieniem STL jest metoda MSTL [3] iteracyjnie aplikująca w odpowiedni sposób dekompozycję STL, by uzyskać dodatkowe składowe sezonowe.

Rozdział 2

Zastosowanie analizy interwencji w wykrywaniu obserwacji odstających

2.1 Analiza interwencji a wykrywanie obserwacji odstających

Analiza interwencji, wprowadzona przez Boxa i Tiao w 1975 roku [7] proponuje zestaw modeli pozwalających na zbadanie wpływu tak zwanej interwencji na zachowanie szeregu czasowego. Interwencja wpływa na funkcję wartości oczekiwanej lub trend badanego szeregu - ogólniej można mówić o funkcji transferowej f_t, która opisuje efekt zewnętrzny, oddziałujący na obserwowany szereg. Interwencja może być naturalna, np. określone zjawiska wpływające na zmianę populacji danego gatunku lub za sprawą ingerencji człowieka, jak np. wprowadzanie surowszych przepisów ruchu drogowego. W ogólności, szereg czasowy poddany interwencji ma postać:

$$Y_t^* = f_t + Y_t, \tag{2.1}$$

gdzie f_t jest efektem interwencyjnym, natomiast Y_t jest wyjściowym szeregiem, o którym możemy założyć, że jest realizacją procesu SARIMA (równanie 1.12). By dopasować odpowiedni model interwencji używa się danych do chwili T (chwili, w której nastąpiła interwencja), które nazywa się danymi przedinterwencyjnymi oraz zapisuje jako $\{Y_t,\, t < T\}$. Do chwili T można założyć, że $f_t = 0$. Do parametryzacji modeli interwencji używa się funkcji schodkowej (*step function*):

$$S_t^{(T)} = \begin{cases} 1, \text{ gdy } t \geq T \\ 0, \text{ gdy } t < T \end{cases} \tag{2.2}$$

oraz funkcji impulsowej (*pulse function*):

$$P_t^{(T)} = \begin{cases} 1, \text{ gdy } t = T \\ 0, \text{ gdy } t \neq T. \end{cases} \tag{2.3}$$

Warto dodać, że $P_t^{(T)} = \nabla S_t^{(T)}$, gdzie ∇ oznacza operator różnicowania. Jeśli chcemy zapisać natychmiastowy i trwały efekt interwencji, skutkujący zmianą poziomu szeregu, możemy zdefiniować funkcję transferową w (2.1) jako::

$$f_t = \omega S_t^{(T)}. \tag{2.4}$$

Często efekt interwencji obserwowany jest z pewnym opóźnieniem. Definiujemy je jako d jednostek czasu przed wystąpieniem efektu interwencji. Analogicznie trwały efekt zaminy poziomu, który nastąpił z opóźnieniem d można zapisać jako:

$$f_t = \omega S_{t-d}^{(T)}. \tag{2.5}$$

Do modelowania jednorazowej zmiany wartości szeregu, która zachodzi tylko w chwili $t = T$ używamy funkcji impulsowej:

$$f_t = \omega P_t^{(T)}. \tag{2.6}$$

W rzeczywistości interwencja może stopniowo wpływać na szereg czasowy, wtedy f_t można zapisać wykorzystując parametryzację podobną do modelu AR(1):

$$f_t = \delta f_{t-1} + \omega P_t^{(T)}, \tag{2.7}$$

w tym przypadku f_t będzie stopniowo wygasać. Do opisania bardziej skomplikowanych efektów, używa się zdefiniowanego wcześniej operatora przesunięcia wstecz – B. Ponieważ $Bf_t = f_{t-1}$ oraz $BP_t^{(T)} = P_{t-1}^{(T)}$ funkcję transferową (2.7) można zapisać równoważnie jako:

$$f_t = \frac{\omega B}{1 - \delta B} P_t^{(T)}. \tag{2.8}$$

Zauważmy, że w tej konwencji można używać również funkcji schodkowej, gdyż $S_t^{(T)} = \frac{1}{1-B} P_t^{(T)}$. Powyżej przedstawione funkcje transferowe można ze sobą łączyć, tworząc w ten sposób bardziej skomplikowane efekty jak np.:

$$f_t = \left[\omega_2 + \frac{\omega_1 B}{1 - \delta B} + \frac{\omega_2 B}{1 - B}\right] P_t^{(T)}. \tag{2.9}$$

Funkcję transferową można zapisać ogólniej używając specyfikacji podobnej do modelu ARMA(p, q):

$$f_t = \frac{\omega(B)}{\delta(B)} P_t^{(T)}, \tag{2.10}$$

gdzie $\omega(B)$ oraz $\delta(B)$ są wielomianami zmiennej B. Dodatkowo, zakładamy, że pierwiastki wielomianów $\omega(B)$ oraz $\delta(B)$ leżą poza okręgiem jednostkowym.

Fundamentalną różnicą, między procedurami analizy interwencji oraz wykrywania obserwacji odstających jest brak (w przypadku wykrywania anomalii) wiedzy a priori o jakichkolwiek nietypowych zdarzeniach lub działaniach, które mogły wpływać na wyjściowe dane, podczas gdy w analizie interwencji posiadamy wiedzę nt. momentu wystąpienia interwencji. Niemniej efekty zewnętrzne, które docelowo chcemy wykrywać, w istocie będą mieć postać bardzo skomplikowanego modelu interwencji z kilkoma lub kilkudziesięcioma składnikami funkcji transferowej.

2.2 Taksonomia punktowych obserwacji odstających

Aby określić ogólny model dla szeregu czasowego, który jest poddany działaniu niepowtarzającego się efektu zewnętrznego można rozwinąć wzory (2.1) oraz (2.10), tzn. otrzymujemy:

$$Y_t^* = Y_t + \omega \frac{A(B)}{G(B)H(B)} P_t^{(T)}, \tag{2.11}$$

gdzie Y_t jest procesem SARIMA. Parametr ω odpowiada za siłę efektu obserwacji odstającej, natomiast wyrażenie $A(B)/\{G(B)H(B)\}$ modeluje jego dynamikę. Jeśli chwila, w której wystąpiła interwencja oraz jej dynamika jest znana, wtedy mamy do czynienia z klasycznym modelem interwencji. Natomiast w kontekście wykrywania obserwacji odstających nie znamy, ani momentu wystąpienia, ani dynamiki anomalii. Model (2.11) pozwala na klasyfikację dynamiki obserwacji odstających na cztery podstawowe typy [11]:

- Addytywna obserwacja odstająca (AO),
- Zmiana poziomu (LS),
- Tymczasowa zmiana (TC),
- Innowacyjna obserwacja odstająca (IO).

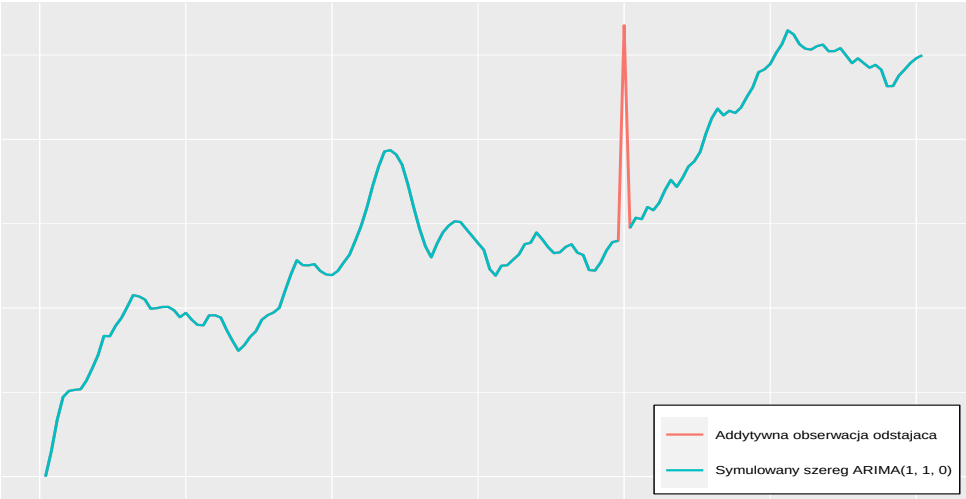
Definicja 2.1. Addytywną obserwacją odstającą (ang. *additive outlier*) nazwiemy obserwację, która wpływa na szereg czasowy jedynie w chwili $t = T$. Jej dynamikę możemy przedstawić następująco:

$$\frac{A(B)}{G(B)H(B)} = 1 \tag{2.12}$$

Dla przykładu, addytywną obserwacją odstającą może być błąd przy rejestrowaniu zapisu ze wskaźnika kontrolującego pracę maszyny (nazywamy również *gross error*). Jeśli obserwacja odstająca występuje w chwili $t = T$, to szereg czasowy Y_t^* (z addytywną anomalią) można zapisać jako:

$$Y_t^* = Y_t + \omega P_t^{(T)}. \tag{2.13}$$

W praktyce AO jest najczęściej występującym rodzajem obserwacji odstających, gdyż w danych rzeczywistych często mamy do czynienia z nagłym wzrostem lub spadkiem, po którym następuje powrót do pierwotnego położenia. Postać funkcji transferowej (2.6) odpowiada dynamice AO, a przykład takiej anomalii przedstawia rysunek 2.1.



Rysunek 2.1: Efekt generowany przez addytywną obserwację odstającą.

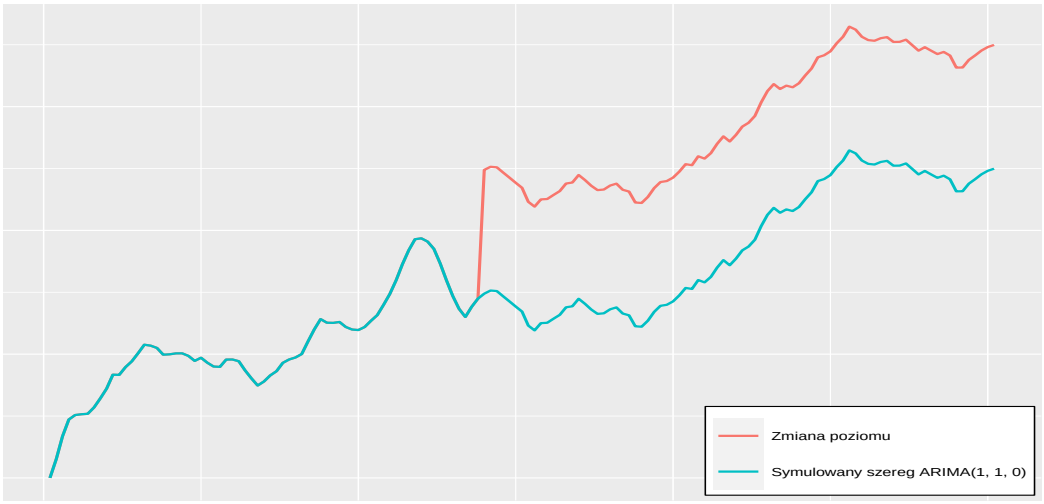
Definicja 2.2. Zmianą poziomu (ang. *level shift*) nazwiemy obserwację odstającą, która generuje nagłe oraz trwałe przesunięcie poziomu szeregu. Jej dynamikę możemy przedstawić jako:

$$\frac{A(B)}{G(B)H(B)} = \frac{1}{1-B}. \tag{2.14}$$

Szereg ze zmianą poziomu można zapisać jako:

$$Y_t^* = Y_t + \omega S_t^{(T)}. \tag{2.15}$$

Postać funkcji transferowej (2.4) odpowiada dynamice efektu generowanego przez LS, a przykład takiej obserwacji przedstawiono na rysunku 2.2.



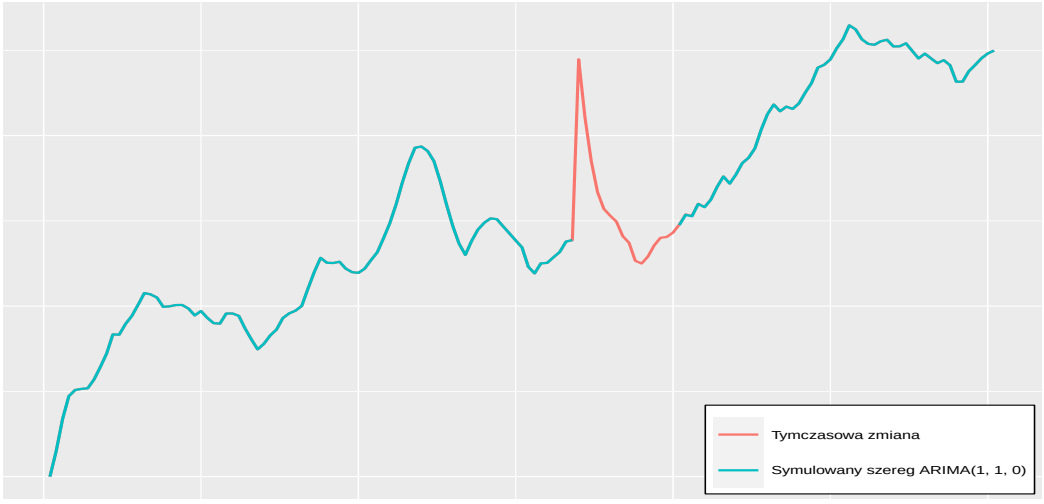
Rysunek 2.2: Efekt generowany przez zmianę poziomu.

Definicja 2.3. Tymczasową zmianą (ang. *temporary change*) w przebiegu szeregu czasowego nazwiemy obserwację odstającą, która generuje nagłe przesunięcie poziomu szeregu, które stopniowo wygasa. W tym przypadku, dynamika efektu obserwacji odstającej może być przedstawiona jako:

$$\frac{A(B)}{G(B)H(B)} = \frac{1}{1-\delta B}, \qquad \delta \in [0,1]. \tag{2.16}$$

Stała δ odpowiada za siłę wygaszania efektu TC. W zastosowaniach praktycznych sugerowaną wartością jest $\delta = 0.7$.

Sposób, w który modelujemy tymczasową zmianę odpowiada funkcji transferowej (2.7) oraz po uproszczeniu (2.8). Rysunek 2.3 przedstawia przykład obserwacji odstającej typu TC.



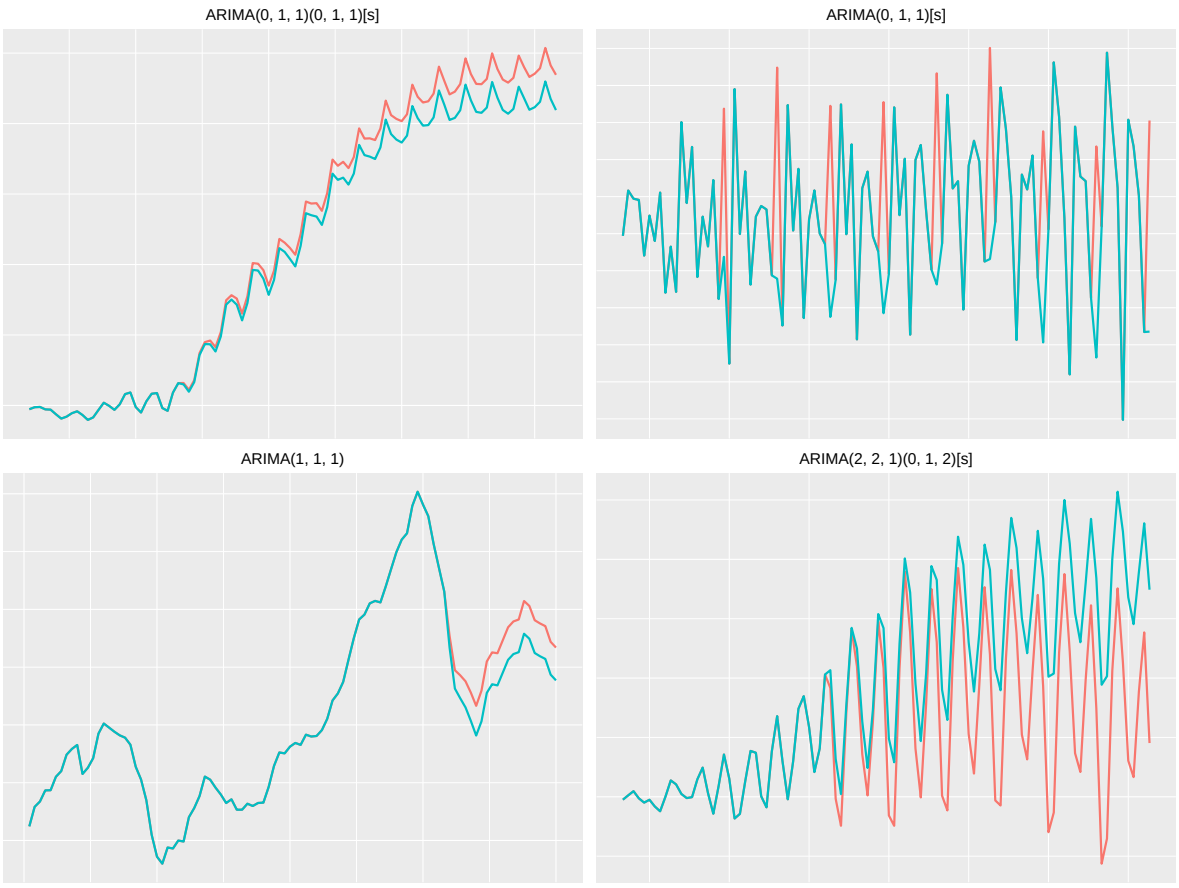
Rysunek 2.3: Efekt generowany przez tymczasową zmianę w przebiegu szeregu.

Definicja 2.4. Innowacyjną obserwację odstającą (ang. *innovative outlier*) nazwiemy obserwację, która ma nagły i trwały wpływ na postać szeregu, a sama jej dynamika zależy od modelu wybranego dla Y_t. W najbardziej ogólnym przypadku, dynamika innowacyjnej obserwacji odstającej może być przedstawiona jako:

$$\frac{A(B)}{G(B)H(B)} = \frac{\theta(B)\Theta(B^s)}{\nabla^d \nabla_s^D \phi(B)\Phi(B^s)}, \tag{2.17}$$

gdy Y_t modelujemy za pomocą pełnego sezonowego modelu ARIMA(p, d, q)(P, D, Q)[s].

Z uwagi na fakt, że IO jako jedyny z wyżej wymienionych typów obserwacji odstających zależy od postaci przyjętego modelu, generowane efekty mogą przyjmować różną formę (rysunek 2.4). Nagła zmiana, która stopniowo przechodzi w zmianę poziomu powstaje, gdy Y_t jest procesem ARIMA(1, 1, 1). Sezonową zmianę poziomu można uzyskać, gdy Y_t jest sezonowym szeregiem ARIMA(0, 1, 1)[s], natomiast gdy dla Y_t przyjmujemy model ARIMA(0, 1, 1)(0, 1, 1)[s] IO powoduje sezonowe zmiany trendu. Najbardziej skomplikowaną strukturę, która uwzględnia praktycznie całkowitą zmianę trajektorii szeregu spowodowała obecność IO dla modelu ARIMA(2, 2, 1)(0, 1, 2)[s].



Rysunek 2.4: Efekty generowane przez innowacyjne obserwacje odstające (czerwone krzywe), gdy dla Y_t (niebieskie krzywe) wybrane są różne modele (opisane w tytułach wykresów).

Modelowanie obserwacji odstających przy użyciu modelu IO może budzić pewne wątpliwości praktyczne z uwagi na fakt, że efekt IO dla sezonowych szeregów może nie być ograniczony i narzucać deterministyczną strukturę (co jest widoczne dla dwóch przykładów,

gdy dla szeregów dopasowane zostały pełne modele SARIMA); w tym momencie trudno rozróżnić wyjściowy szereg od poddanego wpływowi anomalii ze względu na skomplikowaną parametryzację efektu.

W pracy [25] zaproponowano rozwiązanie tego problemu, wprowadzając parametryzację sezonowych obserwacji odstających, które nie zależą od wybranego modelu. W szczególności autorzy pracy Kaiser i Maravall położyli nacisk na model dynamiki obserwacji odstającej, który określili jako sezonową zmianę poziomu (SLS).

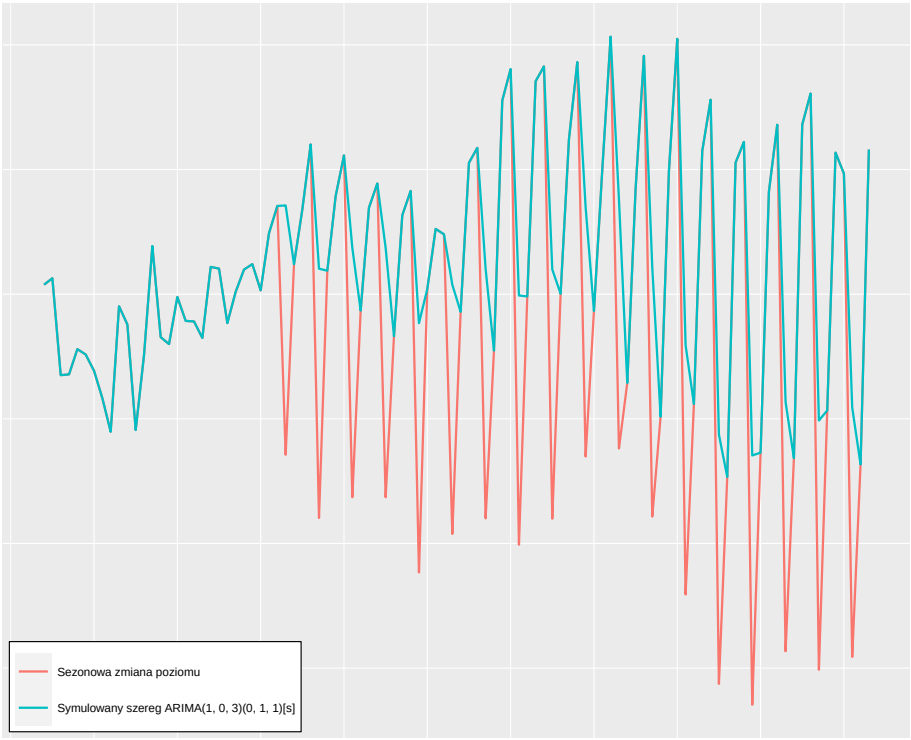
Definicja 2.5. Sezonową zmianą poziomu (ang. *seasonal level shift*) nazwiemy obserwację odstającą, która generuje addytywne zmiany o charakterze cyklicznym oraz zmianę poziomu trendu. Dynamika związana z sezonową zmianą poziomu może być przedstawiona jako:

$$\frac{A(B)}{G(B)H(B)} = \frac{1}{1-B^s}. \tag{2.18}$$

Sezonowe anomalie tego typu występują w szeregach finansowych; dodatkowo jako jedyne z wyżej wymienionych typów mają charakter deterministyczny, co pozwala na uwzględnienie ich na etapie prognozowania. Szereg z sezonową zmianą poziomu można zapisać jako:

$$Y_t^* = Y_t + \frac{\omega}{1-B^s} P_t^{(T)}. \tag{2.19}$$

Przykład obserwacji typu SLS przedstawiono na rysunku 2.5



Rysunek 2.5: Efekt generowany przez sezonową zmianę poziomu.

Wprawdzie dla modelu ARIMA(0, 1, 1)[s] obserwacja odstająca typu IO, również tworzy efekt sezonowej zmiany poziomu, natomiast, gdy chcemy użyć w procesie połączonej estymacji wybrany przez nas model, jednocześnie estymując efekty obserwacji odstających, uwzględniając typ SLS mamy dowolność w wyborze modelu w odróżnieniu od ograniczenia do konkretnego modelu, które nakłada wybór IO.

2.3 Metoda połączonej estymacji parametrów modelu ARIMA oraz efektów obserwacji odstających

Przedstawimy teraz szczegółowo procedurę Chena i Liu [11]. Jest to metoda połączonej estymacji współczynników modelu ARIMA oraz efektów obserwacji odstających różnych rodzajów.

2.3.1 Estymowanie efektów obserwacji odstających

By zbadać efekty obserwacji odstających występujące w estymowanych resztach (po dopasowaniu modelu SARIMA), definiujemy wielomian $\pi(B)$ jako:

$$\pi(B) = \frac{\tilde{\phi}(B)\tilde{\alpha}(B)}{\tilde{\theta}(B)} = 1 - \pi_1 B - \pi_2 B^2 - \dots, \tag{2.20}$$

gdzie $\tilde{\phi}(B) = \Phi(B^s)\phi(B)$, $\tilde{\alpha}(B) = \nabla^d \nabla_s^D$ oraz $\tilde{\theta}(B) = \Theta(B^s)\theta(B)$ (zgodnie z parametryzacją modelu SARIMA). Estymowane reszty $\hat{e}_t$, w których mogą być obecne efekty generowane przez obserwacje odstające, są postaci:

$$\hat{e}_t = \pi(B) Y_t^*. \tag{2.21}$$

Dla zdefiniowanych w podrozdziale 2.2 typów obserwacji odstających mamy odpowiednio:

$$\hat{e}_{IO,\,t} = \omega P_t^{(T)} + Z_t, \tag{2.22}$$

$$\hat{e}_{AO,\,t} = \omega \pi(B) P_t^{(T)} + Z_t, \tag{2.23}$$

$$\hat{e}_{LS,\,t} = \frac{\omega \pi(B)}{1 - B} P_t^{(T)} + Z_t, \tag{2.24}$$

$$\hat{e}_{TC,\,t} = \frac{\omega \pi(B)}{1 - \delta B} P_t^{(T)} + Z_t. \tag{2.25}$$

Estymatory najmniejszych kwadratów siły efektu generowanego przez obserwacje odstającą są wówczas postaci:

$$\hat{\omega}_{IO}(T) = \hat{e}_{IO,\,t}, \tag{2.26}$$

$$\hat{\omega}_{AO}(T) = -\frac{\sum_{t=T}^{n} \hat{e}_{AO,\,t}\, \pi_{T-t}}{\sum_{t=T}^{n} \pi_{T-t}^2}, \tag{2.27}$$

$$\hat{\omega}_{LS}(T) = \frac{\sum_{t=T}^{n} \hat{e}_{LS,\,t}\, (1 - \sum_{j=1}^{T-t} \pi_j)}{\sum_{t=T}^{n} (1 - \sum_{j=1}^{T-t})^2}, \tag{2.28}$$

$$\hat{\omega}_{TC}(T) = \frac{\sum_{t=T}^{n} \hat{e}_{TC,\,t}\, (\delta^{T-t} - \sum_{j=1}^{T-t-1} \delta^{T-t-j} \pi_j - \pi_{T-t})}{\sum_{t=T}^{n} (\delta^{T-t} - \sum_{j=1}^{T-t-1} \delta^{T-t-j} \pi_j - \pi_{T-t})^2}. \tag{2.29}$$

Zauważmy także, że dla ostatniej obserwacji szeregu ($T = n$) zachodzi równość:

$$\hat{\omega}_{IO}(n) = \hat{\omega}_{AO}(n) = \hat{\omega}_{LS}(n) = \hat{\omega}_{TC}(n) = \hat{e}_n. \tag{2.30}$$

Z uwagi na (2.30) można wywnioskować, że dla ostatniej obserwacji szeregu niemożliwe jest empiryczne odróżnienie typu obserwacji odstającej. Chang, Tiao i Chen [10] zaproponowali

metody wykrywania obserwacji odstających na podstawie standaryzowanych statystyk testowych postaci:

$$\hat{\tau}_{IO}(T) = \frac{\hat{\omega}_{IO}(T)}{\hat{\sigma}_Z}, \tag{2.31}$$

$$\hat{\tau}_{AO}(T) = \frac{\hat{\omega}_{IO}(T)}{\hat{\sigma}_Z}\sqrt{\sum_{t=T}^{n}\pi_{T-t}^2}, \tag{2.32}$$

$$\hat{\tau}_{LS}(T) = \frac{\hat{\omega}_{LS}(T)}{\hat{\sigma}_Z}\sqrt{\sum_{t=T}^{n}\left(1-\sum_{j=1}^{T-t}\right)^2}, \tag{2.33}$$

$$\hat{\tau}_{TC}(T) = \frac{\hat{\omega}_{TC}(T)}{\hat{\sigma}_Z}\sqrt{\sum_{t=T}^{n}\left(\delta^{T-t}-\sum_{j=1}^{T-t-1}\delta^{T-t-j}\pi_j-\pi_{T-t}\right)^2}. \tag{2.34}$$

Powyższe statystyki mają (w przybliżeniu) standardowy rozkład normalny dla ustalonego czasu wystąpienia obserwacji odstającej – T (przy założeniu normalności reszt).

By estymować wartości statystyk (2.31)-(2.34) należy najpierw wyznaczyć estymator $\hat{\sigma}_Z$. Użycie próbkowego odchylenia standardowego dla reszt, w których występują obserwacje odstające może oczywiście prowadzić do przeszacowania wartości $\hat{\sigma}_Z$. Z tego względu autorzy proponują zastosować odchylenie medianowe (MAD), tzn.:

$$\hat{\sigma}_Z = 1.483 \cdot med(|\hat{e}_t - med(\hat{e}_t)|). \tag{2.35}$$

Alternatywnie można wykorzystać metodę $\alpha\%$ ucinanego odchylenia standardowego, w której przed wyliczeniem próbkowego odchylenia standardowego usuwamy $\alpha\%$ największych (co do wartości bezwzględnej) obserwacji. Natomiast podejściem do estymacji, które nie wymaga sortowania wartości szeregu (oszczędność czasu obliczeń) jest metoda pomijająca obserwację w chwili T (chwila, dla której w danym momencie wyliczamy statystykę testową).

Jednokrotny test dla wszystkich czterech typów obserwacji odstających sprowadza się wówczas do sprawdzenia warunku::

$$\eta_t = \max_{t=1,\ldots,n}\{|\hat{\tau}_{IO}(t)|, |\hat{\tau}_{AO}(t)|, |\hat{\tau}_{LS}(t)|, |\hat{\tau}_{TC}(t)|\} > C, \tag{2.36}$$

gdzie C jest predefiniowaną stałą, którą określa się na podstawie długości szeregu oraz czułości detekcji obserwacji odstających. Gdy $\eta_T > C$, wtedy obserwację w chwili T można uznać za odstającą, a jej typ jest określany jako ten odpowiadający największej wartości bezwzględnej dla statystyk (2.31)-(2.34). Jednokrotny test ma swoje ograniczenia, w szczególności nie mamy pewności, czy estymatory $\hat{\omega}$ (oraz w konsekwencji $\hat{\tau}$) dla zadanej chwili T są nieobciążone, gdyż ich wartości mogą być zawyżone, bądź zaniżone ze względu na obecność sąsiednich obserwacji odstających. Między innymi dlatego w [10] i [11] została zaproponowana iteracyjna metoda wspólnej estymacji efektów oraz siły obserwacji odstających. W [25] rozszerzono tę procedurę o typ SLS (*Seaosnal Level Shift*).

2.3.2 Metoda estymacji parametrów modelu ARIMA w przypadku obecności wielu obserwacji odstających

Ogólny model dla szeregu Y_t^* poddanego wpływowi m obserwacji odstających pojawiających się w chwilach $T_1, \dots T_m$ jest postaci:

$$Y_t^* = \sum_{j=1}^{m} \omega_j L_j(B) P_t^{(T_j)} + \frac{\tilde{\theta}(B)}{\tilde{\phi}(B)\tilde{\alpha}(B)} Z_t, \tag{2.37}$$

gdzie $L_j(B)$ jest określone jak w (2.12), (2.18), (2.16) oraz (2.4) (w zależności od typu obserwacji odstającej). Nie czyniąc rozróżnienia na parametry estymowane i prawdziwe, reszty po dopasowaniu modelu Y_t^* możemy przedstawić jako:

$$\hat{e}_t = \sum_{j=1}^{m} \omega_j L_j(B) \pi(B) P_t^{(T_j)} + Z_t. \tag{2.38}$$

Jeśli znany jest czas wystąpienia oraz siła efektu obserwacji odstających, wtedy możemy na podstawie wzoru (2.37) skorygować te efekty i później przejść do estymacji parametrów modelu. Z drugiej strony, jeśli znane są parametry modelu, wtedy na podstawie wzoru (2.38) można zidentyfikować obserwacje odstające oraz estymować siłę ich efektów.

By wdrożyć powyżej zaproponowane idee Chen i Liu [11] zaproponowali iteracyjną procedurę, którą można podzielić na trzy etapy:

- Na początku przeprowadzana jest wstępna estymacja parametrów modelu i na jej podstawie wyznaczane są pozycje T_j oraz efekty $L_j(B)$ obserwacji odstających,
- Drugim etapem jest wspólna estymacja parametrów modelu oraz efektów generowanych przez obserwacje odstające, wykorzystująca wyniki uzyskane w poprzednim kroku,
- W trzecim etapie procedury obserwacje odstające oraz ich efekty zostają ponownie estymowane na podstawie uaktualnionych estymatorów parametrów, które są w mniejszym stopniu obciążone efektami anomalii.

Przedstawimy teraz bardziej szczegółowo kolejne kroki omawianej procedury.

1 *Wstępna estymacja parametrów modelu i identyfikacja obserwacji odstających*

1.1 Wyznaczenie estymatorów największej wiarogodności parametrów modelu na podstawie wyjściowego lub skorygowanego szeregu. Dla pierwszej iteracji używana jest wyjściowa postać szeregu, następnie w kolejnych iteracjach algorytmu używany jest skorygowany szereg.

1.2 Na podstawie reszt wyznaczonych jak w (1.1) wyznaczane są statystyki (2.31)-(2.34) dla każdej obserwacji szeregu i przeprowadzany jest jednokrotny test, zgodnie ze wzorem (2.36) dla chwili, w której maksymalną wartość osiąga $|\eta_T|$.

1.3 Jeśli nie została wykryta żadna obserwacja odstająca przechodzimy do kolejnego kroku. W przeciwnym przypadku należy usunąć efekt obserwacji z szeregu reszt oraz usunąć samą obserwację (uwzględniając jej typ) oraz powtórzyć krok 1.2, by sprawdzić czy w danych występują inne obserwacje odstające.

1.4 Po opuszczeniu pętli obejmującej kroki 1.1-1.3, jeśli w danych nie wykryto obserwacji odstających procedura się kończy – zaobserwowany szereg jest wolny od anomalii. Jeśli w pętli zostały zidentyfikowane jakiekolwiek obserwacje odstające dla parametrów z kroku 1.1, należy do niego powrócić i estymować parametry modelu na podstawie skorygowanego szeregu. Jeśli po ponownym przejściu przez 1.1-1.3 nie zostały wykryte nowe obserwacje odstające należy przejść do kolejnego kroku. W przeciwnym przypadku po raz kolejny powinniśmy estymować parametry dla nowo zidentyfikowanych anomalii. Procedurę należy powtarzać do momentu niepojawienia się nowych anomalii lub przekroczenia ustalonej liczby iteracji dla pętli wewnętrznej obejmującej etapy 1.1-1.4 (warto rozważyć zmniejszenie czułości detekcji poprzez zwiększenie wartości progowej C).

2 *Wspólna estymacja parametrów modelu oraz efektów obserwacji odstających*

2.1 Jeśli m obserwacji, zarejestrowanych w chwilach $T_1, \dots T_m$ zostało zidentyfikowane jako obserwacje odstające, możemy estymować łącznie ich efekty ω_j używając modelu regresji wielokrotnej (2.38), gdzie wyrażenie $L_j(B)P_t^{(T_j)}$ można traktować jako zmienną objaśniającą, natomiast $\hat{e}_t$ jako zmienną objaśnianą.

2.2 Następnie dla każdej z m obserwacji należy wyznaczyć statystykę $\hat{\tau}$. Sprawdzamy następnie czy $\min_j |\hat{\tau}_j| \leq C$, gdzie C jest tą samą ustaloną wartością progową, co w (1.2). Jeśli znajdziemy T_j, dla którego ten warunek zachodzi uznajemy taką anomalię za nieistotną i usuwamy ze zbioru obserwacji odstających.

2.3 Kolejnym krokiem jest wyznaczenie skorygowanego szeregu poprzez usunięcie efektów obserwacji odstających, używając najbardziej aktualnych estymatorów ω_j z etapu 2.1.

2.4 Kolejnym krokiem jest estymacja parametrów modelu metodą największej wiarogodności dla skorygowanego szeregu z 2.3. Jeśli zmiana standardowego błędu dla reszt między ostatnimi estymatorami jest większa niż ustalony poziom tolerancji ϵ (0.001 jest proponowaną przykładową wartością przez autorów), należy powtórzyć kroki 2.1-2.3, w przeciwnym przypadku można przejść do finalnego etapu metody.

3 *Identyfikacja obserwacji odstających bazująca na końcowych parametrach modelu*

3.1 Wyznaczamy reszty wykorzystując parametry modelu z kroku 2.4.

3.2 Wykorzystując reszty z 3.1 wykonujemy ponownie fazy 1-2 uwzględniając to, że parametry w fazie 1 są zastąpione tymi uzyskanymi w 2.4 oraz kroki 2.3-2.4 są pominięte. Estymatory $\hat{\omega}$ dla ostatniej iteracji po kroku 2.1 są końcowymi efektami zidentyfikowanych obserwacji odstających.

2.4 Studium przypadku: modelowanie ARIMA uwzględniające efekty obserwacji odstających

2.4.1 Opis schematu analizy

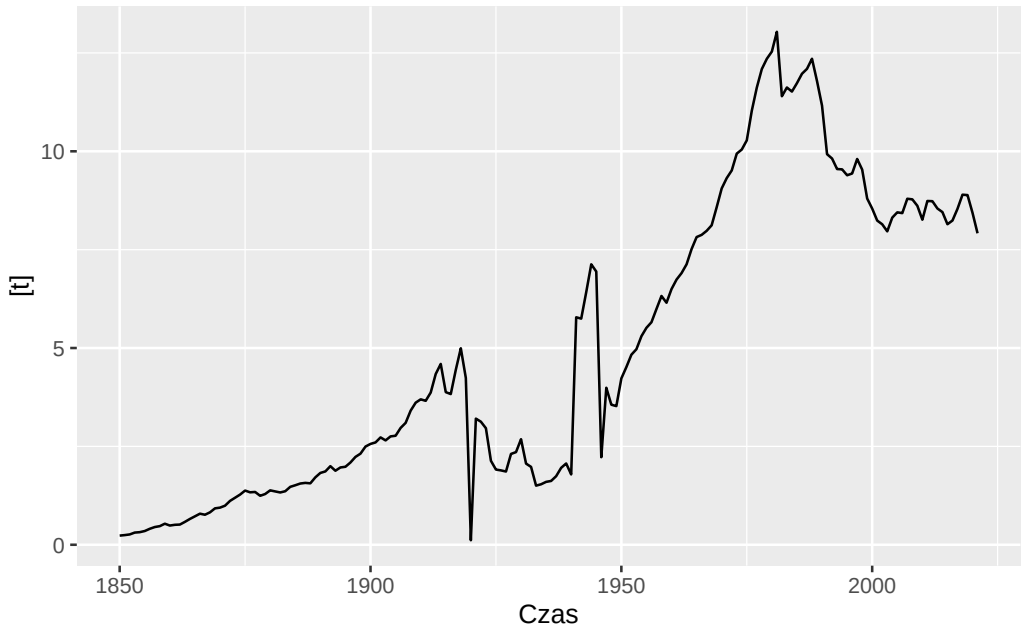
Poniższe studium przypadku (ang. *case study*) ma na celu zaprezentowanie istotności uwzględnienia efektów generowanych przez obserwacje odstające podczas modelowania

szeregów czasowych z wykorzystaniem modeli z rodziny ARIMA. Analiza będzie obejmować porównanie modeli ARIMA z częścią interwencyjną oraz klasycznych modeli z tej rodziny dla konkretnych danych rzeczywistych – wartości emisji CO2 per capita w Polsce na przestrzeni lat 1850-2020 (dane roczne) [20]. Dane te zostały wybrane, gdyż w pierwszej połowie XX wieku można zaobserwować ciekawe, pod kątem modelowania, nieregularności (można zakładać, że są one spowodowane pewnymi wydarzeniami historycznymi). W analizie uwzględnione zostaną następujące etapy:

- Opis danych, uwzględnienie transformacji potęgowej, podział na zbiór treningowy oraz testowy,
- Wstępna identyfikacja rzędów modeli AR(p) oraz MA(q) na podstawie funkcji ACF oraz PACF (dla danych zróżnicowanych),
- Identyfikacja modeli ARIMA(p,d,q) w oparciu o krokową procedurę minimalizującą wybrane kryterium informacyjne,
- Uzupełnienie zidentyfikowanych w poprzednich etapach modeli o modele ARIMA(p,d,q) + (AO, LS, TC, IO) uwzględniające efekty obserwacji odstających,
- Analiza reszt – weryfikacja poprawności dopasowania modeli,
- Analiza istotności współczynników modeli,
- Wyznaczenie oraz analiza dokładności prognoz dla modeli ARIMA,
- Porównanie prognoz dla modeli ARIMA z wybranymi metodami referencyjnymi.

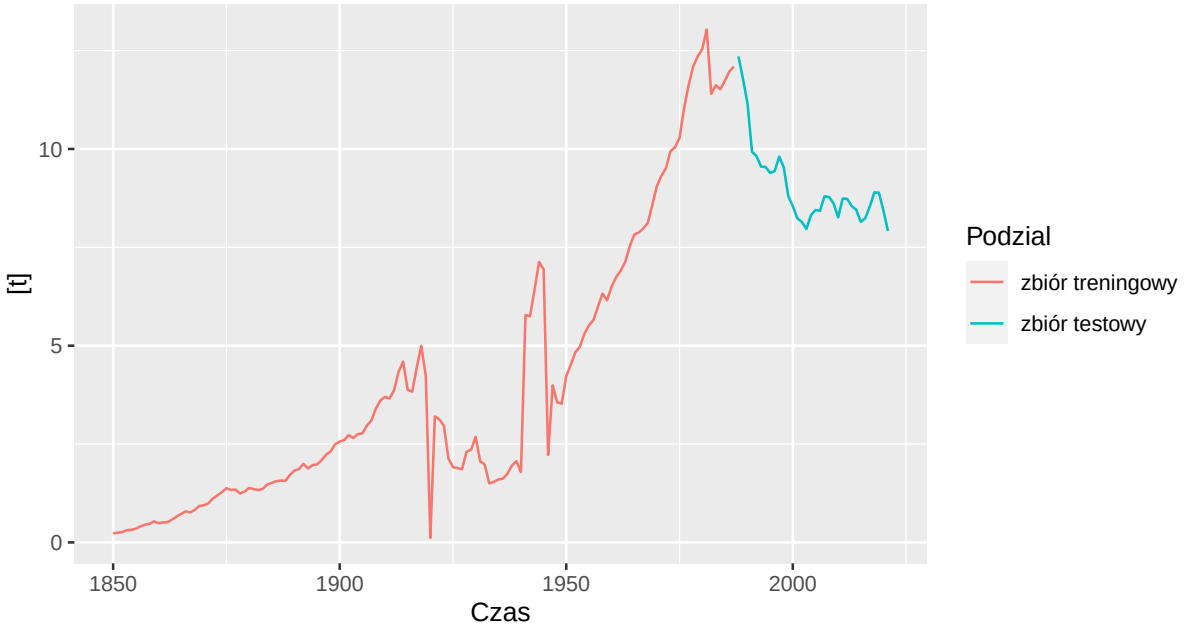
2.4.2 Opis danych

Przebieg analizowanego szeregu czasowego został zilustrowany poniżej (rysunek 2.6).



Rysunek 2.6: Emisja dwutlenku węgla ze spalania paliw kopalnych w celu wytworzenia energii w Polsce w okresie 1850-2020 wyrażona w tonach per capita.

W powyższym szeregu można zaobserwować wyraźny trend deterministyczny oraz zauważalną zmienność wariancji w wybranych interwałach czasowych, co implikuje jednoznacznie, że badany szereg jest szeregiem niestacjonarnym. Szczególne nieregularności występują w okresie I oraz II wojny światowej oraz 20-leciu międzywojennym. Przedział czasowy 1920-1950 wydaje się być kluczowym okresem jeżeli chodzi o detekcję obserwacji odstających typu AO, LS czy TC. W danych nie ma widocznych wahań sezonowych, zatem nie będzie konieczne uwzględnienie w procedurze estymacji sezonowych obserwacji odstających. W celu porównania oraz oceny skuteczności prognoz konstruowanych na podstawie dopasowanych modeli wyjściowe dane zostaną podzielone na zbiór treningowy oraz testowy (obejmujące odpowiednio 8/10 i 2/10 frakcji danych). Podział danych został przedstawiony na rysunku 2.7.



Rysunek 2.7: Podział danych na zbiór treningowy, który posłuży do dopasowania modeli z rodziny ARIMA oraz zbiór testowy, który posłuży do oceny dokładności prognoz.

2.4.3 Transformacja Boxa-Coxa

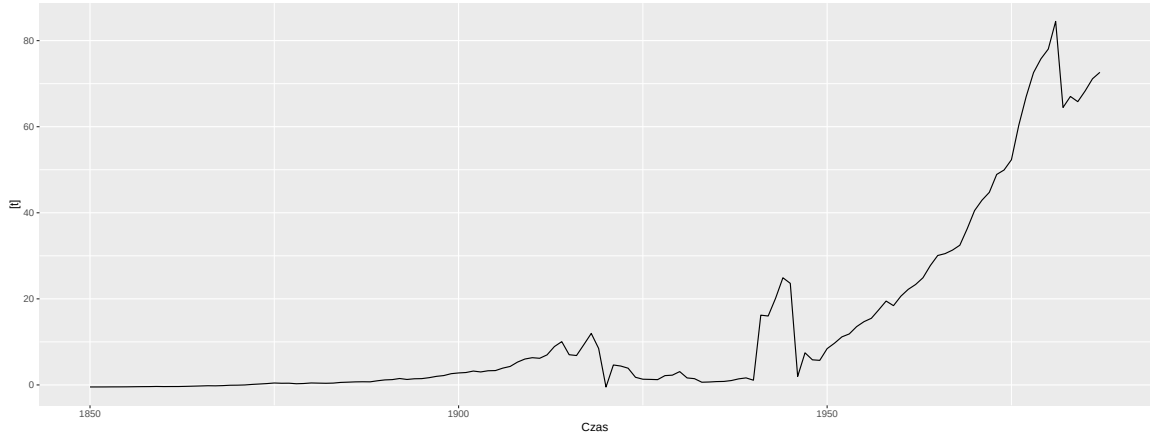
Kluczowe dla zredukowania efektów zewnętrznych oraz zaburzeń występujących w wyjściowych danych jest podejście uwzględniające transformację potęgową Boxa-Coxa. Dane po wykonanej transformacji będą mieć postać:

$$Y_t^{(\lambda)} = \begin{cases} \frac{Y_t^{\lambda}-1}{\lambda}, & \text{gdy } \lambda \neq 0 \\ \log(Y_t), & \text{gdy } \lambda = 0. \end{cases} \tag{2.39}$$

Do wyboru optymalnej wartości parametru λ wykorzystamy metodę [19], która jest popularnym wyborem automatyzującym ten etap analizy.

Dalsza analiza będzie przeprowadzana dwutorowo, by sprawdzić jaki wpływ na samą procedurę detekcji i estymacji oraz dokładność prognoz ma wcześniejsze użycie transformacji Boxa-Coxa. Na rysunku 2.8 przedstawiony został szereg po zastosowaniu transformacji

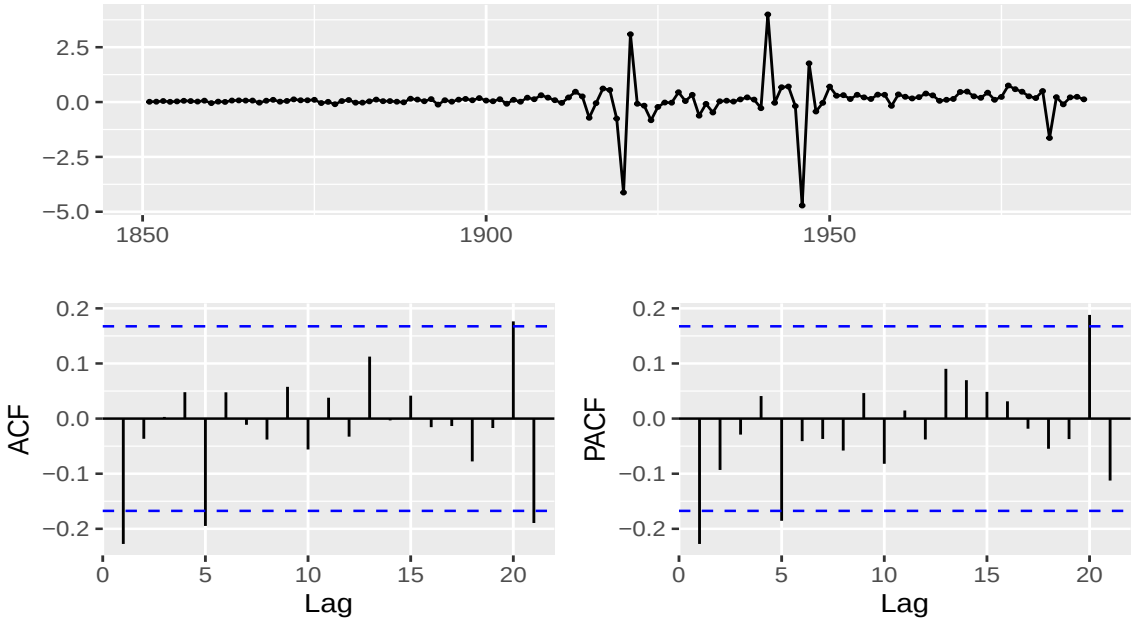
potęgowej z parametrem $\lambda = 2$. Przekształcony szereg czasowy (zbiór treningowy) charakteryzuje się o wiele mniejszymi wahaniami wariancji, co może istotnie zredukować liczbę zidentyfikowanych obserwacji odstających uwzględnionych na etapie modelowania.



Rysunek 2.8: Szereg czasowy (zbiór treningowy) po transformacji Boxa-Coxa.

2.4.4 Identyfikacja modeli w oparciu o funkcje PACF oraz ACF

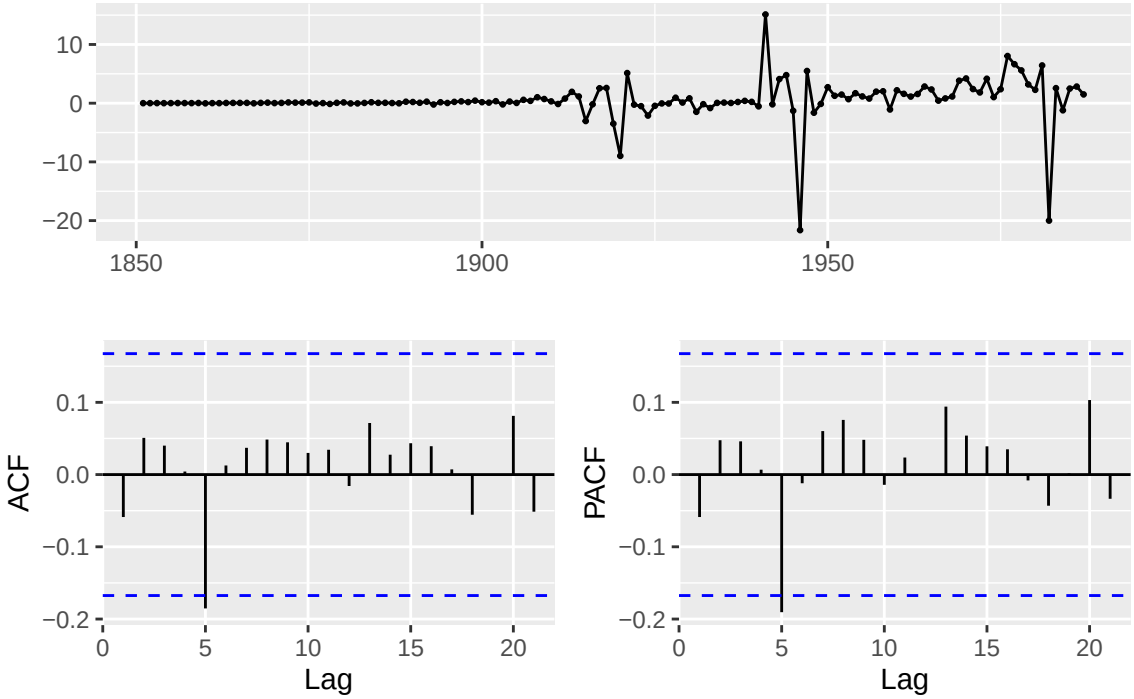
Analizowany szereg czasowy jest niestacjonarny, zatem, by dopasować prostsze modele stacjonarne konieczne będzie wykonanie operacji różnicowania (1.10). By następnie na podstawie funkcji autokorelacji oraz cząstkowej autokorelacji wskazać odpowiednie rzędy modeli stacjonarnych MA(q) oraz AR(p) (zgodnie z metodyką opisaną w podrozdziale 1.3.1). Szereg ∇Y_t (zbiór treningowy) wraz z korelogramami dla kolejnych opóźnień funkcji ACF oraz PACF jest zaprezentowany poniżej (rysunek 2.9).



Rysunek 2.9: Zróżnicowany szereg czasowy wraz z funkcjami ACF oraz PACF.

Na rysunku 2.9 można zaobserwować szybkie wygasanie funkcji ACF oraz PACF, co wskazuje na stacjonarność zróżnicowanego szeregu. Potwierdza to również p-wartość (mniejsza od 0.01) testu ADF, który odrzucił hipotezę H_0, o występowaniu w danych pierwiastka jednostkowego, względem alternatywy wskazującej na stacjonarność jednokrotnie zróżnicowanego szeregu. Zatem dla szeregu ∇Y_t możemy na podstawie funkcji ACF oraz PACF zidentyfikować wstępne rzędy modeli stacjonarnych MA(q) oraz AR(p). Będą to modele odpowiednio MA(1), MA(5) oraz AR(1) i AR(5). Za istotne można również uznać opóźnienia 20 oraz 21 dla ACF i 20 dla PACF, natomiast z uwagi na zbyt dużą złożoność (20 lub 21 potencjalnych parametrów) modelu, zostaną pominięte w dalszej analizie.

Rysunek 2.10 przedstawia wykres szeregu po zastosowaniu transformacji Boxa-Coxa oraz jednokrotnym zróżnicowaniu z opóźnieniem 1 wraz z korelogramami.



Rysunek 2.10: Zróżnicowany szereg czasowy (zbiór treningowy) po transformacji potęgowej wraz z funkcjami ACF oraz PACF.

Dla szeregu $\nabla Y_t^{(\lambda)}$ można zaobserwować silniejsze zanikanie funkcji ACF oraz PACF; jest to spowodowane użyciem transformacji Boxa-Coxa z parametrem $\lambda = 2$, która poprawiła stacjonarność badanego szeregu. Wartości funkcji ACF oraz PACF wskazują na wybór modeli stacjonarnych MA(5) oraz AR(5).

2.4.5 Automatyczna procedura wyboru optymalnego modelu ARIMA

Innym podejściem jest automatyczna procedura wyboru modelu opracowana przez Hyndmana i Khandakara [23]. Na początku, na podstawie odpowiednich testów statystycznych (domyślnie test KPSS lub wcześniej wspomniany test ADF – omówiony w podrozdziale 1.3.1) wybieramy rząd różnicowania (d), a następnie odpowiedni rząd wielomianu autoregresyjnego (p) oraz ruchomej średniej (q) dla modelu ARIMA(p,d,q) w oparciu o

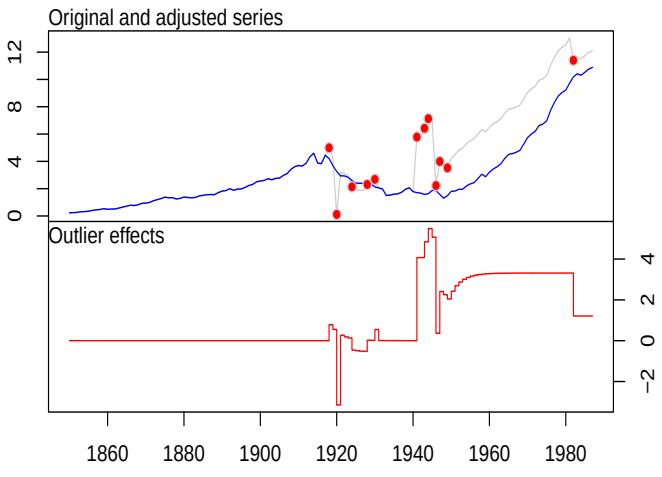
minimalizację wybranego kryterium informacyjnego (domyślnie AICc). W środowisku `R` wyżej opisana procedura krokowa zaimplementowana jest jako funkcja `auto.arima` w pakiecie `forecast`. We wspomnianej procedurze oczywiście znów warto uwzględnić wariant, w przypadku którego najpierw wykorzystana będzie odpowiednia transformacja potęgowa Boxa-Coxa.

Dla danych treningowych, na podstawie minimalizacji kryteriów AIC oraz AICc został dopasowany model ARIMA(0,1,1) z dryfem, natomiast kryterium BIC wskazało na na model ARIMA(0,1,1), który został już wcześniej wstępnie zidentyfikowany. Natomiast w przypadku danych uwzględniających transformację potęgową kryteria zgodnie wskazały na model ARIMA(0,2,1).

2.4.6 Automatyczna procedura wyboru optymalnego modelu ARIMA z uwzględnieniem efektów obserwacji odstających

Wykorzystaną w podrozdziale 2.4.5 automatyczną (krokową) procedurę wyboru modelu można uzupełnić o dodatkowe regresory w postaci efektów generowanych przez obserawcje odstające. W literaturze model tej postaci nosi nazwę (S)ARIMAX (w naszym przypadku brak części sezonowej) – gdzie X oznaczna *external regressors*, czyli zewnętrzne zmienne objaśniające. Do uwzględnienia efektów generowanych przez obserwacje odstające w modelowaniu ARIMA wykorzystana zostanie procedura zaproponowana przez Chena i Liu (patrz podrozdział 2.3.2), która w pakiecie `R` jest zaimplementowana jako funkcja `tso` w bibliotece `tsoutliers`.

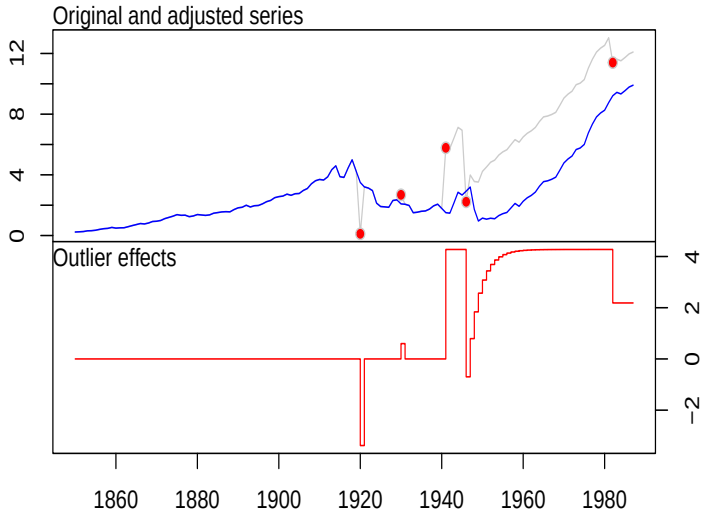
W konstruowanych modelach uwzględnione zostaną różne konfiguracje typów obserwacji odstających, biorąc również pod uwagę potencjalne modyfikacje poziomu progowego C, jeśli będzie to wymagane. Konstrukcja modeli zostanie przeprowadzana najpierw dla szeregu Y_t, następnie dla szeregu $Y_t^{(\lambda)}$. Na rysunku 2.11 przedstawiony jest wykres z wyznaczonymi w tabeli własnościami zidentyfikowanych obserwacji odstających (typ, moment wystąpienia, siła efektu - $\hat{\omega}$ oraz wartość statystyki testowej), dorysowany jest również skorygowany przebieg szeregu oraz łączny efekt wygenerowany przez zidentyfikowane obserwacje odstające.



typ	T	$\hat{\omega}$	$\hat{\tau}$
TC	1918	0.786	5.520
AO	1920	-3.535	-35.470
LS	1924	-0.556	-3.612
LS	1928	0.557	3.438
AO	1930	0.538	5.665
LS	1941	4.072	25.243
AO	1943	0.770	4.926
TC	1944	1.423	6.580
TC	1946	-4.408	-27.965
AO	1947	0.936	8.926
LS	1949	-0.758	-4.482
LS	1982	-2.104	-14.099

Rysunek 2.11: Zidentyfikowane obserwacje odstające (uwzględniając typy: AO, LS i TC) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Algorytm w sytuacji ukazanej na rysunku 2.11 rozpatruje trzy typy anomalii: AO, LS i TC. Wykrytych zostało, aż 12 obserwacji odstających, natomiast do danych skorygowanych został dopasowany model ARIMA(1,1,3) (z 12 dodatkowymi regresorami). Analizując dokładniej wykres, metoda wydaje się być zbyt czuła, co sugeruje zbyt niską wartość poziomu progowego (w tym wypadku wyliczanej ze wzoru $3.0025(n - 50)$ – wartość sugerowana przez twórców biblioteki). Poziom progowy wynosił podczas tej iteracji $C = 3.3025$. W następnym kroku przyjmiemy $C = 4$, by zmniejszyć czułość detekcji dla procedury łącznej estymacji efektów obserwacji odstających oraz parametrów modelu, co powinno znacznie zmniejszyć efekt nadmiernego dopasowania do danych treningowych. Efekty przeprowadzonej zmiany dobrze obrazuje ilustracja 2.12.



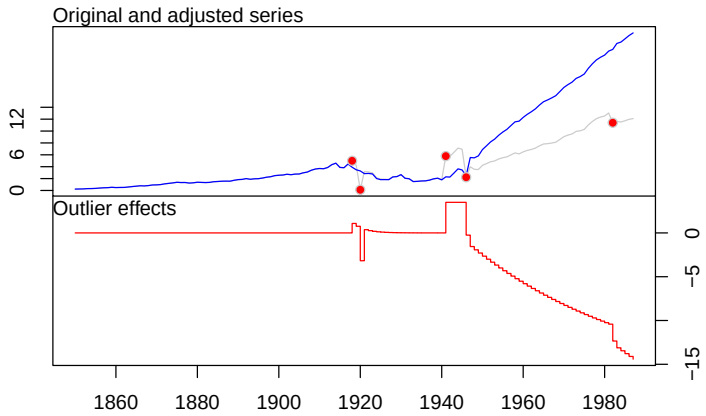
typ	T	$\hat{\omega}$	$\hat{\tau}$
AO	1920	-3.382	-28.763
AO	1930	0.599	5.074
LS	1941	4.274	20.707
TC	1946	-4.973	-26.989
LS	1982	-2.089	-9.845

Rysunek 2.12: Zidentyfikowane obserwacje odstające (uwzględniając typy: AO, LS i TC) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Dla zwiększonej wartości C procedura w dosyć ciekawy sposób zmodyfikowała wyjściowy szereg. Wykrytych zostało 5 obserwacji odstających: dwie addytywne, które analizując wykres nie budzą większych wątpliwości, natomiast zdecydowanie ciekawsza jest struktura kolejnych trzech anomalii w postaci dwóch zmian poziomu przeplecionych tymczasową zmianą. Taka sekwencja obserwacji odstających generuje dosyć nietypowy efekt, który skutkuje stałą zmianą poziomu. Dla skorygowanego szeregu dopasowany został model ARIMA(0,1,1) z dodatkowymi 5 zmiennymi modelującymi efekty interwencji.

Patrząc holistyczne na zidentyfikowane obserwacje odstające w kontekście specyfiki analizowanych danych – uwzględniając kontekst historyczny – można dojść do ciekawych konkluzji. Pierwszą chronologicznie AO można powiązać z końcem pierwszej wojny światowej oraz zmianami uwarunkowanymi przez odzyskanie przez Polskę niepodległości. O wiele mniejsza pod względem siły efektu obserwacja odstająca typu AO, która wystąpiła w 1930 roku może być związana z powojennym wzrostem gospodarczym. Kolejne dwie (chronologicznie) anomalie można odbierać jako echa początku i końca drugiej wojny światowej. Natomiast zmianę poziomu w 1982 jako następstwo wprowadzenia stanu wojennego.

Następny skonstruowanym modelem będzie model zawierający dodatkowo typ IO; rysunek 2.13 przedstawia charakterystyki przeprowadzonej procedury detekcji dla konfiguracji uwzględniającej dodatkowo IO.

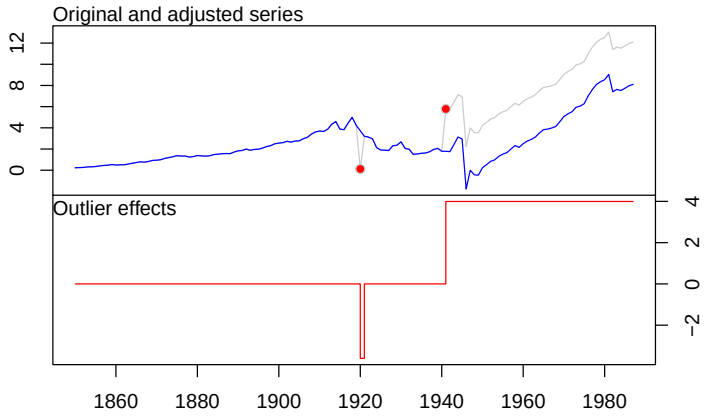


typ	T	$\hat{\omega}$	$\hat{\tau}$
TC	1918	1.081	3.801
AO	1920	-3.706	-14.148
LS	1941	3.491	11.299
IO	1946	-3.743	-12.535
IO	1982	-1.756	-6.266

Rysunek 2.13: Zidentyfikowane obserwacje odstające (uwzględniając typy: AO, IO, LS i TC) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Algorytm wykrył nietypową obserwację odstającą typu IO w roku 1946, jako dodany regresor do modelu ARIMA(1,1,2) (oraz dodatkowo cztery inne obserwacje odstające jako dodatkowe zmienne), która ma nieograniczony efekt wpływający na postać szeregu. Uwzględnienie w modelowaniu anomalii typu IO może rodzić pewne wątpliwości, gdyż skorygowany przebieg szeregu nie wygląda z intuicyjnego punktu widzenia na poprawny.

W kolejnym kroku uwzględnione będą jedynie anomalie typu AO oraz LS.

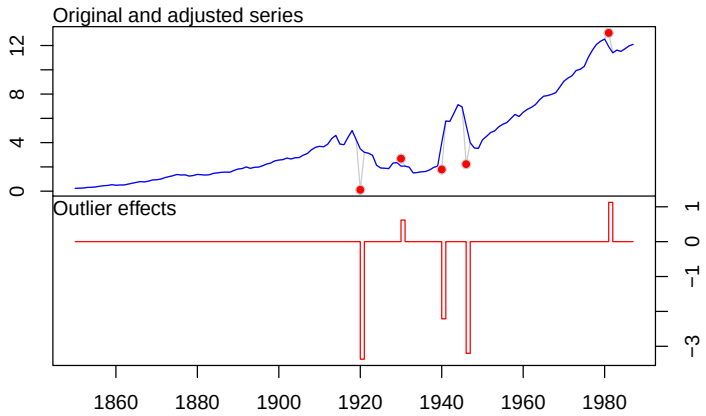


typ	T	$\hat{\omega}$	$\hat{\tau}$
AO	1920	-3.608	-9.643
LS	1941	3.994	7.549

Rysunek 2.14: Zidentyfikowane obserwacje odstające (uwzględniając typy: AO oraz LS) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Przy uwzględnieniu tylko typów AO oraz LS wykryta została jedna zmiana impulsowa oraz druga w postaci trwałej zmiany poziomu. W wyniku zastosowania tak skalibrowanego algorytmu dopasowany został model ARIMA(0,1,0) z dwoma zmiennymi fikcyjnymi (ang. *dummy variables*) w postaci wymienionych w tabeli na rysunku 2.14 efektów generowanych przez obserwacje odstające. Z punktu widzenia specyfiki danych mogą one odpowiadać za wpływ końca pierwszej oraz początku drugiej wojny światowej na emisję dwutlenku węgla.

Najbardziej naturalną w odbiorze wizualnym korekcję, bazując na rysunku 2.15, udało się uzyskać poprzez uwzględnienie w procesie estymacji oraz detekcji tylko addytywnych obserwacji odstających.

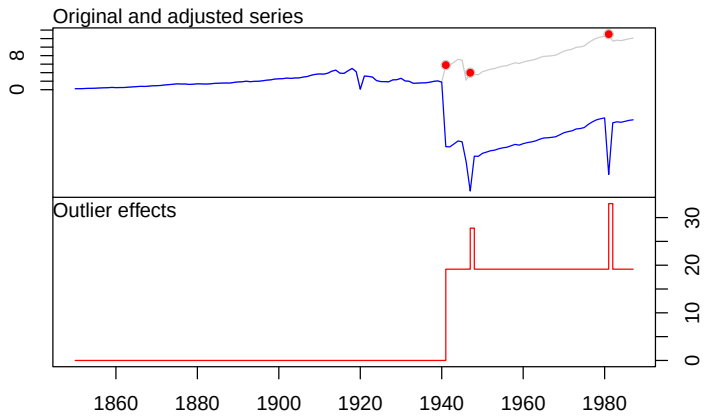


typ	T	$\hat{\omega}$	$\hat{\tau}$
AO	1920	-3.370	-24.053
AO	1930	0.619	4.387
AO	1940	-2.213	-16.012
AO	1946	-3.202	-23.026
AO	1981	1.128	8.171

Rysunek 2.15: Zidentyfikowane obserwacje odstające uwzględniając typ AO wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Zostało wykrytych pięć addytywnych anomalii impulsowych, które zostały uwzględnione w finalnie dopasowanym modelu ARIMA(0,1,1). Podobnie, jak w przypadku interpretacji poprzednich wyników, zidentyfikowane zmiany o charakterze impulsowym mogły być indukowane wspomnianymi wcześniej wydarzeniami historycznymi.

Kolejnym krokiem analizy jest uwzględnienie we wspólnej procedurze estymacji parametrów oraz efektów obserwacji odstających transformacji Boxa-Coxa. Analizując szereg $Y_t^{(\lambda)}$ będziemy konstruować modele uwzględniające te same typy obserwacji odstających, dla tych samych konfiguracji jak to miało miejsce w przypadku szeregu Y_t. Na rysunku 2.16 podczas procedury detekcji zostały użyte standardowe typy anomalii: AO, LS oraz TS.

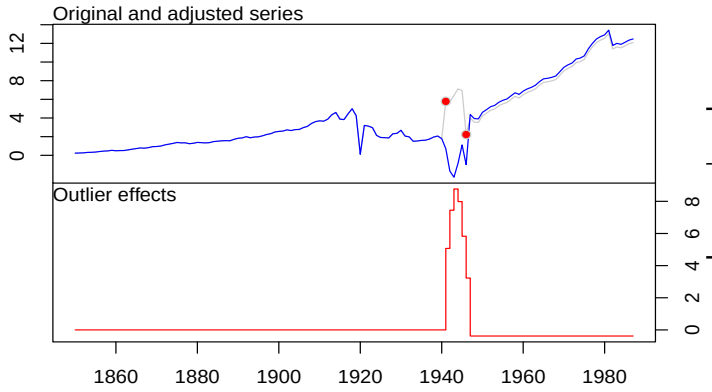


typ	T	$\hat{\omega}$	$\hat{\tau}$
LS	1941	19.166	7.606
AO	1947	8.622	6.760
AO	1981	13.774	11.361

Rysunek 2.16: Zidentyfikowane obserwacje odstające (uwzględniając typy: AO, LS i TC oraz transformację potęgową Boxa-Coxa) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Wynik analizy przedstawione na rysunku 2.16 odrobinę różnią się od tych uzyskanych dla szeregu Y_t. Żadna anomalia typu TC nie została zidentyfikowana. Wartości estymatorów $\hat{\omega}$ dla anomalii w roku 1941 oraz 1981 są o wiele większe niż jakiekolwiek wartości estymatorów siły efektu dla poprzednich detekcji przeprowadzonych w oparciu o szereg

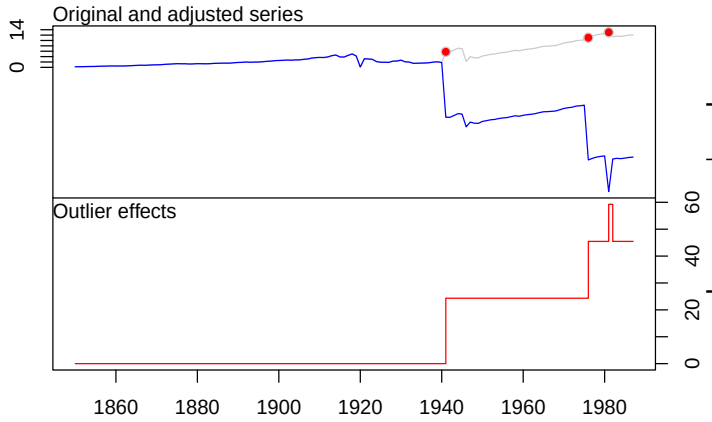
Y_t. Do poprawionego przebiegu szeregu dopasowany został model ARIMAX(1,1,1) z trzema regresorami skonstruowanymi na podstawie dwóch zidentyfikowanych anomalii impulsowych oraz zmiany poziomu. W kolejnym kroku przyjrzymy się jaki wpływ na wyniki będzie miało uwzględnienie IO jako dodatkowego typu anomalii dla szeregu $Y_t^{(\lambda)}$.



typ	T	$\hat{\omega}$	$\hat{\tau}$
IO	1941	5.07	0.965
LS	1946	-0.38	-0.090

Rysunek 2.17: Zidentyfikowane obserwacje odstające (uwzględniając typy: AO, IO, LS i TC oraz transformację potęgową Boxa-Coxa) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

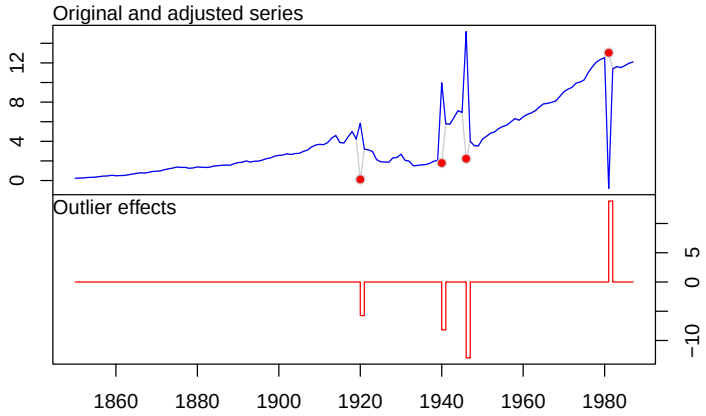
Uwzględnienie typu IO poskutkowało zastąpieniem obserwacji odstających typu AO innowacją w roku 1941. Co ciekawe, obydwie zidentyfikowane anomalie przypadają na okres II wojny światowej. Do danych wyznaczonych po zastosowaniu procedury detekcji anomalii, dopasowany został model ARIMA(0,0,5), inaczej MA(5) uwzględniający dodatkowo dwie anomalie. Jest to dość zaskakujący wynik, zważywszy iż na rysunku 2.17 skorygowany szereg w dalszym ciągu wydaje się posiadać wyraźny trend deterministyczny, przy czym dobrany rząd różnicowania (d) dla modelu wyniósł 0 – co implikuje, że szereg powinien być stacjonarny (wątpliwości, co do poprawności dopasowanego modelu będą w kolejnym podrozdziale dokładnie zweryfikowane). W kolejnym kroku przyjrzymy się wynikom detekcji oraz estymacji przy uwzględnieniu typów AO oraz LS dla szeregu $Y_t^{(\lambda)}$.



typ	T	$\hat{\omega}$	$\hat{\tau}$
LS	1941	24.335	9.165
LS	1976	21.126	5.338
AO	1981	13.779	5.106

Rysunek 2.18: Zidentyfikowane obserwacje odstające (uwzględniając typy: AO i LS oraz transformację potęgową Boxa-Coxa) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Analogicznie jak to miało miejsce w przypadku detekcji oraz modelowania ukazanego na rysunku 2.16, korekta bazowego szeregu wydaje się być w dużym stopniu obciążona dużymi wartościami estymatorów $\hat{\omega}$ siły efektów obserwacji odstających, w wyniku czego kompensacja oryginalnych efektów, które znajdują się w szeregu jest zdecydowanie zawyżona. Jeśli chodzi o dobór modelu, dopasowany został model MA(4) z trzema dodatkowymi regresorami, co rodzi podobne wątpliwości, jeśli chodzi o poprawność dopasowania modelu (tak jak to miało miejsce w przypadku wyników przedstawionych na rys. 2.17). Ostatnią konfiguracją badanych typów parametrów będzie detekcja AO dla szeregu $Y_t^{(\lambda)}$.



typ	T	$\hat{\omega}$	$\hat{\tau}$
AO	1920	-5.748	-7.144
AO	1940	-8.190	-10.453
AO	1946	-12.969	-16.349
AO	1981	13.846	17.647

Rysunek 2.19: Zidentyfikowane obserwacje odstające (uwzględniając typ AO oraz transformację potęgową Boxa-Coxa) wraz ze skorygowanym przebiegiem szeregu oraz łącznym efektem interwencyjnym.

Na rysunku widać znów podobne charakterystyki, które były widoczne w procesach estymacji efektów dla szeregu $Y_t^{(\lambda)}$ – wysokie wartości (co do wartości bezwzględnej) estymatorów siły efektów obserwacji odstających. Cztery zidentyfikowane impulsy scharakteryzowane na rysunku 2.19 zostały uwzględnione przy konstrukcji modelu ARIMA(0,1,1) dla odpowiednio przekształconych danych.

Patrząc holistycznie na zidentyfikowane obserwacje, w przypadku przekształconego szeregu (po zastosowaniu transformacji Boxa-Coxa) ponownie można zauważyć możliwe powiązania z wcześniej wspomnianymi wydarzeniami historycznymi dla zidentyfikowanych anomalii. Otrzymane skorygowane szeregi dla $Y_t^{(\lambda)}$ charakteryzowały się mniej naturalnym przebiegiem porównując do korekcji dla szeregu Y_t.

2.4.7 Przegląd zidentyfikowanych modeli

W tabelach 2.1 oraz 2.2 zestawiono wstępnie zidentyfikowane modele ARIMA wraz z wartościami kryteriów informacyjnych oraz wartościami logarytmu funkcji wiarogodności. Niestety do analizy porównawczej dobroci dopasowania na podstawie kryteriów informacyjnych można wykorzystać algorytmy, dla których dane były przekształcone w jednakowy sposób, więc różny rząd różnicowania, transformacja Boxa-Coxa oraz korekcja ze względu na efekty obserwacji odstających znacznie utrudniają porównywanie różnych klas dopasowanych modeli.

By ograniczyć liczbę rozważanych modeli zrezygnujemy z modeli o liczbie parametrów (k) większej bądź równej 5 jeśli chodzi o modele wybrane na podstawie ACF i PACF

dla szeregu Y_t oraz ograniczymy się do modelu ARIMA(0,1,5) dla szeregu Y_t (wybory wiąża się z niższymi wartościami kryteriów informacyjnych względem konkurujących modeli). W tym momencie należałoby również ograniczyć liczbę modeli ARIMA z częścią interwencyjną (choćby ze względu na liczbę parametrów), natomiast z uwagi na główny cel przeprowadzanej analizy wszystkie modele z częścią interwencyjną zostaną pozostawione i uwzględnione na etapie analizy reszt.

Tabela 2.1: Wstępnie zidentyfikowane modele ARIMA

	k	AIC	AICc	BIC	$\log(\mathcal{L})$
Bez części interwencyjnej					
ARIMA(1, 1, 0)	1	313.42	313.51	319.26	-154.71
ARIMA(5, 1, 0)	5	316.73	317.38	334.25	-152.37
ARIMA(0, 1, 1)	1	312.82	312.91	318.66	-154.41
ARIMA(0, 1, 5)	5	316.97	317.61	334.49	-152.48
Z uwzględnieniem części interwencyjnej					
ARIMA(0, 1, 1) + dryf	2	311.56	311.74	320.32	-152.78
ARIMA(1, 1, 3) + (AO + LS +TC)	16	-64.83	-59.69	-15.19	49.42
ARIMA(0, 1, 1) + (AO + LS + TC)	6	32.24	33.11	52.68	-9.12
ARIMA(1, 1, 2) + (AO + LS +TC + IO)	8	115.16	116.58	141.44	-48.58
ARIMA(0, 1, 0) + (AO + LS)	2	220.37	220.55	229.13	-107.18
ARIMA(0, 1, 1) + AO	6	95.95	96.82	116.39	-40.98

Tabela 2.2: Wstępnie zidentyfikowane modele ARIMA z transformacją Boxa-Coxa

	k	AIC	AICc	BIC	$\log(\mathcal{L})$
Bez części interwencyjnej					
ARIMA(5, 1, 0)	5	737.97	738.61	755.49	-362.98
ARIMA(0, 1, 5)	5	736.84	737.48	754.36	-362.42
ARIMA(0, 2, 1)	1	731.47	731.56	737.30	-363.74
Z uwzględnieniem części interwencyjnej					
ARIMA(1, 1, 1) + (AO + LS +TC)	5	666.15	666.80	683.67	-327.08
ARIMA(0, 0, 5) + (AO + LS +TC + IO)	8	851.16	852.57	877.51	-416.58
ARIMA(0, 0, 4) + (AO + LS)	8	778.48	779.89	804.83	-380.24
ARIMA(0, 1, 1) + AO	6	572.55	573.42	592.99	-279.27

2.4.8 Analiza reszt

Formalna ocena poprawności dopasowania modeli została przeprowadzona wykorzystując testy losowości oraz normalności reszt (podrozdział 1.4). W przypadku testów Ljungi-Boxa (L-B), Boxa-Pierce'a (B-P) oraz McLeod-Li (M-L) uwzględniona została korekta stopni swobody (patrz podrozdział 1.4). Maksymalne opóźnienie użyte w wyżej wymienionych testach wyniosło $h = 20$, natomiast w przypadku gdy $h - k < 10$, przyjęto $h - k = 10$, by uwzględnić dostateczną liczbę opóźnień do wyliczenia statystyk testowych. Pozostałe wykorzystane testy diagnostyczne zostały szczegółowo opisane w podrozdziale 1.4. Tabele

2.3 i 2.4 przedstawiają uzyskane wyniki, tzn. p-wartości dla wszystkich rozważanych testów diagnostycznych.

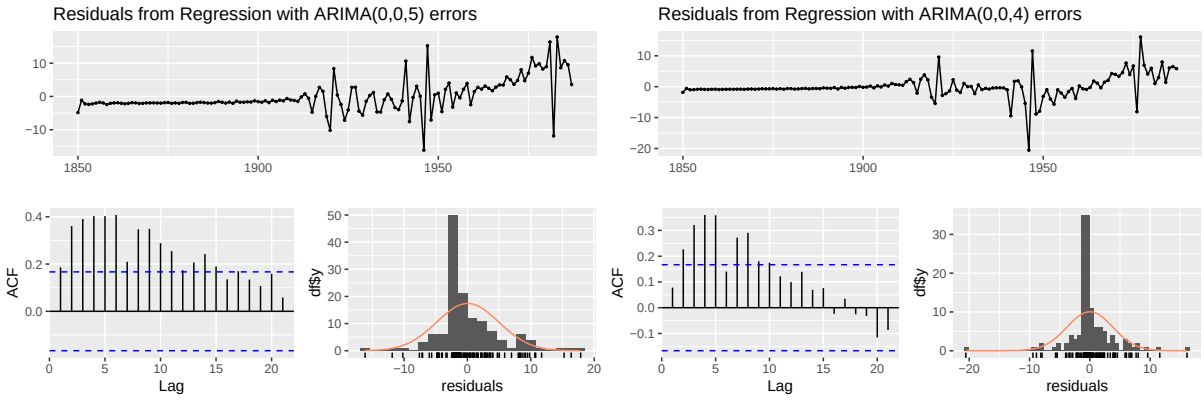
Tabela 2.3: Wartości p dla testów losowości oraz normalności.

	B-P	L-B	M-L	TP	R	DS	J-B
ARIMA(0, 1, 1)	0.845	0.768	0.661	0.119	0.000	0.659	0
ARIMA(0, 1, 1) + dryf	0.809	0.724	0.725	0.175	0.000	0.659	0
ARIMA(1, 1, 3) + (AO + LS +TC)	0.196	0.130	0.057	0.278	0.171	0.186	0
ARIMA(0, 1, 1) + (AO + LS + TC)	0.442	0.344	0.864	0.058	0.002	0.463	0
ARIMA(1, 1, 2) + (AO + LS +TC + IO)	0.658	0.594	0.136	0.049	0.325	0.304	0
ARIMA(0, 1, 0) + (AO + LS)	0.990	0.984	1.000	0.946	0.000	0.304	0
ARIMA(0, 1, 1) + AO	0.233	0.163	0.030	0.007	0.023	0.883	0

Tabela 2.4: Wartości p dla testów losowości oraz normalności z transforamcją Boxa-Coxa.

	B-P	L-B	M-L	TP	R	DS	J-B
ARIMA(0, 2, 1)	0.947	0.925	0.988	0.946	0	0.463	0
ARIMA(0, 1, 5)	0.995	0.991	0.561	0.735	0	0.463	0
ARIMA(1, 1, 1) + (AO + LS +TC)	0.878	0.835	1.000	0.498	0	0.883	0
ARIMA(0, 0, 5) + (AO + LS +TC + IO)	0.000	0.000	0.000	0.049	0	0.463	0
ARIMA(0, 0, 4) + (AO + LS)	0.000	0.000	0.001	0.456	0	0.106	0
ARIMA(0, 1, 1) + AO	0.445	0.351	0.049	0.000	0	0.659	0

Analizując tabele 2.3 oraz 2.4, większość modeli wydaje się być poprawnie dopasowana z wyłączeniem modeli ARIMA(0,0,5) + (AO + LS +TC + IO) oraz ARIMA(0,0,4) + (AO + LS), dla których testy L-B oraz B-P odrzuciły hipotezę zerową o losowości reszt (przyjmując poziom istotności $\alpha = 0.05$), dlatego modele te wymagają dokładniejszej analizy za pomocą narzędzi graficznych (rysunek 2.20).

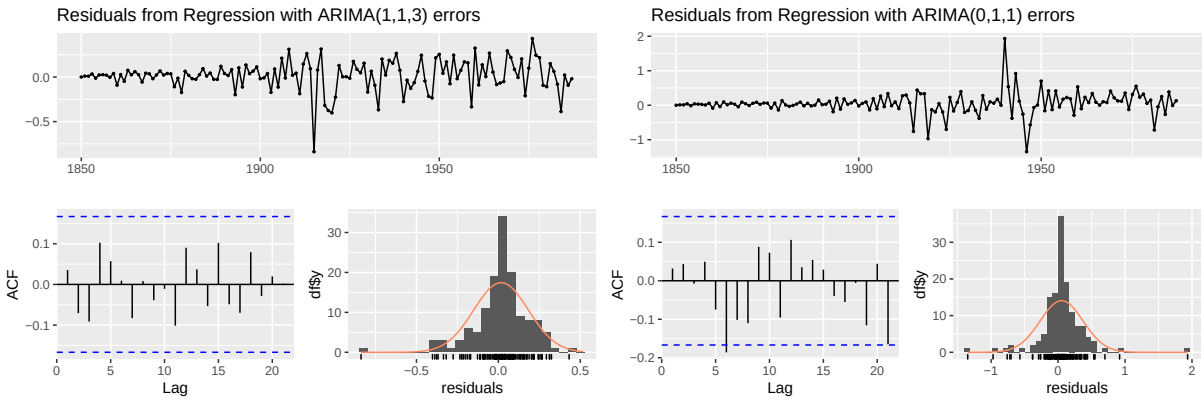


Rysunek 2.20: Analiza reszt dla modeli ARIMA(0,0,5) + (AO + LS +TC + IO) oraz ARIMA(0,0,4) + (AO + LS).

Wykresy reszt dla modelu ARIMA(0,0,5) + (AO + LS +TC + IO) oraz ARIMA(0,0,4) + (AO + LS) wskazują na ich istotną autokorelację a nawet możliwą obecność trendu

deterministycznego (patrz podrozdział 1.2), co automatycznie dyskwalifikuje je z dalszej analizy.

Z uwagi na dużą liczbę parametrów w modelu ARIMA(1,1,3) + (AO + LS + TC) oraz potencjalne efekty heteroskedastyczne występujące dla reszt z modelu ARIMA(0,1,1) + AO (dla szeregu Y_t), przyjrzymy się bliżej tym rezyduom, znów stosując odpowiednie narzędzia graficzne (rysunek 2.21).



Rysunek 2.21: Analiza reszt dla modeli ARIMA(1,1,3) + (AO + LS +TC) oraz ARIMA(0,1,1) + AO.

Dla modelu ARIMA(1,1,3) + (AO + LS +TC) z bardzo dużą liczbą parametrów ($k = 16$) można zaobserwować szybkie zanikanie funkcji ACF, co potwierdza hipotezę o białoszumowości reszt modelu. Na wykresie reszt widoczna jest niejednorodność ich wariancji, co wskazuje na występowanie efektów heteroskedastycznych w danych, które mogłyby być uwzględnione przy pomocy modeli z rodziny GARCH. Rozkład rezyduów jest rozkładem ciężkoogonowym – zjawisko, które również może być modelowane poprzez odpowiedni model warunkowo heteroskedastyczny.

Wykres funkcji ACF dla reszt modelu ARIMA(0,1,1) + AO (bez transformacji Boxa-Coxa) (rys. 2.21) wskazuje na istotną korelację dla szóstego oraz dwudziestego pierwszego opóźnienia, co budzi wątpliwości odnośnie białoszumowości reszt. W danych można również zaobserwować efekty heteroskedastyczne widoczne na wykresie rezyduów.

Wracając do pozostałych wniosków, test M-L wskazał na korelację kwadratów reszt dla modeli ARIMA(0,1,1) + AO (jeden z nich uwzględnia transformację potęgową Boxa-Coxa). Test TP odrzucił hipotezę o losowości reszt dla modeli ARIMA(0,1,1) + AO oraz modeli uwzględniających zmienne modelujące efekty IO. Co ciekawe test R, z wyłączeniem dwóch modeli, odrzucił hipotezę zerową o losowości, natomiast test DS dla każdego modelu ją przyjął.

Test J-B dla każdego modelu odrzucił hipotezę o normalności reszt, zatem konstrukcja przedziałów ufności, testy istotności współczynników czy test używany w procedurze opisanej przez Chena i Liu będą charakteryzowały się mniej dokładną (dokładną w sensie asymptotycznym) precyzją. Atutem metody Chena i Liu (gdy reszty mają rozkład normalny) są średnio dużo mniejsze wartości statystyki testowej N_{J-B}, czego niestety w naszej analizie nie udało się pokazać (autorzy pakietu `tsoutliers` [15] w dokumentacji przeprowadzili obszerną analizę odnośnie poprawy normalności reszt w dopasowanych modelach, gdy w analizie zostały uwzględnione efekty obserwacji odstających).

2.4.9 Testowanie istotności współczynników

Kolejnym etapem analizy poprawności dopasowania modeli ARIMA jest testowanie istotności współczynników. Statystyka testowa Z jest postaci:

$$Z = \frac{k_i}{se_i}, \tag{2.40}$$

gdzie k_i to i-ty współczynnik modelu (z części ruchomej średniej, autoregresyjnej, bądź interwencyjnej), natomiast se_i to jego błąd standardowy. Przy założeniu, że reszty dla danego modelu mają rozkład normalny, wtedy Z ma (asymptotyczny) standardowy rozkład normalny. W naszym przypadku test Jarque-Bera odrzucił hipotezę o normalności reszt (dla każdego testowanego modelu), zatem uzyskane p-wartości będą charakteryzowały się ograniczoną dokładnością.

Korzystając z powyższego testu, dla pozostałych w analizie modeli jedynie współczynnik dryfu z modelu ARIMA(0,1,1) + dryf uznany został za nieistotny (przyjmujemy $H_0 : k_i = 0$ na poziomie istotności 0.05). Reszta modeli, szczególnie współczynniki interwencyjne uzyskała p-wartości bardzo bliskie zeru. Dodatkowo, dla (uprzednio wykluczonego z analizy) modelu ARIMA(0,0,5) + (AO + LS +TC + IO) test wskazał za nieistotny współczynnik dla IO. Uwzględniając wyniki testów istotności, w dalszej analizie model ARIMA(0,1,1) z dryfem nie będzie brany pod uwagę tzn. ograniczymy się wyłącznie do modelu ARIMA(0,1,1).

2.4.10 Analiza dokładności prognoz

Najważniejszym, z praktycznego punktu widzenia, etapem analizy jest porównanie dokładności prognoz (oraz dopasowania dla zbioru treningowego) przy użyciu kryteriów dokładności predykcji. W poniższym porównaniu wykorzystamy pierwiastkek błędu średniokwadratowego ($RMSE$) zdefiniowany jako:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\widehat{Y}_t - Y_t)^2}{n}}, \tag{2.41}$$

gdzie $\hat{Y}_t$ to prognozowana (dopasowana) wartość odpowiadająca Y_t. Dodatkowym kryterium będzie średni bezwzględny błąd procentowy:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\widehat{Y}_t - Y_t}{Y_t}\right|. \tag{2.42}$$

Istotą porówniania 34-krokowych (liczba obserwacji zbioru testowego) prognoz będzie ocenienie poprawy ich dokładności dla modeli z częścią interwencyjną względem standardowych modeli ARIMA oraz metody naiwnej i optymalnego modelu ETS, dobranego na podstawie kryterium AICc (również uwzględniając transformację Boxa-Coxa dla optymalnej wartości parametru λ).

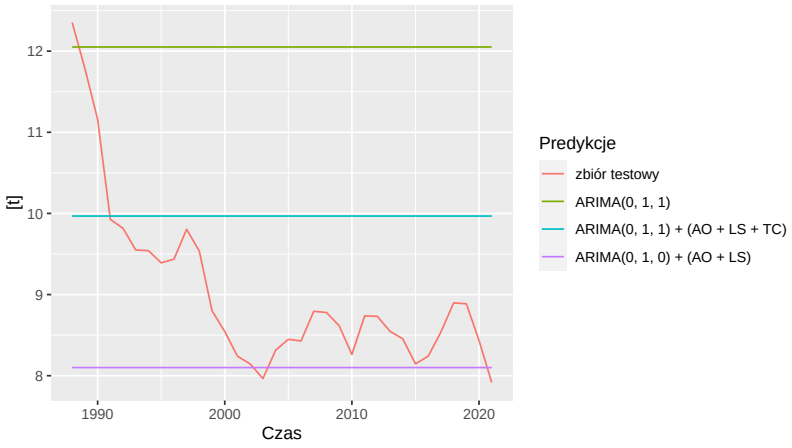
Uzyskane wartości kryteriów dokładności predykcji zestawiono w tabeli 2.5. Analizując otrzymane wyniki, można wysnuć kilka istotnych wniosków z przeprowadzonej analizy. Modele z częścią interwencyjną uzyskały lepsze dopasowanie do danych treningowych niż model ARIMA(0,1,1) czy ETS(M,N,N) (proste wygładzanie wykładnicze z multiplikatywną strukturą reszt). Jest to jednak obarczone kosztem niedokładnych prognoz dla modeli ARIMA(1,1,3) + (AO + LS +TC) oraz ARIMA(1,1,2) + (AO + LS +TC + IO), co

jest związane z nadmiernym dopasowaniem modeli do danych uczących. Najmniejsze błędy prognoz osiągnęły modele z częścią interwencyjną, odpowiednio: ARIMA(0,1,1) + (AO + LS + TC), ARIMA(0,1,0) + (AO + LS) oraz ARIMA(1,1,1) + (AO + LS +TC). Transformacja Boxa-Coxa nie przyniosła znaczącej poprawy dokładności predykcji, porównując do prognoz skonstruowanych w przypadku jej pominięcia.

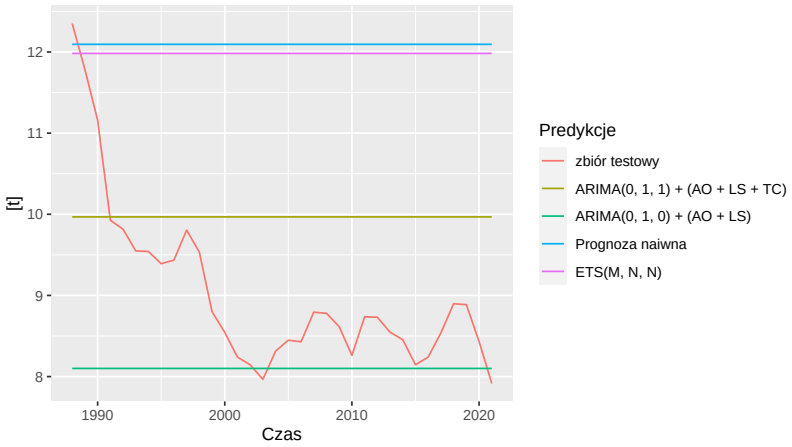
Tabela 2.5: Dopasowanie na zbiorze treningowym oraz dokładność prognoz na zbiorze testowym wybranych modeli (wyróżnione trzy najlepsze wyniki w każdej kategorii).

	Dane treningowe		Dane testowe	
	RMSE	MAPE	RMSE	MAPE
Bez transformacji potęgowej				
ARIMA(0, 1, 1)	0.744	36.503	3.184	34.999
ARIMA(1, 1, 3) + (AO + LS + TC)	**0.167**	**6.773**	3.832	41.476
ARIMA(0, 1, 1) + (AO + LS + TC)	**0.257**	**6.782**	**1.381**	**14.211**
ARIMA(1, 1, 2) + (AO + LS + TC + IO)	**0.341**	**8.868**	24.589	275.082
ARIMA(0, 1, 0) + (AO + LS)	0.527	11.201	**1.381**	**9.592**
metoda naiwna	0.766	35.351	3.225	35.463
ETS(M, N, N)	0.765	38.120	3.119	34.265
Uwzględniając transformację potęgową				
ARIMA(0, 2, 1)	0.782	34.773	4.943	53.636
ARIMA(0, 1, 5)	0.747	37.738	3.198	35.159
ARIMA(1, 1, 1) + (AO + LS + TC)	0.805	34.367	**1.739**	**18.823**
ARIMA(0, 1, 1) + AO	0.595	30.874	4.097	44.796
metoda naiwna	0.766	35.351	3.225	35.463
ETS(A, N, N)	0.743	35.165	3.222	35.421

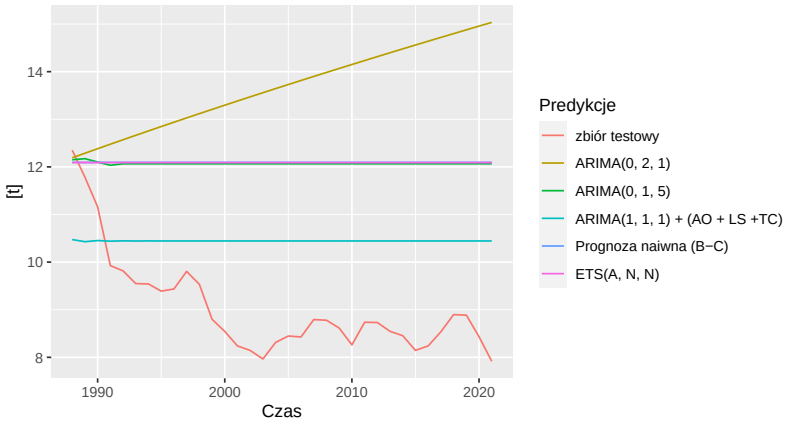
Rysunki 2.22, 2.23 oraz 2.24 przedstawiają skonstruowane prognozy, porównując je dla różnych grup modeli na tle rzeczywistych danych testowych.



Rysunek 2.22: Prognozy dla wybranych modeli ARIMA z naniesionym przebiegiem zbioru testowego.



Rysunek 2.23: Prognozy dla najdokładniejszych modeli ARIMA z naniesionym przebiegiem zbioru testowego, prognozą naiwną oraz prognozą na bazie modelu ETS.



Rysunek 2.24: Prognozy dla wybranych modeli uwzględniających transformację Boxa-Coxa z naniesionym przebiegiem zbioru testowego.

Wykresy 2.22, 2.23 i 2.24 reprezentują prognozy dla różnych podgrup badanych modeli. Na podstawie zarówno tych wykresów jak i tabeli 2.5 można jednoznacznie stwierdzić, że uwzględnienie modeli ARIMA z częścią interwencyjną jest jak najbardziej istotne. Modele ARIMA(0,1,1) + (AO + LS + TC) oraz ARIMA(0,1,0) + (AO + LS) cechują się najlepszym dopasowaniem oraz trafnością prognoz (ze względu na analizowane kryteria), dodając przy tym zaledwie 5 oraz 2 dodatkowe zmienne do modelu. Wyżej wspomniane modele oparte na analizie interwencji lepiej poradziły sobie z prognozowaniem spadku emisji CO2 w porównaniu do optymalnie wyznaczonych modeli ARIMA oraz ETS.

Kluczowym aspektem analizy (poszukiwania „najlepszego" modelu) było odpowiednie dobranie poziomu progowego C, by uniknąć nadmiernego dopasowania do danych oraz przeparametryzowania modelu. Wykorzystanie transformacji Boxa-Coxa spowodowało zmniejszenie liczby wykrytych obserwacji odstających, lecz nie spowodowało zwiększenia dokładności prognoz. Uwzględnienie typu innowacyjnej obserwacji odstającej prowadziło do najgorzej dopasowanych modeli i w konsekwencji najmniej dokładnych prognoz, co potwierdza wątpliwości dotyczące zastosowania parametryzacji IO w praktyce (patrz podrozdział 2.2 oraz [25]).

Rozdział 3

Metody wykrywania punktowych obserwacji odstających

Gdy analizujemy dane rzeczywiste, często niemożliwe bądź niepraktyczne jest pozostanie przy szerokiej taksonomii obserwacji odstających. Bazowe, intuicyjne podejście w procedurze wykrywania punktowych obserwacji odstających występujących w danych rzeczywistych jest oparte na nierówności:

$$|Y_t - \hat{Y}_t| > \tau, \tag{3.1}$$

gdzie jako anomalię określimy obserwację szeregu, dla której wartość bezwzględna różnicy Y_t i jej wartości prognozowanej lub dopasowanej ($\widehat{Y}_t$) przekracza określony poziom τ. Algorytmy wykrywania punktowych anomalii, które bazują na wzorze (3.1) nazywane są metodami opartymi na modelach (ang. *model-based*). Metody te dzielą się na te oparte na estymacji, które wykorzystują obserwacje przed i po konkretnej chwili T (dla której przeprowadzana jest procedura) oraz na metody oparte na prognozowaniu, gdzie obserwacje po chwili T nie są brane pod uwagę. Podejście oparte na prognozowaniu ma pewną przewagę względem podejścia opartego na estymacji, gdyż mamy wtedy możliwość używania algorytmu detekcji w czasie rzeczywistym.

3.1 Jednostronna oraz dwustronna metoda ruchomego okna

3.1.1 Metoda jednostronnej oraz dwustronnej mediany

Metoda jednostronnej oraz dwustronnej mediany została zaproponowana przez Meckesheimera i Basu [4]. Motywacją do stworzenia tej prostej procedury była potrzeba skutecznej oraz przejrzystej detekcji obserwacji odstających w danych rejestrowanych przez czujniki sygnałowe. Idea metody opiera się bezpośrednio na wzorze (3.1). $\widehat{Y}_t$ wyznaczane jest za pomocą dwustronnej bądź jednostronnej mediany. Metoda jednostronnej mediany wyznacza prognozowaną wartość w danym punkcie na podstawie $2k$ poprzednich obserwacji szeregu. Wartość tę można zapisać jako:

$$\widehat{Y}_t = med(\{Y_{t-2k}, \ldots, Y_{t-1}\}), \tag{3.2}$$

natomiast dla metody dwustronnej $\widehat{Y}_t$ wyznaczamy jako medianę k najbliższych sąsiadów danej obserwacji:

$$\widehat{Y}_t = med(\{Y_{t-k}, \ldots, Y_{t-1}, Y_{t+1}, \ldots Y_{t-k}\}). \tag{3.3}$$

Wybór wartości progowej τ jest zależny od zestawu danych z uwagi na fakt, że jest to metoda stricte umotywowana zastosowaniami praktycznymi. Autorzy sugerują że τ powinna być określona przez eksperta.

Obie metody zaliczają się do kategorii metod opartych na modelach. Metoda dwustronnej mediany bazuje na estymacji $\widehat{Y}_t$ dla okna obserwacji o szerokości $2k$. Natomiast

jej jednostronną modyfikację można przyporządkować do kategorii metod opartych na prognozowaniu Y_t na podstawie zaobserwowanych $2k$ obserwacji.

3.1.2 Metoda ruchomego okna

Idea opisana w podrozdziale 3.1.1 była podstawą do stworzenia własnych, bardziej ogólnych procedur, stosowanych jako metody referencyjne w analizie porównawczej przeprowadzonej w rozdziale 3.5. Motywacją do stworzenia algorytmu `OSWM` (ang. *one-sided window method*) oraz `TSWM` (ang. *two-sided window method*) była potrzeba szybkiej, w miarę możliwości bardziej automatycznej oraz elastycznej detekcji obserwacji odstających. Zamiast ruchomej mediany do wyznaczenia $\widehat{Y}_t$ jako bazowej funkcji używamy (domyślnie) ruchomej średniej, ze względu na mniejszą złożoność obliczeniową (brak konieczności sortowania wartości szeregu – szczególnie istotny dla długich szeregów). W procedurze `OSWM` oraz `TSWM` można alternatywnie wybrać inną dowolną funkcję służącą do predykcji czy estymacji kolejnej wartości. Dla przykładu, w przypadku `OSWM` mogą być to nawet proste modele z rodziny ARIMA lub warianty wygładzania wykładniczego (warto zwrócić uwagę na złożoność modelu przy detekcji obserwacji odstających dla długich szeregów). Dla τ ustalamy wartość domyślną jako $\tau = \alpha s$, gdzie s to próbkowe odchylenie standardowe. Krotność odchylenia α zależy od siły efektów generowanych przez anomalie występujące w danych; w rozdziale 3.5 przyjmowane są różne wartości α w zależności od charakterystyk badanej grupy szeregów. Ponadto będziemy zakładali, że wartości te należą do $(\frac{1}{2}, 5]$. Dodatkową opcją, którą jet uwzględniona w obu algorytmach jest możliwość przeprowadzenia dekompozycji MSTL (patrz podrozdział 1.5), by następnie właściwą detekcję przeprowadzać dla szeregu po usunięciu sezonowości oraz trendu.

3.2 Dekompozycja MSTL oraz reguła Tukeya

Kolejnym omawianym algorytmem jest metoda zaimplementowana jako funkcja `tsoutliers` w bibliotece `forecast` [22] pakietu `R` przez prof. Roba J. Hyndmana. Procedura wykorzystuje addytywną dekompozycję szeregu do postaci:

$$Y_t = T_t + S_t + Z_t, \tag{3.4}$$

gdzie T_t to składowa odpowiadająca za trend, S_t to komponent sezonowy, a Z_t to losowe fluktuacje (reszty) będące szeregiem stacjonarnym. Część sezonowa jest opcjonalna i może zawierać kilka wzorców sezonowych, które odpowiadają różnym okresowościom występującym w danych. Idea metody opiera się na estymacji S_t oraz T_t oraz usunięciu estymowanych efektów z danych i zastosowaniu procedury wykrywania obserwacji odstających dla szeregu residuów $\widehat{Z}_t = Y_t - \widehat{T}_t + \widehat{S}_t$ (patrz podrozdział 1.5).

Dla szeregów czasowych o okresie większym niż rok do estymacji komponentów S_t oraz T_t stosowana jest metoda odporna (ang. *robust*), która w pierwszym kroku wykorzystuje algorytm MSTL (ang. *Multiple Seasonal-Trend decomposition using Loess*) [3]. MSTL iteracyjnie estymuje część sezonową (szczegółowy opis procedury STL znajduję się w podrozdziale 1.5), a następnie ustalana jest siła sezonowości korzystając ze wzoru:

$$F_s = 1 - \frac{\mathrm{Var}(Y_t - \widehat{T}_t - \widehat{S}_t)}{\mathrm{Var}(Y_t - \widehat{T}_t)}. \tag{3.5}$$

Jeśli $F_s > 0.6$, uznajemy sezonowość za istotną oraz określamy nowy szereg po usunięciu części sezonowej jako $Y_t^* = Y_t - \widehat{S}_t$. Miara siły sezonowości jest uwzględniania na tym

etapie metody, gdyż w przypadku gdy sezonowość jest zbyt „słaba" $\widehat{S}_t$ może być nadmiernie dopasowany do danych, co spowoduje przesłonięcie części obserwacji odstających niedokładnie dopasowanym komponentem sezonowym. Natomiast, gdy $F_s \leq 0.6$, część sezonową określamy jako „słabą"i przyjmujemy, że $Y_t^* = Y_t$.

Kolejnym krokiem jest estymacja trendu dla szeregu Y_t^*. W tym celu prof. Hyndman proponuje użycie metody wygładzania szeregu zaproponowanej przez Friedmana (ang. *Friedman's super smoother*) [17]. Następnie właściwa procedura detekcji odbywa się na bazie szeregu $\widehat{Z}_t = Y_t^* - \widehat{T}_t$.

Końcowa reguła wykorzystana do identyfikacji obserwacji odstających opiera się na rozstępie międzykwartylowym – $IQR = Q_3 - Q_1$, gdzie Q_3 i Q_1 to odpowiednio trzeci i pierwszy kwartyl próbkowy. Obserwację nazwiemy odstającą, gdy:

$$\widehat{Z}_t < Q_1 - 3 \cdot IQR \quad \text{lub} \quad \widehat{Z}_t > Q_3 + 3 \cdot IQR. \tag{3.6}$$

Powyższe wartości graniczne były używane przez Tukey'a [35] w oryginalnej definicji obserwacji odstających (ang. *far out values*) dla wykresu pudełkowego.

Jeśli $\widehat{Z}_t$ ma rozkład normalny, wtedy prawdopodobieństwo, że obserwacja zostanie zidentyfikowana jako odstająca to w przybliżeniu 1 do 427000.

Obserwacje odstające zidentyfikowane w ten sposób zostają zastąpione przez liniowo interpolowane wartości obserwacji sąsiednich. Cała procedura jest następnie powtórzona dla tak przekształconego szeregu (w praktyce potrzebne są zwykle maksymalnie 2 iteracje całej procedury).

3.3 S-H-ESD

3.3.1 Test Grubbsa oraz test Rosnera

Punktem wyjścia w przypadku algorytmu S-H-ESD jest klasyczny test Grubbsa [18] (*maximum normalized residual test*), który pozwala na testowanie, czy w danych znajduje się (dokładnie jedna) obserwacja odstająca przy założeniu, że nasze dane pochodzą z rozkładu normalnego. Testujemy hipotezę zerową H_0, orzekającą, że w danych nie ma obserwacji odstającej, względem alternatywy – w danych jest dokładnie jedna obserwacja odstająca. Statystyka testowa G ma postać:

$$G = \frac{\max_{t=1,\ldots,n} |Y_t - \overline{Y}|}{s}, \tag{3.7}$$

gdzie $\overline{Y}$ oraz s to odpowiednio średnia oraz odchylenie próbkowe. Na poziomie istotności α hipoteza H_0 braku obserwacji odstającej zostaje odrzucona, gdy:

$$G > \frac{n-1}{\sqrt{n}} \sqrt{\frac{(t_{\alpha/2n\,,n-2})^2}{n-2+(t_{\alpha/2n\,,n-2})^2}}, \tag{3.8}$$

gdzie wyrażenie $t_{\alpha/2n\,,n-2}$ przedstawia wartość krytyczną dla rozkładu t-Studenta, o $n-2$ stopniach swobody oraz poziomie istotności $\alpha/2n$. Powyżej opisany test to wariant dwustronny, natomiast warianty jednostronne pozwalają na weryfikację hipotezy, czy maksimum bądź minimum jest obserwacją odstającą.

W praktyce bardzo rzadko spotyka się sytuację, by w danych rzeczywistych występowała zaledwie jedna obserwacja odstająca, co stało się motywacją do różnych ogólnień testu

Grubbsa. Modyfikację, która pozwala na testowanie, czy dokładnie k obserwacji jest obserwacjami odstającymi wprowadzili Tietjen i Moore [34] . Jest to niestety tylko częściowe rozwiązanie problemu, bo równie rzadko znamy dokładną liczbę anomalii występujących w danych.

Istotny postęp w tej kwestii osiągnął Rosner [31], który zaproponował test ESD (*Extreme Studentized Deviate*), który wymaga podania jedynie górnego ograniczenia r (gdzie $r < n/2$) na liczbę obserwacji odstających występujących w danych. Test ESD ma charakter iteracyjny; r statystyk testowych wyznaczanych jest ze wzoru:

$$R_i = \frac{\max_{t=1,\ldots,n} |Y_t - \overline{Y}|}{s}, \quad i = 1, 2, \ldots, r, \tag{3.9}$$

gdzie po zakończeniu pierwszej iteracji usuwamy obserwację maksymalizującą $|Y_t - \overline{Y}|$ i stosując analogiczną procedurę r razy dostajemy statystyki testowe $R_1, \ldots, R_r$. Przybliżone wartości krytyczne odpowiadające tym statystykom wyznacza wzór:

$$\lambda_i = \frac{(n-i)t_{p,\,n-i-1}}{(n-i+1)(n-i+1+t^2_{p,\,n-i-1})}, \quad i = 1, 2, \ldots, r, \tag{3.10}$$

gdzie $p = 1 - \frac{\alpha}{2(n-i+1)}$. Liczbę obserwacji odstających określamy znajdując największe i takie, że $R_i > \lambda_i$.

Rosner zbadał symulacyjnie, że aproksymowana wartość poziomu krytycznego jest bardzo dokładna powyżej 25 obserwacji, a rozsądna powyżej 15. Test ESD jest sekwencyjną modyfikacją testu Grubbsa, wprowadza on korektę wartości krytycznej w zależności od liczby testowanych przypadków. Zwykłe sekwencyjne testowanie przy użyciu testu Grubbsa (bez korekty) mogłoby skończyć się przedwcześnie, co jest kolejną wadą dostrzeżoną przez Rosnera.

Warto również dodać, że wspólnym ograniczeniem wszystkich wspomnianych wyżej testów jest założenie o (przybliżonej) normalności naszego szeregu, które (w rzeczywistości) nie musi być spełnione, np. dla szeregów finansowych, gdzie często spotykane są rozkłady ciężkoogonowe.

3.3.2 Modyfikacje testu Rosnera – S-ESD i S-H-ESD

Zespół analityków Twittera opracował metodę poprawiającą pewne aspekty testu ESD w kontekście wykrywania anomalii w danych rzeczywistych [21]. Zaproponowana metoda S-ESD polega na użyciu zmodyfikowanej dekompozycji STL (patrz podrozdział 1.5):

$$Y_t = S_t + med(Y_t) + Z_t, \tag{3.11}$$

gdzie estymowany składnik odpowiadający za trend zostaje zastąpiony wartością mediany; analogicznie jak w przypadku metody z podrozdziału 3.2, S_t to estymowana sezonowość, a Z_t stacjonarne reszty. Następnie na bazie estymowanych reszt szeregu $\widehat{Z}_t = Y_t - \widehat{S}_t - med(Y_t)$ stosowany jest test ESD.

Motywacją wprowadzonej modyfikacji jest chęć wykrycia lokalnych anomalii, które są przesłonięte przez bardziej znaczące globalne obserwacje odstające. Dzięki wykorzystaniu reszt $(\widehat{Z}_t)$ jako danych do detekcji obserwacji odstających, częściej (powołując się na autorów metody) spełniane jest założenie o (przybliżonej) normalności szeregu. Istotną wadą metody S-ESD są jednak słabe wyniki osiągane w przypadku szeregów o dużej liczbie anomalii.

By uniknąć tego problemu zaproponowano modyfikację metody S-ESD o nazwie S-H-ESD, która redefiniuje konserwatywne podejście testu ESD, zastępując w (3.9) próbkowe odchylenie oraz średnią (które mogą zostać sztucznie zawyżone, co spowoduje dużą liczbę anomalii sklasyfikowanych jako fałszywie negatywne) odpowiednio średnim odchyleniem medianowym (MAD) oraz medianą. Z uwagi na konieczność posortowania wartości szeregu przy użyciu metody S-H-ESD, dla szeregów o bardzo dużej liczbie obserwacji oraz niewielkiej liczbie anomalii zaleca się użycie procedury S-ESD, aby uniknąć problemów z dużą złożonością obliczeniową algorytmu S-H-ESD.

3.4 Model mieszanin gaussowskich oraz klasteryzacja

Kolejnym uwzględnionym przez nas algorytmem wykorzystywanym do wykrywania obserwacji odstających jest metoda będąca częścią biblioteki `tsrobprep` [29] dla pakietu `R`, stworzonej na potrzeby przygotowywania danych w postaci jednowymiarowych szeregów czasowych. Sam algorytm detekcji obserwacji odstających wykorzystuje techniki wielowymiarowej analizy danych, co wyróżnia go na tle pozostałych metod. Procedura jest dosyć skomplikowana i składa się z następujących kroków:

1 Opcjonalna dekompozycja szeregu na bazie metody STL (patrz podrozdział 1.5), dla pierwszego podanego okresu z wektora okresów dla sezonowości występujących w danych – parametr S. W kolejnym kroku algorytm wykorzystuje szereg Y_t^*, który może być wyjściowym szeregiem $Y_t^* = Y_t$ lub szeregiem reszt $Y_t^* = Z_t$, w przypadku wariantu uwzględniającego dekompozycje.

2 Następnym krokiem jest utworzenie dodatkowych zmiennych długości n na podstawie szeregu Y_t^*, które pomogą w nienadzorowanej (ang. *unsupervised*) detekcji anomalii. Kolejnymi zmiennymi (kolumnami macierzy) wykorzystywanymi w dalszych krokach algorytmu są:

   - *org.series* – szereg Y_t^*,
   - *seasonality* – deterministyczne sezonowości estymowane na bazie dekompozycji STL dla wektora S (liczba zmiennych zależna od długości wektora S),
   - *gradient* – suma wektorów $dyl = (0, \nabla Y_t^*)$ oraz $dyr = (\nabla Y_t^*, 0)$, gdzie ∇ oznacza operator różnicowania,
   - *abs.gradient* – wyraża się wzorem $|dyl| + |dyr| - |dyl - dyr|$,
   - *rel.gradient* – to stosunek zmiennej $pos.neg.gradient = dyl + sign(dyl) \cdot |dyr|$ do ruchomej wartości MAD (ang. *median absolute deviation*) dla dominującej sezonowości z wektora S,
   - *abs.seas.gradient* – zmienne wyliczane analogiczne do zmiennej *abs.gradient*, lecz dla zmiennych *seasonality* zamiast szeregu Y_t^* (liczba zmiennych zależna od długości wektora S).

   Na rysunku 3.1 znajduje się przykład ilustrujący utworzone zmienne dla szeregu czasowego zawierającego informacje dotyczące ruchu pasażerskiego w Twin Cities Metro Area w Minnesocie (dane ze zbioru NAB [2]).

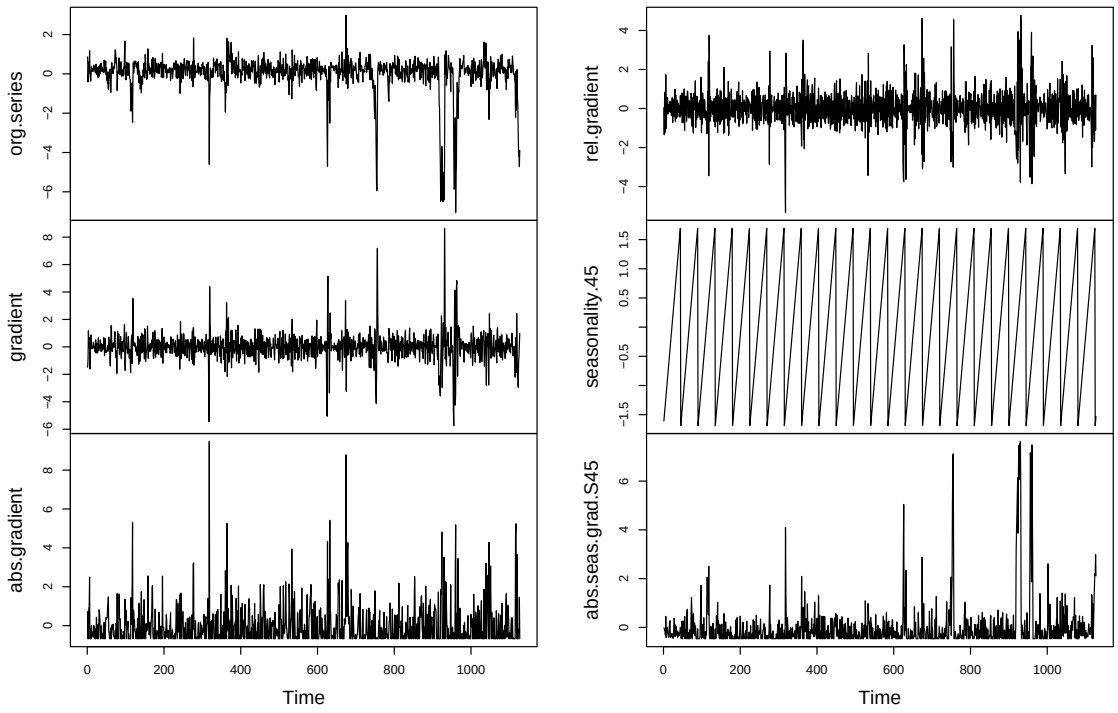
   Cała procedura tworzenia nowych zmiennych zwróci macierz M o wymiarach n x $3 + 2 \cdot seas.num$, (gdzie $seas.num$ to liczba komponentów sezonowych), do której

opcjonalnie można dołączyć składowe główne otrzymane w wyniku zastosowania algorytmu redukcji wymiaru – PCA (*Principal Component Analysis*).

3 Idea metody opiera się na klasteryzacji zainicjowanej na modelu mieszanin gaussowskich, by funkcję gęstości rozkładu każdej ze zmiennych z macierzy M można było zapisać w postaci modelu mieszaniny gaussowskiej, składające się z G komponentów, tzn.:

$$f_{MIX}(X, \Theta) = \sum_{k=1}^{G} \pi_k f_k(x_i, \theta_k), \tag{3.12}$$

gdzie $X = x_1, \ldots, x_n$ są ciągiem i.i.d, Θ to przestrzeń parametrów, π_k to prawdopodobieństwo, że obserwacja x_i należy do k-tego komponentu ($\sum_{k=1}^{G} \pi_k = 1$). Wyrażenie $f_k(x_i, \theta_k)$ wyznacza gęstość gaussowską dla obserwacji x_i i wektora parametrów k-tego komponentu mieszaniny – θ_k. Istotnym założeniem jest iż wszystkie G komponentów pochodzi z tego samego wielowymiarowego rozkładu normalnego $f_k(x_i, \theta_k) \sim \mathcal{N}(\mu_k, \Sigma_k)$.



Rysunek 3.1: Wykresy zaprojektowanych zmiennych używanych w wielowymiarowej detekcji anomalii dla szeregu czasowego zawierającego informacje dotyczące ruchu pasażerskiego w Twin Cities Metro Area w Minnesocie.

4 Spośród G komponentów mieszaniny jeden jest odpowiedzialny za modelowanie obserwacji odstających. Jest on wprowadzony przez sztuczne zwiększenie liczby obserwacji odstających poprzez „eksplozję" macierzy kowariancji. Wielkość tego klastra możemy zapisać jako $n_{out} = \xi\sqrt{n}$, gdzie ξ (domyślnie równe 2) jest parametrem algorytmu. Dodatkowym parametrem jest u (wartość domyślna równa 1), który występuje w wyrażeniu $u \log(n)^2$ kontrolującym proces „eksplozji" macierzy kowariancji. Ten

sztucznie wprowadzany komponent ma wielowymiarowy rozkład normalny, gdzie wektor średnich to średnie próbkowe kolumn macierzy M, a macierz kowariancji to macierz kowariancji dla M przemnożona przez wyżej wspomniane wyrażenie.

5 Do modelowania danych w postaci mieszaniny gaussowskiej autorzy metody używają algorytmu `Mclust` [32] (biblioteka `mclust`). Do estymacji parametrów skończonej gaussowskiej mieszaniny wykorzystuje się algorytm *Expectation-Maximization* (EM). By znaleźć odpowiednią liczbę komponentów G oraz strukturę kowariancyjną wykorzystywane jest kryterium BIC. Ponadto, aby przyśpieszyć proces estymacji parametrów wykorzystywane są wartości początkowe wyznaczone na bazie aglomeracyjnej hierarchicznej analizy skupień [16]. Z inicjalizacją tą wiąże się problem odporności oraz dużego wpływu na ostateczne parametry estymowane przez parametr EM. Autorzy metody zredukowali ten problem przeprowadzając estymacje na różniących się od siebie podzbiorach danych, by zwiększyć jej odporność oraz panować nad czasem obliczeń (co sprawia, że metoda jest obarczona pewnym błędem losowości).

6 Znając parametryzację komponentów modelu można następnie wyliczyć prawdopodobieństwo tego, że dana obserwacja znajduję się w skupieniu obserwacji odstających. Następnie w przestrzeni zmiennych obserwacje są zidentyfikowane jako odstające na podstawie wybranego poziomu odcięcia dla wyliczonego prawdopodobieństwa.

3.5 Porównanie skuteczności metod wykrywania anomalii punktowych

Analiza będzie miała na celu porównanie skuteczności metod detekcji obserwacji odstających dla rzeczywistych szeregów czasowych oraz dla danych symulowanych, do których sztucznie dodano anomalie. Dodatkowym aspektem analizy jest porównanie algorytmów detekcji obserwacji odstających dostępnych w pakiecie `R`. Oprócz gotowych funkcji oraz algorytmów wykorzystane będą prostsze, intuicyjne metody wykrywania obserwacji odstających (podrozdział 3.1.2), opracowane i zaimplementowane przez autora niniejszej pracy jako metody referencyjne specjalnie na potrzeby analizy porównawczej.

3.5.1 Zestawienie porównywanych algorytmów

W analizie uwzględnione zostaną dwie (własnej implementacji) metody referencyjne `OSWM` oraz `TSWM`, które jako argumenty przyjmują: szerokość okna m, bądź liczbę sąsiadów k, funkcję obliczającą wartości $\hat{Y}_t$ dla wybranej szerokości okna – odpowiednio $p.func$ oraz $e.func$ (domyślnie średnia próbkowa), funkcję, która odpowiada za ustalenie wartości progowej (τ – *treshold*) $t.func$ (domyślnie odchylenie próbkowe), której wartość jest przemnożona przez wybraną stałą α. Dodatkowym parametrem jest *decomp* (domyślnie `F`, co oznacza wartość logiczną `FALSE`), który odpowiada za przeprowadzenie dekompozycji MSTL i wykorzystanie w analizie reszt, otrzymanych po wyeliminowaniu składników sezonowych oraz trendu.

Kolejne dwie metody bazują na teście statystycznym opartym na resztach modelu ARIMA. Pierwszy, bardziej złożony wariant, to implementacja procedury iteracyjnej zaproponowanej przez Chena i Liu [11]. Jako alternatywne podejście wykorzystana będzie prostsza metoda opierająca się na teście dla jednokrotnie wyznaczonych reszt dla (optymalnego) modelu ARIMA wybranego przy użyciu funkcji `auto.arima` (dokładny opis

testu oraz procedury iteracyjnej znajduje się w podrozdziale 2.3). Wykorzystane będą implementacje wspomnianych metod dostępne w funkcjach `tso` oraz `locate.outliers`, gdzie dodatkowym parametrem uwzględnionym w porównaniu będzie wartość poziomu progowego *cval* dla obydwu metod.

Kolejna propozycja to prosta metoda zaimplementowana w funkcji `tsoutliers` w pakiecie `forecast`, dla której dodatkowo uwzględniona będzie wartość okresu przekazywana do dekompozycji MSTL.

Dla funkcji `detect_outliers` odpowiadającej podejściu omówionemu w podrozdziale 3.4 uwzględnimy parametry: S – wektor kolejnych okresów dla wyszczególnionych sezonowości, *replications* – liczba iteracji, *decomp* – czy dekompozycja STL dla pierwszego okresu z S powinna być wykonana przed procedurą (domyślnie `T`, co oznacza wartość logiczną `TRUE`) oraz *proba* – wartość progowa dla oszacowanego prawdopodobieństwa tego, że dana obserwacja jest anomalią.

Ostatnią uwzględnioną metodą jest procedura S-H-ESD, której odpowiada funkcja `ad_vec` o parametrach poziomu istotności *alpha* oraz okresu szeregu – *period* i wielokrotności okresu (dodatkowy parametr, uwzględniany przy więcej niż jednym komponencie sezonowym) – *landscape_period*.

Tabela 3.1: Wykaz algorytmów wykorzystanych w analizie.

algorytm	implementacja
`OSWM`(m, *alpha*, $t.func = sd$, $p.func = mean$, *decomp* = `F`)	implementacja własna
`TSWM`(k, *alpha*, $t.func = sd$, $e.func = mean$, *decomp* = `F`)	implementacja własna
`tso`(*cval*)	`tsoutliers`
`locate.outliers`(*cval*)	`tsoutliers`
`tsoutliers`($frequency$)	`forecast`
`detect_outliers`(S, *replications*, *proba*, *decomp* = `T`)	`tsrobprep`
`ad_vec`(*alpha* = 0.05, *period*, *landscape_period* = `NULL`)	`AnomalyDetection`

3.5.2 Miary wykorzystane do oceny skuteczności i porównania metod

Badane zagadnienie można sprowadzić do problemu predykcji, gdzie metody będą wskazywały czy dana wartość szeregu jest anomalią (1), bądź takową nie jest (0). Miary, których użyjemy do oceny jakości klasyfikacji to czułość (ang. *recall*):

$$rec = \frac{tp}{tp + fn}, \tag{3.13}$$

precyzja (ang. *precision*):

$$prec = \frac{tp}{tp + fp} \tag{3.14}$$

oraz ich ważona średnia harmoniczna – F1:

$$F1 = \frac{2}{\frac{1}{prec} + \frac{1}{rec}}. \tag{3.15}$$

We wzorach (3.13)-(3.15) tp oznacza prawdziwą liczbę anomalii wykrytych w badanym szeregu, natomiast fn to liczba anomalii, które nie zostały wykryte, a fp to liczba typowych

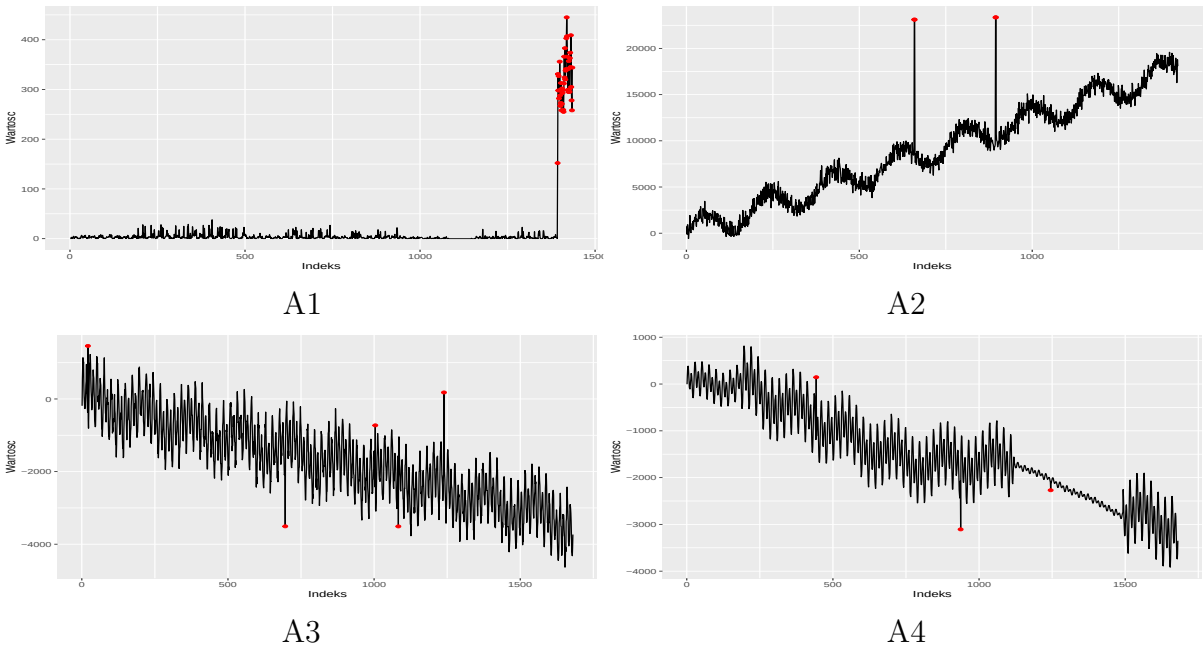
obserwacji, które zostały błędnie sklasyfikowane jako anomalie. Dodatkowo, gdy w danych nie będzie anomalii – $tp = 0$ oraz $fn = 0$ – a przy tym $fp = 0$, wtedy wszystkie miary przyjmują wartość 1. Natomiast, gdy $tp = 0$ i choć jedna z wartości fp i fn będzie różna od zera, wtedy wszystkie miary przyjmują wartość 0.

Powyższe miary będą uśrednione dla czterech jakościowo różnych podgrup szeregów, by szczegółowo zbadać czy i w jakim stopniu specyfika danych wpływa na skuteczność rozważanych algorytmów detekcji obserwacji odstających. Dodatkowym kryterium, które wykorzystamy do porównania zaproponowanych metod będzie uśredniony czas detekcji dla jednego szeregu (obliczenia zostały wykonane na procesorze Intel Core i5-4690K 3.5GHz).

3.5.3 Opis danych

Na potrzeby porównania metod dla jakościowo różnych szeregów użyjemy bogatego zbioru 367 szeregów czasowych Yahoo! [1] z oznaczonymi anomaliami. W zbiorze tym możemy wyodrębnić cztery zróżnicowane grupy (klasy) szeregów czasowych. Pierwszą grupę (A1) stanowią dane produkcyjne, zbierane co godzinę, w których eksperci oznaczyli, według własnej wiedzy oraz uznania, występujące w danych anomalie. Ich osąd nie jest jednoznaczny, dlatego interpretując wyniki dla tej klasy danych warto kierować się przede wszystkim średnią czułością dla 67 rzeczywistych szeregów czasowych.

Pozostałe trzysta szeregów składa się na trzy kolejne grupy (sto szeregów w każdym zestawie). Szeregi zaliczone do drugiej grupy (A2) mają losowo przydzielony trend, zakłócenie oraz sezonowość. Do trzeciej klasy (A3) zostały zaliczone dane charakteryzujące się złożoną wielookresową sezonowością (odpowiednio dwunastogodzinną, dobową oraz tygodniową) oraz addytywnym trendem wraz z zakłóceniem losowym, natomiast w czwartym zestawie (A4) dodatkowo uwzględnione są punkty zmian (ang. *change-points*). Rysunek 3.2 przedstawia przykładowe szeregi czasowe z naniesionymi anomaliami z każdej wyżej opisanej klasy danych z całego zbioru danych, natomiast w tabeli 3.2 przedstawione są ich podstawowe charakterystyki.



Rysunek 3.2: Wykresy reprezentujące przebieg oraz anomalie występujące w przykładowych szeregach zawartych w różnych grupach zbioru danych Yahoo!.

Tabela 3.2: Podstawowe własności szeregów czasowych Yahoo!

typ	długość	% anomalii
A1	741-1435	1.76
A2	1421	0.33
A3	1680	0.56
A4	1680	0.50

3.5.4 Dane produkcyjne

Wyniki analizy dla szeregów A1 danych rzeczywistych (pierwsza grupa opisana w podrozdziale 3.5.3) rzeczywistych zamieszczono w tabeli 3.3. Analizując skuteczność detekcji anomalii w tej podgrupie, kilka metod wypadło o wiele lepiej na tle pozostałych. Algorytm `OSWM` z ruchomą średnią, oknem o szerokości $m = 10$ i poziomem odcięcia $\tau = 2.8s$ uzyskał najwyższą wartość miary $F1$ oraz precyzji, natomiast funkcja `tsoutliers` z przyjętą sezonowością dobową osiągnęła najwyższą uśrednioną czułość. Inną wyróżniającą się metodą dla tej klasy szeregów czasowych jest algorytm S-H-ESD, który dla poziomu istotności $alpha = 0.01$ oraz dla sezonowości dobowej uzyskał drugą najwyższą wartość $F1$.

Tabela 3.3: Wyniki analizy dla danych rzeczywistych (A1) (najlepsze wyniki zostały oznaczone pogrubioną czcionką).

metoda	*prec*	*rec*	$F1$	czas $[s]$
`tso`($cval = 7$)	0.5596	0.3438	0.3388	39.9945
`locate.outliers`($cval = 8$)	0.4131	0.458	0.3542	0.5592
`locate.outliers`($cval = 7.5$)	0.3862	0.4647	0.3499	0.5659
`tsoutliers`($frequency = optimal$)	0.3228	0.7215	0.3682	0.0571
`tsoutliers`($frequency = 24$)	0.3056	**0.7651**	0.364	0.0641
`detect_outliers`($S = 12$, $replications = 10$, $proba = 0.5$)	0.3035	0.6268	0.3389	1.1929
`detect_outliers`($S = (24, 24 \cdot 7)$, $replications = 5$, $proba = 0.9$)	0.4406	0.5821	0.3987	0.7712
`detect_outliers`($S = (24, 24 \cdot 7)$, $replications = 10$, $proba = 0.9$, $decomp = F$)	0.4515	0.597	0.4387	1.4572
`TSWM`($k = 6$, $alpha = 1.5$, $t.func = sd$, $e.func = median$)	0.4429	0.4841	0.3396	0.0359
`TSWM`($k = 6$, $alpha = 1.8$, $t.func = sd$, $e.func = median$)	0.5097	0.4265	0.3537	0.0352
`TSWM`($k = 6$, $alpha = 2$, $t.func = sd$, $e.func = mean$)	0.4334	0.4351	0.3228	**0.0053**
`OSWM`($m = 12$, $alpha = 2$, $t.func = sd$, $p.func = median$)	0.3644	0.7191	0.4082	0.0592
`OSWM`($m = 10$, $alpha = 2.8$, $t.func = sd$, $p.func = median$)	0.591	0.596	0.5278	0.0547
`OSWM`($m = 10$, $alpha = 2.8$, $t.func = sd$, $p.func = mean$)	**0.6397**	0.6053	**0.5576**	0.0163
`ad_vec`($alpha = 0.05$, $period = 24$)	0.4255	0.4256	0.445	0.0583
`ad_vec`($alpha = 0.01$, $period = 24$)	0.4624	0.7077	0.4583	0.0565

Jeśli chodzi o średni czas detekcji, wyżej wymienione warianty oraz `TSWM` działają w podobnym czasie z minimalną przewagę metod opartych na przesuwającym się oknie (0.005s do 0.065s). Nieco wolniej działają funkcje `detect_outliers` oraz `locate.outliers` – średni czas detekcji jest w granicach 0.5s do 1.5s. Najbardziej problematyczna okazała się implementacja procedury Chena i Liu (`tso`), która charakteryzuje się największą złożonością obliczeniową wśród rozważanych metod, co potwierdza średni czas bliski 40s (kilka nietypowych szeregów zawyżyło średni czas), do tego dla kilku szeregów czasowych pojawiły się problemy z dopasowaniem odpowiedniego modelu ze względu na dużą liczbę obserwacji (średnie n wynosi około 1420).

Na wykresie 3.2 dobrze widać problematyczną strukturę danych A1, gdzie anomalie mogą tworzyć różnego rodzaju grupy oraz nietypowe nieregularności, które mogą sprawić trudność algorytmom. Dla procedur opartych na resztach modelu ARIMA z praktycznego punktu widzenia możliwa jest jedynie wspólna detekcja (oraz estymacja) dla jednego typu anomalii, ze względu na charakterystykę danych najlepiej przyjąć jest AO.

3.5.5 Dane symulowane

Na rysunku 3.2 można zaobserwować podstawowe własności poszczególnych podgrup szeregów syntetycznych. Dla wszystkich trzech podzbiorów obserwacje odstające mają charakter addytywny. Dla grupy A2 łatwo widać wyróżnione anomalie, natomiast dla klas A3 oraz A4, gdy dochodzą zjawiska takie jak zmiany intensywności szumu, zmiany trendu czy wielokrotne sezonowości zagadnienie identyfikacji anomalii wydaje się być dalekie od trywialnego.

W przypadku danych syntetycznych A2 (tabela 3.4), jeśli chodzi o uśrednioną precyzję oraz miarę $F1$ najlepiej spisała się metoda `TSWM` dla liczby sąsiadów $k = 6$, ruchomej mediany jako funkcji estymującej oraz wartości progowej postaci $\tau = 1.25s$. Najwyższą wartość uśrednionej czułości uzyskała funkcja `detect_outliers` z parametrami $S = (12, 12 \cdot 5)$, $proba = 0.8$ dla 10 iteracji. Blisko 80% $F1$ uzyskała średnio funkcja `tsoutliers`, nieco gorzej pod względem tego kryterium spisały się algorytmy `OSWM` (0.725), `detect_outliers` (0.6838) oraz `locate.outliers`. Znów jedną z najmniej skutecznych funkcji okazała się być `tso`, tym razem uzyskując nieco lepszy wynik niż najgorzej spisujący się dla tej grupy danych algorytm S-H-ESD.

Tabela 3.4: Wyniki analizy dla danych syntetycznych A2 (najlepsze wyniki zostały oznaczone pogrubioną czcionką).

metoda	*prec*	*rec*	$F1$	czas $[s]$
`tso`($cval = 8$)	0.6473	0.4464	0.5001	3.7989
`tso`($cval = 6$)	0.7711	0.5031	0.5749	3.9585
`tso`($cval = 5.5$)	0.7918	0.5116	0.58736	4.6186
`locate.outliers`($cval = 7.5$)	0.623	0.7617	0.6397	0.4162
`locate.outliers`($cval = 6.5$)	0.6247	0.7972	0.6562	0.4272
`locate.outliers`($cval = 6$)	0.605	0.8055	0.6359	0.4283
`tsoutliers`($frequency = optimal$)	0.8024	0.8256	0.7956	0.0509
`detect_outliers`($S = 24$, $replications = 5$, $proba = 0.7$)	0.5189	0.9217	0.6464	0.6986
`detect_outliers`($S = (12, 12 \cdot 5)$, $replications = 10$, $proba = 0.8$)	0.5402	**0.98**	0.6792	1.5566
`detect_outliers`($S = (24, 24 \cdot 7)$, $replications = 10$, $proba = 0.9$)	0.5562	0.95	0.6838	1.8163
`TSWM`($k = 6$, $alpha = 1.5$, $t.func = sd$, $e.func = median$)	0.9	0.8925	0.8916	0.0391
`TSWM`($k = 6$, $alpha = 1.25$, $t.func = sd$, $e.func = median$)	**0.9275**	0.945	**0.9285**	0.0406
`OSWM`($m = 10$, $alpha = 2.8$, $t.func = sd$, $p.func = median$)	0.7225	0.67	0.677	0.0483
`OSWM`($m = 20$, $alpha = 2.6$, $t.func = sd$, $p.func = median$)	0.7378	0.7116	0.701	0.0555
`OSWM`($m = 20$, $alpha = 2.5$, $t.func = sd$, $p.func = mean$)	0.7518	0.75	0.725	**0.0132**
`ad_vec`($alpha = 0.05$, $period = 24$)	0.31	0.2622	0.2774	0.0181
`ad_vec`($alpha = 0.1$, $period = optimal$)	0.31	0.3	0.3033	0.0193
`ad_vec`($alpha = 0.05$, $period = optimal$, $longterm\_period = optimal \cdot 7$)	0.5259	0.6511	0.534	0.0749

Nie znając a priori okresu badanego szeregu pojawia się problem, gdy chcemy skorzystać z metod opartych na dekompozycji MSTL bądź STL. Z tego względu wartość *optimal* dla funkcji `tsoutliers` oraz `ad_vec` została wyznaczona za pomocą funkcji `findfrequency`

z pakietu `forecast`, która określa okres dominującej sezonowości korzystając z narzędzi analizy spektralnej szeregów czasowych. Dodatkową opcją, którą warto rozważyć dla metody S-H-ESD jest określenie parametru *longtime_period*, który znacznie poprawia skuteczność detekcji w przypadku długich szeregów. Porównując średni czas detekcji anomalii dla pojedynczego szeregu, wartości dla grupy algorytmów: S-H-ESD, `OSWM`, `TSWM` oraz `tsoutliers` wahają się między 0.0135s, a 0.075s, natomiast funkcja `tso` spisuje się znacznie lepiej (dla tej grupy danych), gdyż średni czas waha się dla niej w okolicy 4s i zależy zależy przede wszystkim od doboru wartości krytycznej (*cval*). Natomiast średni czas działania metody `detect_outliers` w dużej mierze zależy od liczby iteracji algorytmu. Dla danych A2 również konieczne będzie uwzględnienie jedynie typu obserwacji odstającej AO (niemniej nie wpływa to na niekorzyść przeprowadzanej detekcji z uwagi na występowanie w danych A2 jedynie zmian impulsowych) `tso` ze względu na dużą złożoność obliczeniową i długość szeregów. Wyniki analiz dla danych A3 oraz A4 znajdują się w tabelach 3.5 oraz 3.6.

Tabela 3.5: Wyniki analizy dla danych A3 (najlepsze wyniki zostały oznaczone pogrubioną czcionką).

metoda	*prec*	*rec*	$F1$	czas $[s]$
`tso`($cval = 6$)	**0.9993**	0.9612	0.9739	5.7692
`tso`($cval = 5$)	**0.9993**	0.9906	0.9947	6.4464
`tso`($cval = 4$)	0.9961	0.9952	**0.9954**	7.8198
`locate.outliers`($cval = 6.5$)	0.9002	0.9364	0.8809	0.5325
`locate.outliers`($cval = 6$)	0.851	0.9695	0.8743	0.5834
`locate.outliers`($cval = 5.5$)	0.7964	0.9895	0.8456	0.558
`tsoutliers`($frequency = optimal$)	0.47	0.1907	0.2446	0.0262
`tsoutliers`($frequency = (24, 24 \cdot 7)$)	0.6011	0.9762	0.7254	0.2428
`tsoutliers`($frequency = 24$)	0.99	0.9477	0.9615	0.0498
`detect_outliers`($S = 24$, $replications = 10$, $proba = 0.9$)	0.3591	0.9375	0.5124	1.571
`detect_outliers`($S = (12, 24, 24 \cdot 7)$, $replications = 5$, $proba = 0.9$)	0.4248	0.9419	0.5396	0.9126
`detect_outliers`($S = (24, 24 \cdot 7)$, $replications = 10$, $proba = 0.8$, $decomp = F$)	0.6712	**0.9988**	0.7859	1.7435
`detect_outliers`($S = (24, 24 \cdot 7)$, $replications = 10$, $proba = 0.9$, $decomp = F$)	0.5491	0.9528	0.7946	1.6919
`TSWM`($k = 3$, $alpha = 0.7$, $t.func = sd$, $e.func = median$)	0.7175	0.7315	0.5547	**0.0114**
`TSWM`($k = 6$, $alpha = 5$, $t.func = sd$, $e.func = mean$, $decomp = T$)	0.981	0.9915	0.9816	0.0298
`TSWM`($k = 8$, $alpha = 5$, $t.func = sd$, $e.func = mean$, $decomp = T$)	0.981	0.9963	0.9842	0.03
`OSWM`($m = 12$, $alpha = 3$, $t.func = sd$, $p.func = median$, $decomp = T$)	0.7917	0.995	0.854	0.1208
`OSWM`($m = 10$, $alpha = 4$, $t.func = sd$, $p.func = mean$, $decomp = T$)	0.932	0.9959	0.95	0.0429
`OSWM`($m = 12$, $alpha = 5$, $t.func = sd$, $p.func = mean$, $decomp = T$)	0.9837	0.992	0.9836	0.0373
`ad_vec`($alpha = 0.1$, $period = 24$)	0.34	0.225	0.2541	0.0263
`ad_vec`($alpha = 0.05$, $period = 24$, $longterm\_period = 24 \cdot 7$)	0.7369	0.7097	0.6874	0.0674

Dla zestawu A3 role się nieco odwróciły. Algorytm `tso` uzyskał niemal perfekcyjny wynik pod względem każdego z rozważanych kryteriów, poza czasem, gdzie dalej potrzeba około 7-8s na detekcje. Skuteczność na poziomie $F1$ ponad 98% uzyskały metody oparte na estymacji oraz predykcji z przesuwającym się oknem. Procedura `tsoutliers` również uzyskała bardzo dobry rezultat – 96%. Nieco poniżej 0.8 uzyskał jednokrotny test dla reszt modelu ARIMA – `locate.outliers` oraz `detect_outliers`. Najgorzej dla tego zbioru sprawdził się algorytm S-H-ESD, przy czym dalej był to wynik na poziomie 0.7 dla miary $F1$.

Dla podobnego zbioru danych, który dodatkowo zawierał punkty zmiany (ang. *change-points*, które mogły zostać błędnie sklasyfikowane jako anomalie (wykrywanie punktów zmiany to problem innej natury, dla którego wykorzystuje się między innymi algorytmy

detekcji oparte na podejściu bayesowskim) skuteczność algorytmów nieco się pogorszyła. Średnia wartość $F1$ dla funkcji `tsoutliers` spadła poniżej 0.6, natomiast tylko metody `tso`, `OSWM` oraz `TSWM` uzyskały wartość $F1$ powyżej 0.8. W przypadku procedury Chena i Liu, ponownie pojawił się problem związany z długim czasem detekcji, gdyż średnio trwała ona blisko pół minuty, co w warunkach komercyjnych stanowi przepaść w porównaniu do szybkości rzędu setnych części sekundy w przypadku procedur opartych na idei ruchomego okna.

Tabela 3.6: Wyniki analizy dla danych syntetycznych A4 (najlepsze wyniki zostały oznaczone pogrubioną czcionką).

metoda	*prec*	*rec*	$F1$	czas $[s]$
`tso`($cval = 7.5$)	0.941	0.7414	0.8001	9.421
`tso`($cval = 5.5$)	0.9369	**0.84**	**0.8653**	27.9805
`locate.outliers`($cval = 6.5$)	0.6526	0.8323	0.6635	0.7059
`locate.outliers`($cval = 7$)	0.7002	0.7987	0.6716	0.6965
`locate.outliers`($cval = 7.5$)	0.731	0.7781	0.6845	0.689
`tsoutliers`($frequency = 24$)	0.6059	0.8012	0.596	0.0656
`tsoutliers`($frequency = (12, 24)$	0.562	0.797	0.5578	0.1696
`detect_outliers`($S = 24$, $replications = 5$, $proba = 0.8$)	0.4595	0.7923	0.5253	0.91
`detect_outliers`($S = (12, 24)$, $replications = 10$, $proba = 0.9$)	0.5349	0.8046	0.5813	1.689
`detect_outliers`($S = (24, 24 \cdot 7)$, $replications = 10$, $proba = 0.9$, $decomp = F$)	0.7272	0.8072	0.7255	1.9024
`TSWM`($k = 8$, $alpha = 5$, $t.func = sd$, $e.func = mean$, $decomp = T$)	0.9224	0.7809	0.8182	0.0306
`TSWM`($k = 4$, $alpha = 5$, $t.func = sd$, $e.func = mean$, $decomp = T$)	**0.9668**	0.7688	0.8344	**0.0301**
`OSWM`($m = 10$, $alpha = 5$, $t.func = sd$, $p.func = mean$, $decomp = T$)	0.9085	0.7926	0.8197	0.0405
`OSWM`($m = 16$, $alpha = 5$, $t.func = sd$, $p.func = mean$, $decomp = T$)	0.9135	0.7901	0.819	0.0344
`ad_vec`($alpha = 0.05$, $period = 24$, $longterm\_period = 24 \cdot 7$)	0.5671	0.6735	0.558	0.0685
`ad_vec`($alpha = 0.1$, $period = 24$, $longterm\_period = 24 \cdot 7$)	0.5638	0.7051	0.5662	0.07599

3.5.6 Wnioski dotyczącego analizy przedstawionych algorytmów detekcji oraz propozycje dalszych badań

- Procedura zaproponowana przez Chena i Liu (podrozdział 2.3.2) stała się przełomowym rozwiązaniem jeśli chodzi o poprawę jakości dopasowania modeli ARIMA, które pozwala na uwzględnienie w procesie estymacji efektów generowanych przez obserwacje odstające. Natomiast dla danych rzeczywistych (gdy $n > 300$) metoda ma pewne ograniczenia związane z kwadratową złożonością obliczeniową. Długi czas detekcji rekompensowany jest bardzo wysoką skutecznością, o ile jesteśmy w stanie stwierdzić, jaki typ obserwacji odstającej występuje w danych (najbezpieczniej zakładać AO, gdyż addytywne obserwacje nie przesłonią kilku anomalii będących blisko siebie w odróżnieniu od innych typów). Ważnym aspektem praktycznym jest także rozważny dobór wartości progowej dla wykonywanego iteracyjne testu, gdzie jej zbyt mała wartość może doprowadzić do zbyt czułej i o wiele dłuższej detekcji (szczególnie istotne, gdy $n > 1000$). Rozwiązaniem powyższych problemów, które wiążą się z wykorzystaniem funkcji `tso` byłaby implementacja algorytmu w szybszym (niskopoziomowym) języku programomania – np. przy użyciu integracji `R` z `C++` za pomocą pakietu `Rcpp`, która pomogłaby przyśpieszyć działanie procedury dla długich szeregów. Metoda nie nadaje się do detekcji anomalii w czasie rzeczywistym dla danych produkcyjnych o tysiącach obserwacji. By przystosować metodę do działania w takim środowisku należałoby prawdopodobnie aplikować ją dla ruchomego

okna o długości, która pozwalałaby na uchwycenie struktury autokorelacyjnej oraz sezonowości, pamiętając przy tym o dużej złożoności obliczeniowej procedury.

- Funkcja `locate.outliers`, będąca częścią funkcji `tso`, dla niektórych szeregów okazała się być lepszym rozwiązaniem z uwagi na czas detekcji oraz skuteczność. Jednokrotny test ma jednak swoje ograniczenia, gdyż w rzeczywistości nie musimy dostać nieobciążonych estymatorów ω (2.26)-(2.29) ze względu na wpływ sąsiednich obserwacji odstających. Dlatego kompletnym schematem wnioskowania statystycznego jest wyżej wspomniana procedura Chena i Liu wspólnej estymacji współczynników modelu ARIMA i efektów obserwacji odstających. Biorąc jednak pod uwagę detekcję obserwacji odstających w czasie rzeczywistym, o wiele mniej problematyczne jest wyliczenie n wartości statystyk τ (2.31)-(2.34) (w dalszym ciągu zakładamy, ze względów praktycznych, jedynie typ AO) oraz ustalenie rozsądnej wartości progowej C. W dalszym ciągu problem stanowi jednak dopasowanie modelu ARIMA dla danych o tak dużej liczbie obserwacji.

- Algorytm `tsoutliers` jest szybką i efektywną metodą identyfikacji anomalii punktowych. W przeprowadzonym badaniu empirycznym tylko dla czwartego zestawu danych algorytm odstawał jeśli chodzi o skuteczność od czołowych wyników. Potencjalny problem w używaniu tej metody powstaje, gdy nie wiemy zbyt dużo o pochodzeniu danych oraz ich sezonowości, natomiast wtedy możemy wykorzystać odpowiednie narzędzia pozwalające na identyfikację okresowości, np. takie jak funkcja `findfrequency` lub inne metody oparte na analizie spektralnej. Zaletą metody jest jej prostota jeśli chodzi o sam etap detekcji (wykorzystujemy kwartyle oraz rozstęp ćwiartkowy), dodatkowym atutem jest użycie dekompozycji MSTL, która charakteryzuje się odpornością ze względu na obserwacje odstające, co pozwala na przeprowadzenie procedury detekcji na bazie reszt otrzymanych po usunięciu regularnych składowych szeregu. Algorytm jak najbardziej nadaje się do identyfikacji obserwacji odstających w przypadku dużych zbiorów danych oraz w czasie rzeczywistym, ze względu na szybki czas działania (efektywną implementację bazującą na funkcjach `stl` oraz `supsmu` z biblioteki `stats`).

- Najbardziej odmiennym podejściem do detekcji anomalii z całej stawki algorytmów charakteryzuje się funkcja `detect_outliers`. Wadą metody w kontekście automatycznej detekcji jest bardzo duża liczba parametrów, która z drugiej strony zapewnia sporą kontrolę nad czasem działania oraz czułością detekcji. Algorytm przed inicjalizacją analizy skupień tworzy nowe obserwacje na podstawie wyjściowego szeregu, które w myśl założeń modelu mieszanin gaussowskich powinny być i.i.d., natomiast komponenty mieszaniny powinny mieć ten sam wielowymiarowy rozkład normalny. Dla danych rzeczywistych (szczególnie dla bardzo dużych n oraz nieregularnej struktury szeregu) założenie to może nie być spełnione. W przeprowadzonej analizie porównawczej, dla żadnej z jakościowo różnych grup szeregów, algorytm nie wykazywał znacznej poprawy skuteczności na tle prostszych metod, by w ten sposób uzasadnić swoją skomplikowaną konstrukcję. Z drugiej zaś strony pomimo użycia bardziej złożonych metod detekcji, algorytm daje proste w interpretacji wyniki w postaci prawdopodobieństw oraz ocenie bezpośredniego wpływu stworzonych zmiennych na detekcję danej obserwacji. Warto dodać, że sama idea projekcji szeregu jednowymiarowego do przestrzeni wielowymiarowej za pomocą ekstrakcji zmiennych, które objaśniają występowanie w danych anomalii w celu przeprowadzenia detekcji

dla jednowymiarowego szeregu jest innowacyjnym podejściem, które nie zostało jeszcze dokładnie zbadane czy zoptymalizowane. Pomysł autorów algorytmu daje spore możliwości do bardziej zaawansowanych analiz skuteczności, gdyż pakiet `R` oferuje szeroki wachlarz algorytmów wielowymiarowej detekcji anomalii, które potencjalnie mogłyby być wykorzystane również dla przypadków jednowymiarowych.

- Funkcje `OSWM` oraz `TSWM` zostały zaimplementowane i opracowane przez autora niniejszej pracy jako proste metody referencyjne dla bardziej skomplikowanych wariantów detekcji anomalii. Wyniki analizy okazały się nieco zaskakujące, gdyż można powiedzieć, że właśnie te metody spisały się najlepiej z całej reszty jeśli chodzi o stosunek czasu detekcji do skuteczności dla całej grupy badanych szeregów. Wielkim atutem metod opartych na ruchomym oknie jest ich prostota, czas działania oraz fakt, że zostały stworzone w celu detekcji w czasie rzeczywistym (metoda `TSWM` może w czasie rzeczywistym identyfikować obserwacje odstające z opóźnieniem k równym liczbie sąsiadów). Wadą metody jest w dalszym ciągu duża liczba parametrów, z których problematycznym może być wybór poziomu odcięcia τ, jeśli nie jesteśmy w stanie wcześniej zapoznać się ze strukturą analizowanych danych. Z przeprowadzonej analizy można wywnioskować również, że koniecznie do efektywnej detekcji jest zidentyfikowanie trendu oraz sezonowości, by zastosować odporną na obserwacje odstające dekompozycję MSTL. Alternatywą dla prognozowania przyszłej wartości na bazie prostej ruchomej średniej bądź mediany może być wykorzystanie modeli wygładzania wykładniczego lub modeli ARIMA aplikowanych dla ruchomego okna. Taka modyfikacja wiązałaby się z dłuższym czasem detekcji, lecz dawałaby szerszy wachlarz możliwości jeśli chodzi o ustalenie poziomu odcięcia (np. wykorzystując przedziały predykcyjne).

- Procedura S-H-ESD wydaje się być mało skuteczną propozycją na tle pozostałych algorytmów (dla żadnej z badanych grup nie przekroczyła progu $F1 = 0.7$). Ograniczenia związane z testem Rosnera (założenie o normalności danych) wydają się być kluczowym problemem nawet biorąc pod uwagę modyfikację S-H-ESD. Modyfikacją tego algorytmu, która być może poprawiłaby jego skuteczność mogłoby być użycie dekompozycji MSTL zamiast klasycznego wariantu STL, by postać danych poddanych testowaniu była pozbawiona wszystkich komponentów sezonowych (oraz oczywiście trendu). Test ESD nie jest dedykowanym narzędziem detekcji obserwacji odstających dla danych w postaci szeregu czasowego – nie uwzględnia w żaden sposób struktury autokorelacyjnej danych. Z pewnością dla innych zbiorów danych, w których po usunięciu trendu oraz sezonowości nie występują odstępstwa od zakładanej normalności, skuteczność procedury S-H-ESD jest o wiele lepsza, co udokumentowali twórcy metody.

Rozdział 4

Metody wykrywania anomalnych sekwencji

Część autorów [8], [6] proponuje taksonomię obserwacji odstających (w kontekście szeregów czasowych), w której oprócz punktowych obserwacji odstających wyróżnia się również sekwencje (podciągi) anomalne (ang. *discord*) oraz kontekstowe obserwacje odstające (do ich wykrywania używa się np. technik głębokiego uczenia). Anomalne podciągi są poniekąd generalizacją punktowych anomalii, gdyż punktowe obserwacje odstające to podciągi o długości 1. By formalnie zdefiniować to pojęcie oraz wprowadzić poszczególne algorytmy wykorzystywane do detekcji anomalnych sekwencji należy omówić kilka niezbędnych pojęć [26].

Definicja 4.1. Sekwencją C szeregu czasowego Y_t nazwiemy próbkowanie $m \leq n$ kolejnych obserwacji Y_t tak, że $C = Y_p, \ldots, Y_{p-m-1}$, gdzie $1 \leq p \leq n - m + 1$.

Definicja 4.2. Funkcją odległości pomiędzy dwoma sekwencjami M i C, nazwiemy symetryczną funkcję $Dist$, która zwraca nieujemną wartość będącą odległością między sekwencjami M i N.

Ważnym pojęciem w kontekście wyeliminowania trywialnych par sekwencji jest nietrywialne dopasowanie (ang. *non-self match*). W procedurze identyfikacji anomalnych sekwencji krytycznym jest by wyeliminować dopasowania sekwencji z innymi sekwencjami, które znajdują się bardzo blisko siebie.

Definicja 4.3. Sekwencja M o początku w chwili p będzie nietrywialnym dopasowaniem dla sekwencji C o początku w chwili q, gdy $|p-q| \geq m$ (przy czym zakładamy, że sekwencje są tej samej długości m).

Definicja 4.4. Sekwencja D, o długości m oraz początku w chwili l, szeregu czasowego Y_t jest nazywana anomalną (ang. *discord*), kiedy D ma największą odległość do najbliższego nietrywialnego dopasowania.

Z praktycznie punktu widzenia warto analizować k (gdzie $k \geq 1$) anomalnych sekwencji, zatem:

Definicja 4.5. K-tą sekwencją anomalną jest sekwencja szeregu czasowego Y_t, która ma k-tą największą odległość do najbliższego nietrywialnego dopasowania (dodatkowo zakładamy, że sekwencje anomalne nie mogą na siebie „nachodzić").

Najpopularniejszą funkcją odległości jest odległość euklidesowa zdefiniowana dla sekwencji $P = (p_1, \ldots, p_m)$ i $K = (k_1, \ldots, k_m)$ o długości m jako:

$$Dist_E = \sqrt{\sum_{i=1}^{m} (p_i - k_i)^2}. \tag{4.1}$$

Od tej pory, odnosząc się do funkcji odległości w opisie kolejnych algorytmów, będziemy zakładali, że chodzi o odległość (4.1). Dodatkowo, będziemy zakładali, że sekwencje przed wyznaczeniem odległości (4.1) są standaryzowane, by miały średnie próbkowe równe 0 oraz próbkowe odchylenia standardowe równe 1. Dla większości analizowanych szeregów rzeczywistych jest to istotne założenie przy porównywaniu ze sobą ich sekwencji.

4.1 Brute force

Najbardziej intuicyjnym, najprostszym sposobem na wyszukanie sekwencji anomalnych w szeregu czasowym wydaje się wyliczenie odległości każdej możliwej sekwencji do jej najbliższego nietrywialnego dopasowania. Podciąg, który uzyska największą odległość będzie sekwencją anomalną owego szeregu czasowego. Wyżej opisane rozwiązanie jest ideą metody *Brute force* [26], na którą składają się dwie pętle. Pierwsza iteruje po wszystkich możliwych sekwencjach szeregu (określonej długości m). Druga natomiast wylicza dla każdej z nich odległość do najbliższego nietrywialnego dopasowania. Algorytm 1 przedstawia szczegółowy opis procedury *Brute force*.

Algorytm 1 Brute force

$best\_so\_far\_dist \leftarrow 0$ {największa aktualnie odległość do najbliższego sąsiada}
$best\_so\_far\_loc \leftarrow NULL$ {odpowiadający jej początek sekwencji}
for $p = 1$ **to** $n - m + 1$ **do**
  $NN\_dist = \infty$
  for $q = 1$ **to** $n - m + 1$ **do**
    if $|p - q| \geq m$ **then**
      if $Dist(Y_p, \ldots, Y_{p+m-1}\ Y_q, \ldots, Y_{q+m-1}) < NN\_dist$ **then**
        $NN\_dist \leftarrow Dist(Y_p, \ldots, Y_{p+m-1}\ \ Y_q, \ldots, Y_{q+m-1})$
      end if
    end if
  end for
  if $NN\_dist > best\_so\_far\_dist$ **then**
    $best\_so\_far\_dist \leftarrow NN\_dist$
    $best\_so\_far\_loc \leftarrow p$
  end if
end for
return $[best\_so\_far\_loc,\ best\_so\_far\_dist]$

Metoda *Brute force* ma jedną bardzo dużą wadę, a jest nią złożoność obliczeniowa $O(n^2)$, co dla bardzo dużej liczby obserwacji (np. setki tysięcy uderzeń serca dla danych z EKG) powoduje, że jej praktyczne zastosowanie jest mocno ograniczone.

4.2 HOT-SAX

Kluczową, by poprawić złożoność obliczeniową algorytmu *Brute force*, jest obserwacja, że w pętli wewnętrznej tak naprawdę nie musimy za każdym razem znajdować prawdziwego najbliższego sąsiada względem sekwencji wybranej w pętli zewnętrznej. Jeśli znajdziemy sekwencję, która jest bliższa obecnemu kandydatowi niż $best\_so\_far\_dist$, możemy wtedy swierdzić, że ten kandydat na pewno nie będzie sekwencją anomalną.

Drugim istotnym czynnikiem, który pozwoli zredukować złożoność obliczeniową algorytmu będzie ustalenie odpowiedniej kolejności wyboru kandydatów na sekwencję anomalną w pętli zewnętrznej oraz ustalenie kolejności dobieranych sekwencji w pętli wewnętrznej, by jak najszybciej móc ją przerwać.

4.2.1 Reguły heurystyczne

Twórcy algorytmu HOT-SAX [26] proponują trzy wstępne podejścia do opracowania reguł, które zredukują złożoność obliczeniową algorytmu *Brute force*.

Pierwsza z nich – losowa – jak nazwa wskazuje polega na losowym przetasowaniu wybieranych sekwencji. Złożoność obliczeniowa dla tej metody będzie wahać się między $O(n^2)$, a $O(n)$.

Kolejną propozycją jest tak zwana „magiczna" reguła (ang. *magic rule*), która zakłada dużo szczęścia lub posiadanie zaprzyjaźnionej wyroczni, gdyż dostajemy idealne posortowanie sekwencji. Dla pętli zewnętrznej sekwencje będą posortowane malejąco względem odległości do najbliższego nietrywialnego dopasowania, co pozwoli opuścić pętle po pierwszej iteracji. Natomiast sekwencje w drugiej pętli będą posortowane rosnąco względem odległości do obecnego kandydata w pętli, co również pozwoli na opuszczenie pętli po pierwszej iteracji. Tak opisany idealny scenariusz ma złożoność obliczeniową $O(m) + O(m) = O(m)$.

Ostatnim badanym scenariuszem przez twórców jest tak zwana „przewrotna" reguła (ang. *perverse rule*), przypadku której otrzymujemy najgorszą możliwą kolejność sekwencji w dwóch pętlach. Porównując do przypadku „magicznej" reguły, by uzyskać taki scenariusz należy zmienić kolejność z malejącej na rosnącą w pętli zewnętrznej oraz z rosnącej na malejącą w pętli wewnętrznej. Reguła ta ma tę samą złożoność co algorytm *Brute force* tzn. $O(n^2)$, dlatego, że szukane sekwencje będą zawsze odnajdywane w ostatnich iteracjach.

Można wyobrazić sobie pełne spektrum metod, których celem będzie poprawienie skuteczności heurystycznej na rzecz jak najlepszej aproksymacji „magicznej" reguły. Główną motywacją w przypadku algorytmu HOT-SAX jest obserwacja iż wystarczy, by największa odległość do najbliższego nietrywialnego dopasowania była znaleziona w kliku pierwszych iteracjach. To samo tyczy się najmniejszej odległości dla pętli wewnętrznej.

Propozycja aproksymacji (opartej na powyżej opisanej obserwacji) „magicznej" reguły heurystycznej opiera się na wykorzystaniu kodowania sekwencji otrzymanego przy użyciu algorytmu SAX.

4.2.2 SAX

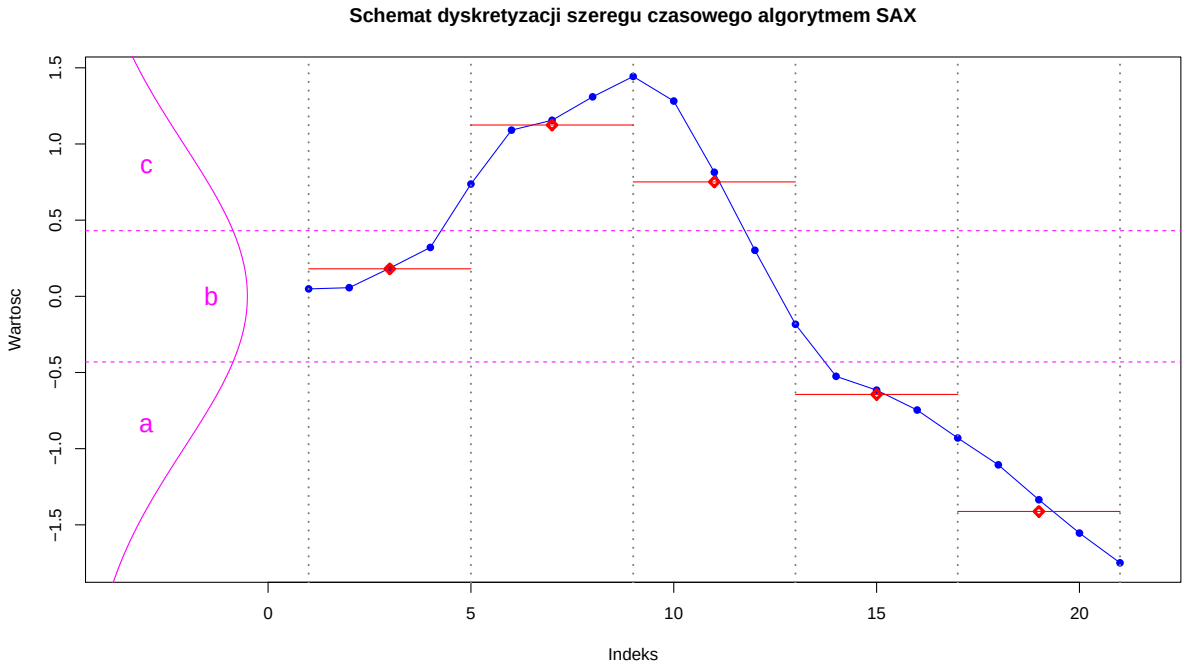
SAX (Symbolic Aggregate ApproXimation) [27] to algorytm, który pozwala na dyskretyzację szeregu czasowego (w tym również jego sekwencji). Szereg Y_t o długości n może być przedstawiony w w-wymiarowej przestrzeni jako wektor $\overline{Y} = (\overline{Y_1}, \ldots, \overline{Y_w})$, gdzie $\overline{Y_i}$ jest postaci:

$$\overline{Y_i} = \frac{w}{n} \sum_{\frac{n}{w}(i-1)+1}^{\frac{ni}{w}} Y_j. \tag{4.2}$$

Można powiedzieć, że algorytm przekształca dane z n do w-wymiarowej przestrzeni, a same dane są ustrukturyzowane w formie ramek o długości w. Następnie wyliczana jest średnia wartości umieszczonych w danej ramce, a wektor tych średnich staje się nową reprezentacją

naszego szeregu o zredukowanej długości. Taka reprezentacja szeregu określana jest w literaturze [27] reprezentacją PAA (*Piecewise Aggregate Approximation*).

Następnie, mając reprezentację PAA badanego szeregu czasowego, potrzebne będą kolejne transformacje, by uzyskać jego dyskretną formę. W szczególności odpowiednie efekty mogłoby dać zastosowanie metody, która przyporządkowywałaby określone symbole z tym samym prawdopodobieństwem. W [26] twórcy przeprowadzili symulacje pokazując, że większość standaryzowanych sekwencji szeregów czasowych będzie mieć rozkład normalny (gdy ten warunek nie będzie spełniony metoda dalej jest poprawna, natomiast charakteryzuje się wtedy mniejszą skutecznością). Zatem na podstawie kwantyli odpowiednich rzędów rozkładu $\mathcal{N}(0, 1)$ można podzielić zbiór wartości szeregu czasowego (jego sekwencji; pamiętając że zostały wcześniej standaryzowane) na segmenty o równym prawdopodobieństwie. Zbiór punktów podziału można zapisać jako $B = (b_1, \ldots, b_{\alpha-1})$ (gdzie b_0 oraz b_α są zdefiniowane jako odpowiednio $-\infty$ oraz ∞), gdzie $\int_{b_i}^{b_{i+1}} \frac{1}{\sqrt{2\pi}} \exp(\frac{-x^2}{2})dx = \frac{1}{\alpha}$. Rysunek 4.1 ilustruje schemat dyskretyzacji na podstawie algorytmu SAX dla przykładowego szeregu czasowego.



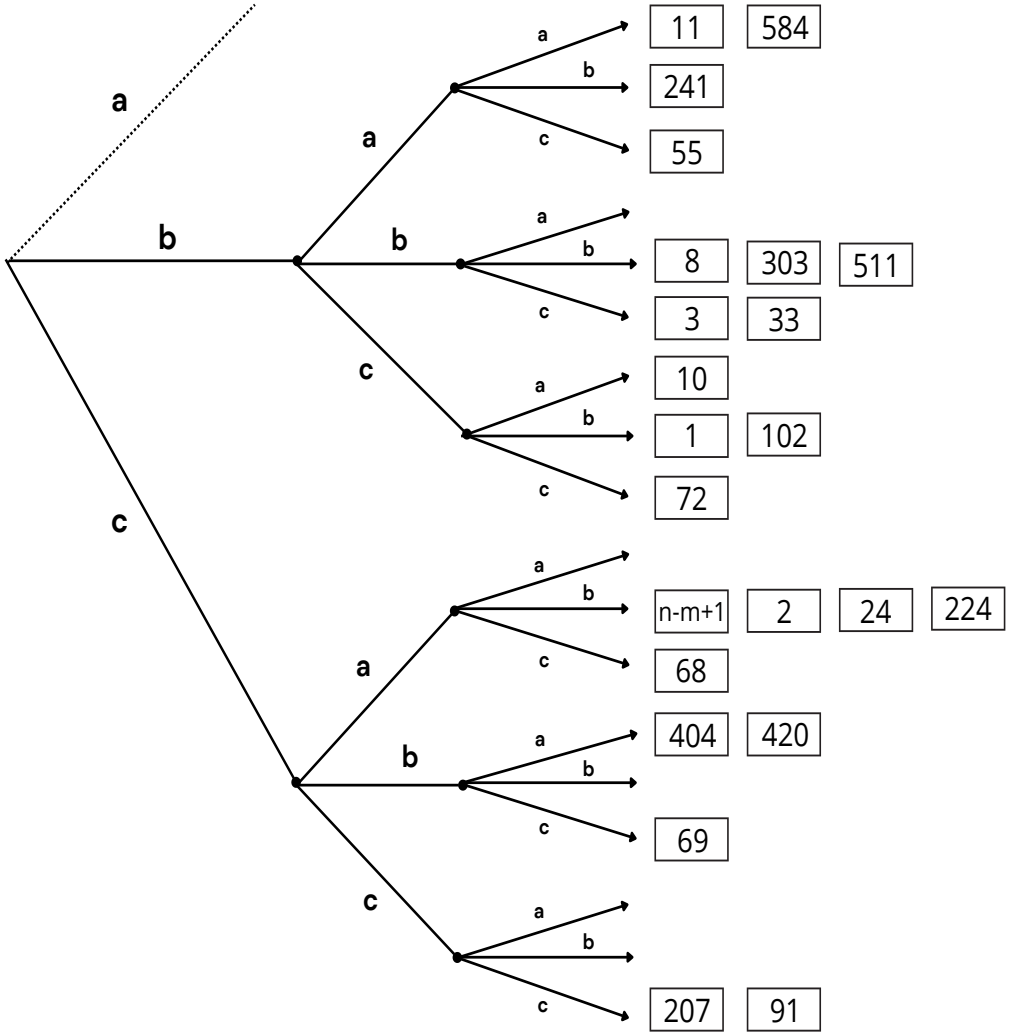
Rysunek 4.1: Wizualna prezentacja dyskretyzacji symulowanego szeregu czasowego z 21 obserwacjami do postaci sekwencji **bccaa**, gdzie parametr, algorytmu PAA, $w = 5$ (liczba uśrednień) oraz $\alpha = 3$, gdzie α jest parametrem wielkości alfabetu

4.2.3 Aproksymacja magicznej reguły heurystycznej (*magic heuristic*)

Aproksymacja „magicznej" reguły dla pętli zewnętrznej opiera się na zakodowaniu sekwencji badanego szeregu czasowego dla ustalonej szerokości okna m. Dodatkowe parametry algorytmu SAX to α – wielkość alfabetu oraz ω – długość słowa (liczba uśrednień dla reprezentacji PAA). Warto dodać, że dobór parametrów α oraz ω wpływa jedynie na efektywność metody, natomiast finalny rezultat zależy tylko od doboru m.

Pierwszym krokiem aproksymacji jest przypisanie każdej sekwencji indeksu z pierwszą obserwacją wyjściowego szeregu oraz zliczenie liczby wystąpień danego kodowania. Można sobie tę strukturę wyobrazić jako macierz M o $(n–m)+1$ wierszach oraz 2 kolumnach reprezentujących kodowanie oraz liczbę sekwencji kodowanych w ten sam sposób.

Kolejną potrzebną strukturą będzie dendrogram, którego liście będą zawierać indeksy początków sekwencji, które powtórzyły się dla danych kodowań SAX. Każda lista słów będzie składać się z indeksów odpowiadających początkowi sekwencji. Przykładowa postać dendrogramu dla zdyskretyzowanego szeregu przedstawiona jest na rysunku 4.2.



Rysunek 4.2: Przykładowa postać dendrogramu HOT-SAX dla hipotetycznego szeregu czasowego; w tym przykładzie alfabet zawiera 3 litery ($\alpha = 3$), natomiast długość słowa (liczba uśrednień PAA) jest również równa 3 ($\omega = 3$).

Analizując powyższą strukturę dla hipotetycznego szeregu zauważamy, że sekwencje kodowane np. jako **baa** zaczynają się od 11 oraz 584 obserwacji (oczywiście ich długość jest równa ustalonej wartości m), natomiast jedyna sekwencja kodowana jako **cbc** rozpoczyna się od 69 obserwacji.

Zaskakującym faktem jest to [26], że obydwie struktury udaje się uzyskać przy złożoności O(n).

Sama aproksymacja „magicznej” heurystyki polega na przeszukaniu drugiej kolumny

macierzy M w celu znalezienia najmniejszej liczby tak samo kodowanych sekwencji (w praktyce będzie to prawie zawsze 1) oraz przekazaniu wszystkich sekwencji, które powtórzyły się tylko raz, jako pierwszych w kolejności do iterowania. Po wyczerpaniu takich sekwencji reszta dobierana jest w sposób losowy.

Intuicyjnie można oczekiwać, że różniące się od reszty sekwencje będą przypisane wyjątkowym albo rzadko występującym kodowaniom SAX. Zaczynając iteracje właśnie od tych słów zwiększamy szansę na uzyskanie dużej wartości zmiennej $best\_so\_far\_dist$, co pozwoli na częstsze spełnienie odpowiedniego warunku (linia 7 w Algorytmie 1), co poskutkuje szybszym zakończeniem pętli wewnętrznej.

Aproksymacja „magicznej" reguły dla pętli wewnętrznej opiera się na rozpoczęciu iteracji dla aktualnego kandydata od tych sekwencji, które mają to samo kodowanie SAX. Po wyczerpaniu takich kandydatów reszta dobierana jest w sposób losowy.

Idea wykorzystywana w przypadku tej reguły jest również dosyć intuicyjna, gdyż zaczynamy iterować od najbardziej podobnych do siebie sekwencji, po to, by znaleźć sekwencję, dla której wartość odległości będzie mniejsza od obecnej wartości zmiennej $best\_so\_far\_dist$. Wszystko w tym celu, by jak najszybciej zakończyć działanie pętli wewnętrznej.

4.3 Profil macierzowy

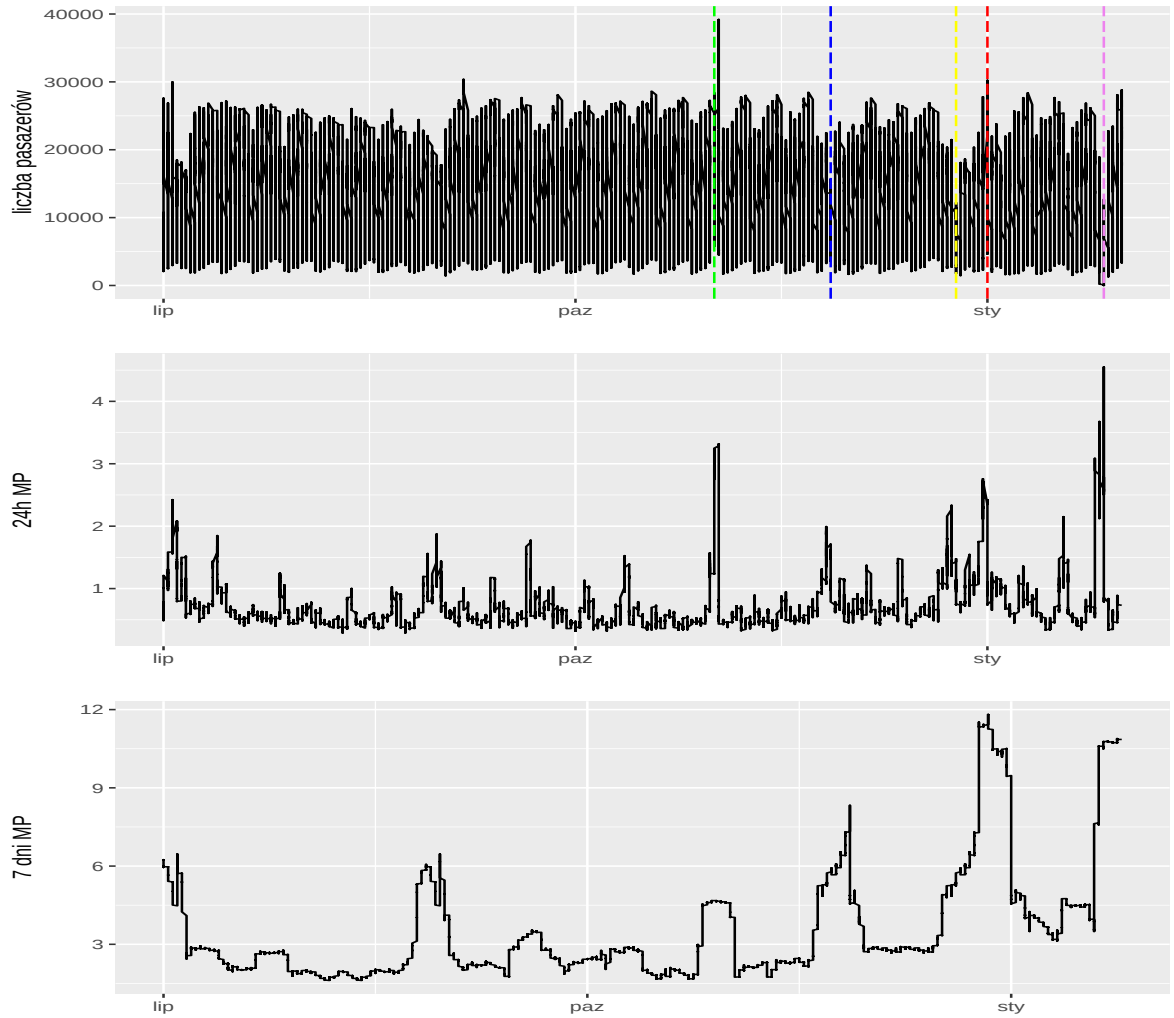
Profil macierzowy (ang. *matrix profile*) jest relatywnie nową strukturą danych, zaproponowaną przez Eamonn Keogha oraz Abdullah Mueen w 2016 [38]. Jest to elastyczna miara podobieństwa sekwencji szeregu, która ma szerokie zastosowanie w pozyskiwaniu wiedzy w oparciu o dane w postaci szeregów czasowych.

Profil macierzowy składa się z dwóch części: profilu odległości (ang. *distance profile*) oraz profilu indeksów (ang. *index profile*). Profil odległości jest wektorem kolejnych odległości euklidesowych (standaryzowanych) sekwencji o długości m (czyli dla i-tego indeksu szeregu porównywana będzie sekwencja $(Y_i, \ldots, Y_{i+m-1})$ względem jej najbliższego sąsiada). Obliczenia wykonywane są dla ruchomego okna o długości m, zatem wektor będzie długości $n - m + 1$. Profil indeksów zawiera indeksy początku sekwencji najbardziej podobnej do ustalonej sekwencji z profilu odległości.

Przedstawiając kolejne kroki algorytmu, w pierwszym wyliczana jest odległość euklidesowa dla i-tej sekwencji względem całego szeregu. Kolejnym etapem jest wyeliminowanie trywialnych dopasowań, tym razem minimalną odległość wyznacza długość okna dla indeksów znajdujących się przed oraz po danej sekwencji. Następnie profil odległości zostaje uzupełniony o odległość i-tej sekwencji do jej najbliższego sąsiada, a profil indeksów o indeks najbliższej sekwencji.

Na podstawie powyższego opisu można wywnioskować, że anomalna sekwencja to taka, która posiada największą wartość profilu odległości (największą odległość do najbliższej sąsiadującej sekwencji). Analogicznie największe k odległości da odpowiednie k kolejnych sekwencji anomalnych ze względu na odległość euklidesową.

Rysunek 4.3 prezentuje wykorzystanie profilów macierzowych do detekcji anomalnych zdarzeń wpływających na liczbę pasażerów w nowojorskich taksówkach (dane użyte w wizualizacji pochodzą ze zbioru NAB [2]). Profil macierzowy oddaje dynamikę zmian, która towarzyszyła w tym przypadku liczbie pasażerów. Piki w wartościach profilu odległości dobrze oddają umiejscowienie ważnych wydarzeń dla życia Nowego Jorku, które miały bezpośredni wpływ na liczbę pasażerów w kolejnych interwałach czasowych.



Rysunek 4.3: Liczba pasażerów nowojorskich taksówek w drugiej połowie 2014 roku (dane zbierane co pół godziny) oraz profil macierzowy dzienny oraz tygodniowy. Przerywane linie sugerują kolejne anomalne okresy: zielona – maraton nowojorski, niebieska – Święto Dziękczynienia, żółta – Święta Bożego Narodzenia, czerwony - Nowy Rok, fioletowy - burzę śnieżną.

4.3.1 Algorytmy stosowane do obliczania profilu macierzowego profil macierzowy

Kluczowym zagadnieniem o charakterze technicznym jest optymalizacja złożoności obliczeniowej czasowej oraz pamięciowej poszczególnych kroków algorytmu wykorzystanego do wyznaczenia profilu macierzowego. Dotychczas zaproponowanych zostało kilka różnych algorytmów o odmiennych własnościach.

Pierwszym algorytmem, który zaproponowała grupa Eamonna Keogha był algorytm *STAMP* [38] (ang. *Scalable Time Series Anytime Matrix Profile*). Algorytmem pośrednim, który służy do wyliczania podobieństwa pomiędzy sekwencjami jest algorytm *MASS* (ang. *Mueen's Algorithm for Similarity Search*) [28], który wykorzystuje szybką transformację Fouriera do wyliczania iloczynów skalarnych poszczególnych sekwencji. Natomiast określenie *Anytime* odnosi się do losowego iterowania po sekwencjach, dla których obliczane jest podobieństwo względem nietrywialnych dopasowań.

Algorytmem, który znacząco przyśpieszył czas obliczeń względem $STAMP$ jest algorytm $STOMP$ [42] (ang. *Scalable Time series Ordered Matrix Profile*), który wykorzystuje fakt, że dla kolejnych dwóch iteracji $m-1$ obserwacji pokrywa się. Pozwoliło to utworzyć algorytm o złożoności obliczeniowej $O(n^2)$ oraz pamięciowej $O(n)$.

Od momentu opublikowania pierwszego artykułu dotyczącego profili macierzowych, który przedstawiał idee oraz algorytm $STAMP$, ukazało się ponad 20 publikacji, które rozszerzają tematykę związaną z tą strukturą danych oraz wprowadzają coraz to szybsze algorytmy obliczające profile jak np. $SCRIMP++$ [41].

4.4 Studium przypadku: wykrywanie anomalnych sekwencji dla danych medycznych

4.4.1 Cel analizy

Celem analizy będzie przedstawienie, omówionych w części teoretycznej, algorytmów wykrywania sekwencji anomalnych w szeregach czasowych oraz porównanie ich efektywności oraz dokładności. W tym celu wykorzystamy dane [24] z badania elektrokardiografem (EKG) zawierające zdiagnozowaną przez eksperta anomalię odpowiadającą rytmowi przedsionkowemu. Badany szereg posiada relatywnie dużą liczbę obserwacji, by uwypuklić różnice związane ze złożonością obliczeniową algorytmów.

4.4.2 Porównywane algorytmy

W tabeli 4.1 zestawiono funkcje, które zostały wykorzystane do przeprowadzenia analizy.

Tabela 4.1: Algorytmy użyte podczas analizy

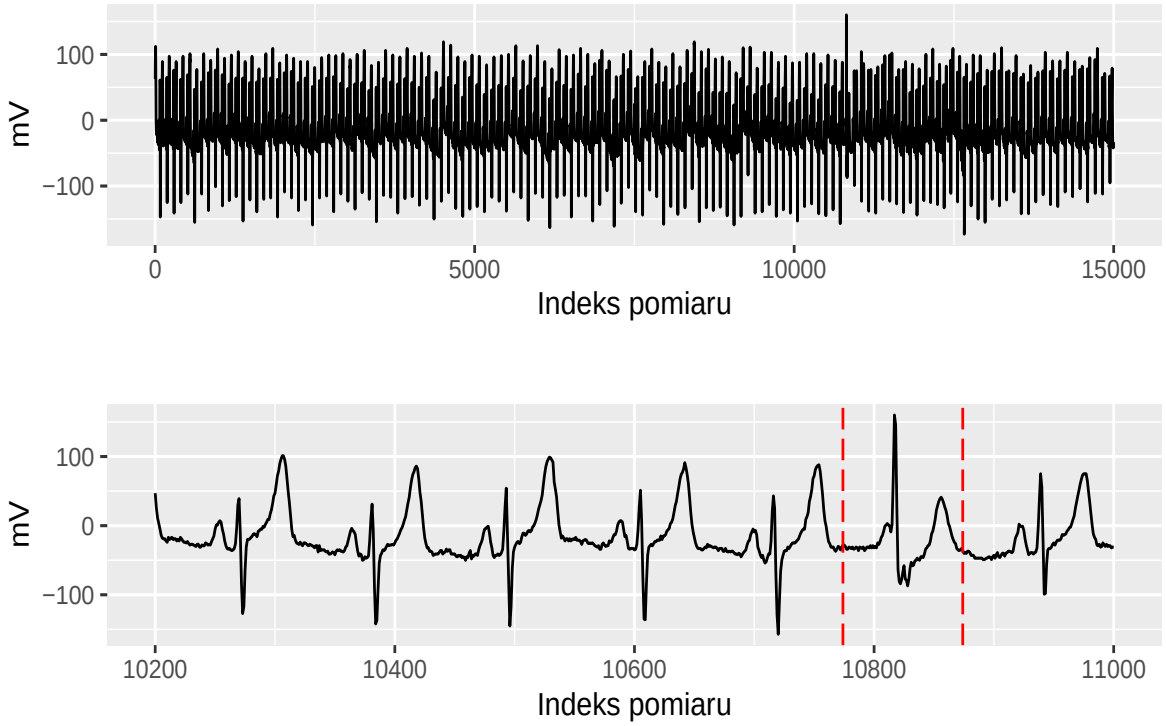
algorytm	funkcja	pakiet
HOT-SAX	`find_discords_hotsax`(Y_t, m, ω, α)	`jmotif`
Profil macierzowy	`tsmp`$(Y_t, m, mode =$ `STOMP`$)$	`tsmp`
Brute force	`find_discords_brute_force`(Y_t, m)	`jmotif`

Na poziomie implementacji wszystkie zostały napisane w języku `C++`, by dodatkowo przyśpieszyć czas ich działania, natomiast `R` wykorzystywany jest jako interfejs użytkownika. To pozwala na przeprowadzenie analizy porównawczej również jeśli chodzi o aspekt czasu potrzebnego na wykonywanie obliczeń (obliczenia zostały wykonane na procesorze Intel Core i5-4690K 3.5GHz).

Najważniejszą częścią porównania będzie ocena dokładności detekcji. W tym celu wyszczególnione zostaną początki trzech „najlepszych" sekwencji anomalnych dla każdego algorytmu oraz frakcja pokrycia z oryginalną sekwencją wyznaczoną przez eksperta. Uwzględnione zostaną różne szerokości okna zbliżone długością do średniej długości cykli występujących w danych. Dodatkowo dla algorytmu HOT-SAX zbadamy, jak parametry związane z dyskretyzacją sekwencji wpływają na skrócenie bądź wydłużenie czasu obliczeń.

4.4.3 Wykrywania anomalnych uderzeń serca w EKG

Dane, które rejestrują kardiolodzy podczas badań serca elektrokardiografem wydają się mieć idealną strukturę umożliwiającą zastosowanie algorytmów wykrywania sekwencji anomalnych. Na potrzeby naszej analizy, rozważamy dane EKG pewnego pacjenta mające postać szeregu czasowego o 15 tysiącach obserwacji, które są kolejnymi pomiarami napięcia. Rysunek 4.4 przedstawia strukturę badanego szeregu oraz wykrytej przez eksperta anomalii, która reprezentuje rytm przedsionkowy.



Rysunek 4.4: Pomiary napięcia badania elektrokardiogramem pewnego pacjenta; poniżej zaznaczona anomalia odpowiadająca rytmowi przedsionkowemu.

Zaznaczona na powyższym rysunku anomalia pojawia się u pacjenta w 10 774 pomiarze badania i trwa kolejne 100 pomiarów. Obserwowana część badania trwała około 3 minut 45 sekund, zatem jeden pomiar na siatce EKG pojawia się co około 0.015s. Liczba danych uzyskanych w tak krótkim czasie jest ogromna i ich analiza w czasie rzeczywistym przez eksperta (lekarza) jest mocno utrudniona.

Oko wprawnego kardiologa nie musi być nieomylne, tym bardziej jeśli mamy do czynienia z obszernymi zbiorami danych nawet o milionach pomiarów dla jednego pacjenta (np. gdy szukamy anomalii w celu potwierdzenia diagnozy). Dla istotnie dużych wolumenów danych cenne będzie wsparcie technologiczne, które, mogą zaoferować algorytmy takie jak HOT-SAX czy profil macierzowy.

W ramach analizy powyższego zbioru danych sprawdzona zostanie skuteczność detekcji anomalnych sekwencji na bazie algorytmu HOT-SAX dla szerokości okna 100, 125 oraz 150 (z racji na długość cykli EKG) oraz dla różnych wielkości parametrów α oraz ω, by ocenić jak ich dobór wpływa na czas detekcji (przypomnijmy, że dobór parametrów dyskretyzacji nie wpływa na wynik algorytmu).

Tabela 4.2: Zidentyfikowane najodleglejsze (względem swojego najbliższego sąsiada) sekwencje za pomocą algorytmu HOT-SAX

m	odległość	początek	pokrycie
100	**9.159**	**10753**	**0.800**
100	3.950	10857	0.180
100	3.460	14511	0.000
125	**9.481**	**10753**	**0.808**
125	5.094	12221	0.000
125	6.764	14467	0.000
150	**11.095**	**10792**	**0.553**
150	5.862	12511	0.000
150	7.399	14446	0.000

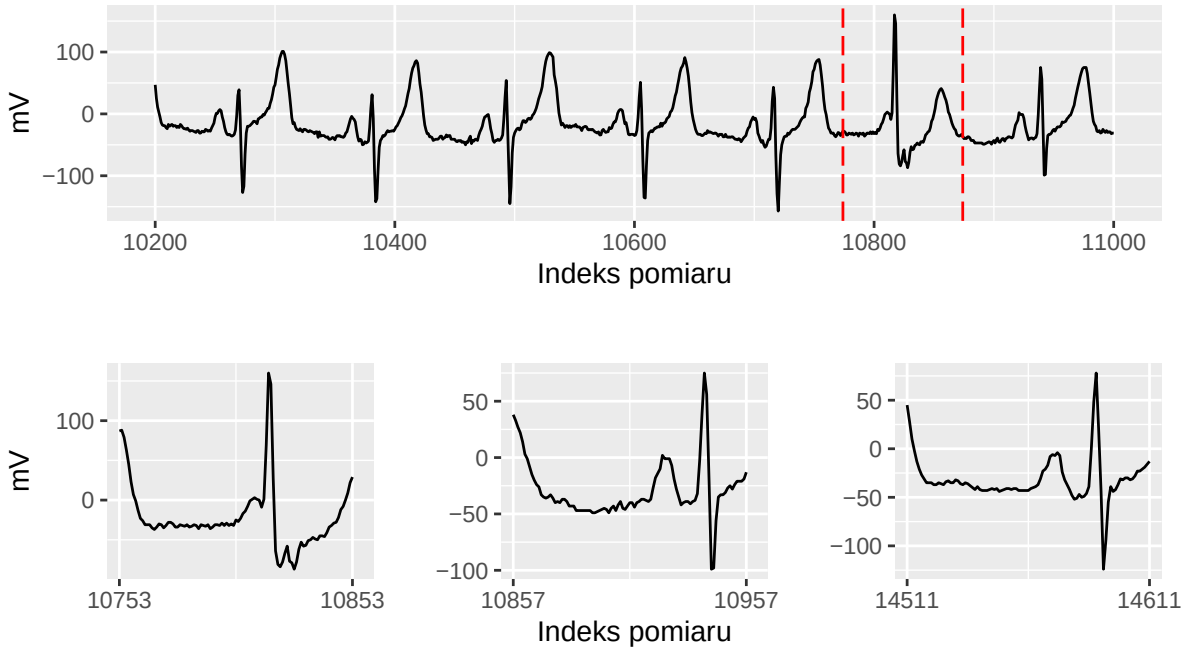
Tabela 4.3: Czasy działania algorytmu HOT-SAX dla różnych konfiguracji parametrów dyskretyzacji SAX

algorytm	czas $[s]$
HOTSAX(100, 3, 3)	3.39
HOTSAX(100, 5, 8)	**2.11**
HOTSAX(100, 8, 10)	3.97
HOTSAX(125, 3, 3)	4.8
HOTSAX(125, 6, 7)	**2.32**
HOTSAX(125, 12, 12)	6.88
HOTSAX(150, 3, 3)	11.46
HOTSAX(150, 6, 6)	**3.21**
HOTSAX(150, 10, 10)	9.78

Tabele 4.2 oraz 4.3 przedstawiają wyniki przeprowadzonej analizy. Co ciekawe, mimo iż określona przez eksperta anomalia ma długość 100 to dla $m = 125$ udało się znaleźć najbardziej zbliżoną sekwencję jeśli chodzi o stosunek pokrycia do długości. Dla $m = 100$ oraz $m = 125$ wyniki są dość zbliżone; został wykryty ten sam początek najbardziej odległej (względem swojego najbliższego sąsiada) sekwencji. Dla $m = 100$ również druga anomalna sekwencja znajduje się w pobliżu piku widocznego dla właściwej anomalii, natomiast dla większych szerokości okna jako drugie w kolejności anomalie zostały wykryte odleglejsze, w sensie indeksu pomiaru, sekwencje.

Jeśli chodzi o szybkość czasu detekcji to algorytm spisuje się bardzo dobrze. Dla parametrów dyskretyzacji z przedziału 5-8 osiągane są najlepsze wyniki rzędu 2-4s.

Z perspektywy wykorzystania rozważanych metod jako pomocy dla eksperta kluczowa będzie wizualizacja wyników algorytmu, które są zaprezentowane na rysunku 4.5.



Rysunek 4.5: Wyniki detekcji algorytmu HOT-SAX dla $m = 100$ – trzy kolejne zidentyfikowane anomalne sekwencje.

Algorytm poprawnie wykrył anomalny pik, natomiast wykryta przez HOT-SAX sekwencja zaczyna się odrobinę wcześniej. Ze względu na to, że algorytm kieruje się jedynie maksymalizacją odległości do najbliższego sąsiada dla każdej sekwencji, nie był on w stanie rozpoznać czy uwzględnić regularności i struktury poprzednich cykli w finalnym rezultacie. Kolejne dwie zidentyfikowane sekwencje są już dużo bardziej podobne względem siebie niż względem pierwszej w kolejności zidentyfikowanej anomalnej sekwencji.

Kolejnym krokiem analizy będzie zastosowanie profilu macierzowego do detekcji najmniej podobnych do reszty sekwencji – maksymalizujących wartość profilu dla danej sekwencji (tabela 4.4).

Tabela 4.4: Zidentyfikowane najodleglejsze sekwencje za pomocą profilu macierzowego

m	profil macierzowy	początek	pokrycie
100	**9.206**	**10754**	**0.810**
100	7.246	10805	0.700
100	3.970	10858	0.170
125	**9.519**	**10754**	**0.808**
125	7.976	10818	0.456
125	4.779	12537	0.000
150	**7.432**	10793	0.547
150	2.242	**10717**	**0.627**
150	1.555	12512	0.000

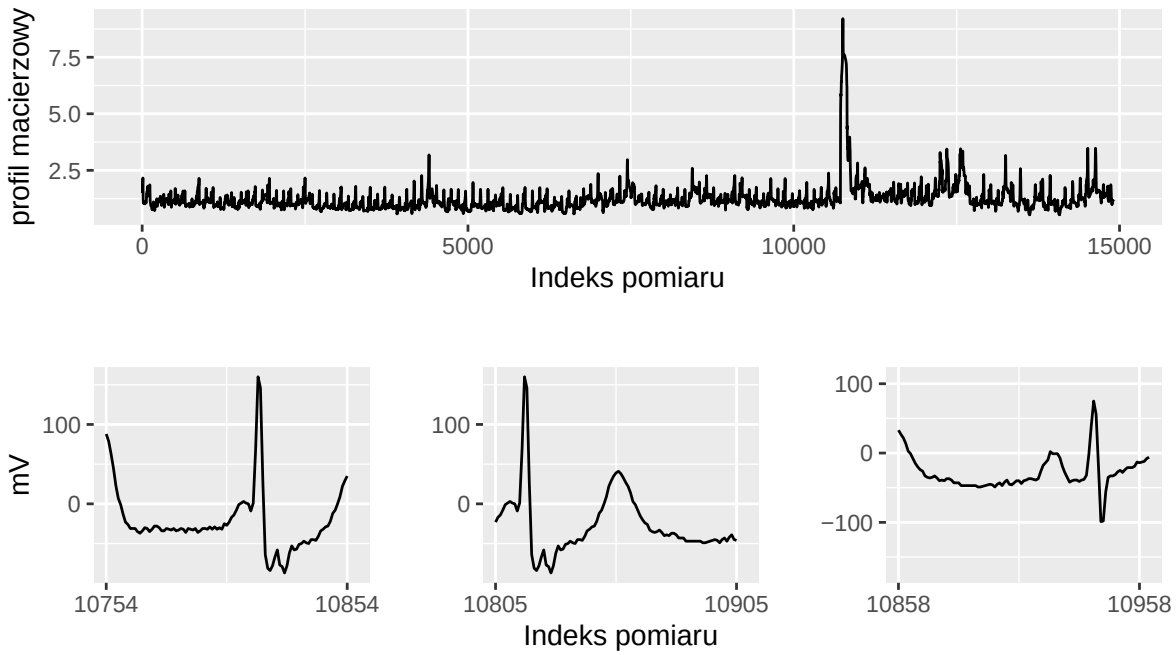
Podobnie jak poprzednio, wybrana została bardzo podobna sekwencja przesunięta

zaledwie o jedną obserwacje względem zidentyfikowanej przy pomocy algorytmu HOT-SAX. Dla $m = 100$ widać, że strefa wykluczenia dla dopasowań trywialnych to 50 obserwacji z uwagi na regularne interwały między początkami sekwencji; pozostałe dwie zidentyfikowane sekwencje, rozpoczynające się od obserwacji 10805 oraz 10858, znajdują się w pobliżu prawdziwej anomalii. Dla innych szerokości okna algorytm uznał za istotne różną od reszty sekwencje rozpoczynającą się w okolicach 12500 obserwacji, co również można było zaobserwować dla algorytmu HOT-SAX.

Tabela 4.5: Czasy obliczeń profilu macierzowego dla różnych szerokości okna

algorytm	czas $[s]$
MP(100)	22.53
MP(125)	21.39
MP(150)	**19.37**

Jeśli chodzi o czas obliczeń, widać ciekawą zależność, gdyż dla większych szerokości okna czas okazał się nieco krótszy. Do obliczania profilu macierzowego wykorzystany został algorytm *STOMP*. W przeciwieństwie do HOT-SAX profil macierzowy jest strukturą danych wyliczaną bez wykorzystywania reguł pozwalających na pominięcie części obserwacji czy reguł heurystycznych, by uzyskać profil dla wszystkich $n-m+1$ sekwencji. To też czyni go odrobinę wolniejszym, ale dającym o wiele więcej możliwości w badaniu szerokiej gamy zagadnień pozyskiwania wiedzy. Na rysunku 4.6 zilustrowany został profil macierzowy z trzema „najlepszymi" sekwencjami anomalnymi.



Rysunek 4.6: Profil macierzowy badanego szeregu oraz 3 sekwencje o najwyższych jego wartościach dla $m = 100$.

Profil macierzowy wskazuje na zdecydowanie bardziej nietypowe sekwencje z charakte-

rystycznym pikiem. Zidentyfikowane podciągi są tak naprawdę kolejnymi przesuniętymi fragmentami anomalii wykrytej przez eksperta, także profil wskazał nawet trzykrotnie ten obszar, jako w tym przypadku potencjalny okres anomalnej pracy serca.

Na koniec warto odnieść uzyskane wyniki do podejścia siłowego opartego na algorytmie *Brute force* (tabela 4.6).

Tabela 4.6: Zidentyfikowane najodleglejsze sekwencje za pomocą algorytmu *Brute force*

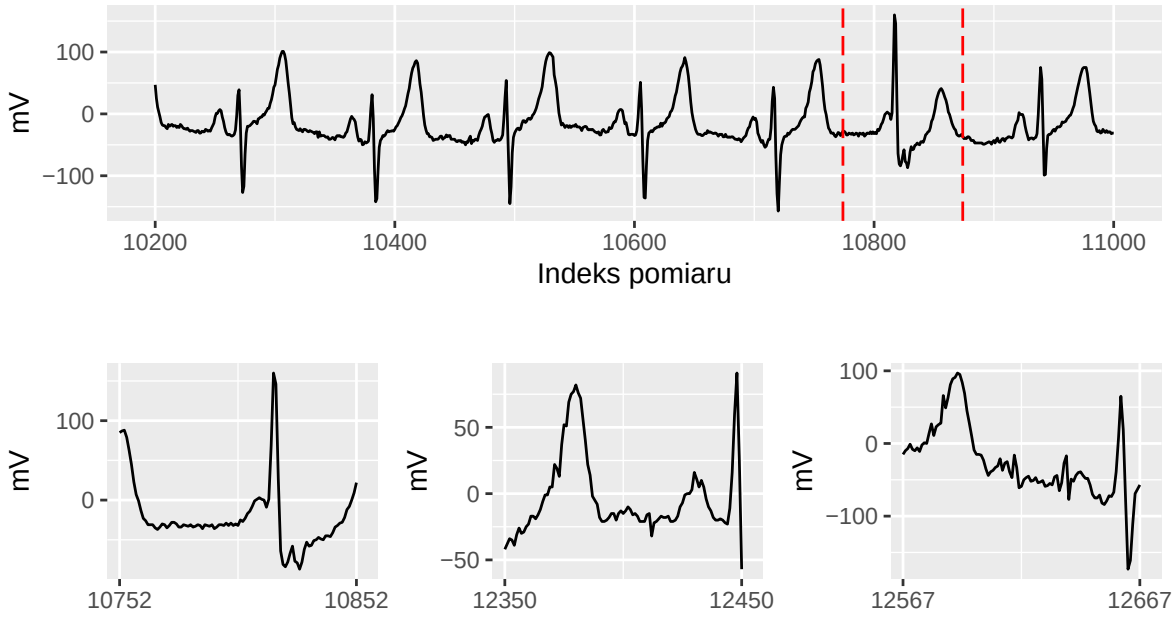
m	odległość	początek	pokrycie
100	**8.711**	**10752**	**0.790**
100	4.179	12350	0.000
100	4.092	12567	0.000
125	**9.294**	**10815**	**0.480**
125	6.521	12338	0.000
125	7.376	14468	0.000
150	**10.314**	**10796**	**0.527**
150	6.697	12339	0.000
150	7.688	14455	0.000

Co ciekawe, teoretycznie najbardziej dokładne (w sensie maksymalizowania odległości euklidesowej) rozwiązanie okazało się w gorszym stopniu pokrywać z oceną eksperta niż w przypadku profilu macierzowego czy algorytmu HOT-SAX. Jest to jak najbardziej możliwe, gdyż żadna z przedstawionych metod nie uwzględnia długości cyklu uderzeń serca, którą z pewnością sugerował się ekspert przy wyborze początku sekwencji. Drugie i trzecie najodleglejsze od swojego najbliższego sąsiada sekwencje w żadnym stopniu nie pokrywają się z oryginalną sekwencją anomalną dla każdej badanej szerokości okna m. Tabela 4.7 pokazuje natomiast największy problem tej metody czyli czas obliczeń.

Tabela 4.7: Czasy obliczeń algorytmu *Brute force* dla różnych szerokości okna

algorytm	czas $[min]$
BF(100)	**9.09**
BF(125)	9.82
BF(150)	11.4

Czas detekcji algorytmu *Brute force* zdecydowanie zależy od długości okna, co potwierdza złożoność algorytmu $O(mn^2)$. Można by pomyśleć, że około 10 minut to jeszcze czas, który można zaakceptować, natomiast jest to prawie 250 razy więcej niż w przypadku algorytmu HOT-SAX, a gdy 15000 obserwacji zamieni się na na setki tysięcy czy miliony (jak przy danych sejsmologicznych) zastosowanie tego algorytmu staje się niemożliwe.



Rysunek 4.7: Wyniki detekcji na bazie algorytmu *Brute force* dla $m = 100$ — trzy kolejne zidentyfikowane anomalne sekwencje.

Rysunek 4.7 przedstawia wyniki detekcji dla $m = 100$ na podstawie algorytmu *Brute force*. Trzy zestawione sekwencje wydają się być wizualnie odbiegające od siebie, natomiast wykres „najlepszej" anomalnej sekwencji potwierdza, że HOT-SAX oraz profil macierzowy spisały się bardzo dobrze jeśli chodzi o balans pomiędzy optymalizacją czasu obliczeń względem dokładności.

4.4.4 Podsumowanie analizy

Każdemu z trzech badanych algorytmów udało się wskazać odpowiedni obszar, który dla każdej rozważanej szerokości okna m przynajmniej w 48% pokrywał się z oznaczonym przez eksperta rytmem przedsionkowym. Analizując czas obliczeń, bezkonkurencyjny okazał się HOT-SAX, który w kilka sekund poradził sobie z obliczeniami; do wyliczenia profilu macierzowego potrzeba już kilkudziesięciu sekund, co jednak w dalszym ciągu jest istotnie lepszym wynikiem w porównaniu do algorytmu *Brute force*, który pod tym względem wypada fatalnie szczególnie gdy mowa o dużych zbiorach danych.

Jeśli chodzi o kompromis pomiędzy dokładnością a czasem detekcji HOT-SAX plasuję się nieco przed profilem macierzowym, który oferuje jeszcze dodatkowe możliwości w zakresie analizy eksploracyjnej badanego szeregu czasowego. Algorytm *Brute force* mimo teoretycznie dokładnego wyniku charakteryzuje się zbyt dużą złożoność obliczeniową, by móc go rozważać w praktycznych zastosowaniach dla obszerniejszych zbiorów danych, które są docelowym obiektem badań związanych z wykrywaniem anomalnych sekwencji.

Podsumowanie

Szeroki zakres przedstawionych metod oraz analiz wykrywania oraz modelowania obserwacji odstających różnego rodzaju wpłynął na mnogość wniosków dotyczących zarówno efektywności czy konstrukcji przedstawionych algorytmów, jak i pomysłów na kolejne badania.

Uwzględnienie na etapie modelowania występujących w danych efektów obserwacji odstających w postaci interwencji (przedstawionych w Rozdziale 2) okazało się kluczowe dla poprawy dokładności prognoz wyznaczonych dla emisji dwutlenku węgla w Polsce. Klasyczne modele ARIMA, metoda naiwna oraz modele wygładzania wykładniczego poradziły sobie o wiele gorzej z prognozowaniem badanego szeregu niż model SARIMAX, wykorzystujący dodatkowe informacje w postaci zmiennych utworzonych na podstawie zidentyfikowanych obserwacji odstających.

W części pracy poświęconej zmianom impulsowym (Rozdział 3) przeprowadzona została szczegółowa analiza porównawcza algorytmów wykrywania obserwacji odstających o charakterze punktowym. Rozważane metody uzyskały szerokie spektrum wyników jeśli chodzi o ich dokładność oraz czas detekcji, w zależności od specyfiki badanej grupy szeregów. Niemniej algorytmami, które zachowały najlepszy balans między czasem działania, a dokładnością detekcji okazały się autorskie metody `OSWM` oraz `TSWM`. Szczegółowe wnioski dotyczące przeprowadzonej analizy porównawczej są zamieszczone w podrozdziale 3.5.6.

Ważnym z punktu widzenia zastosowań w różnych dziedzinach nauki i techniki jest wykrywanie niepożądanych sekwencji bądź ich fragmentów wskazujących na potencjalne zaburzenia występujące w danych. Omówione w Rozdziale 4 techniki wykrywania anomalnych podciągów pozwalają istotnie przyśpieszyć poszukiwania najbardziej nietypowych sekwencji. Jest to szczególnie istotne dla danych o dużej liczbie obserwacji, co ilustruje studium przypadku poświęcone detekcji anomalnych sekwencji w EKG (podrozdział 4.4). Metody wykorzystujące reguły bądź algorytmy przyśpieszające obliczenia nie tracą wiele względem metody dokładnej, redukując zauważalnie czas jej działania, co z pewnością będzie istotne dla praktyka analizującego wyniki badań.

Możliwym kierunkiem dalszych badań jest opracowanie odpowiednich algorytmów głębokiego uczenia, w momencie gdy dysponujemy dużymi zbiorami danych. Niemniej wnioski wyciągnięte na podstawie analizy porównawczej w podrozdziale 3.5 sugerują, że proste algorytmy oparte na dekompozycji oraz odpowiednich regułach decyzyjnych są w stanie konkurować z bardziej zaawansowanymi technikami, zatem celem dalszych badań może być również skonstruowanie prostej, w miarę możliwości automatycznej, metody wykrywania anomalii punktowych, która przy tym będzie cechować się wysoką skutecznością. Alternatywnym podejściem zaproponowanym w podrozdziale 3.4 jest konstrukcja odpowiednich zmiennych, które mają dobre własności dyskryminacyjne pod kątem wykrywania anomalii, na podstawie badanego szeregu czasowego. Kluczowymi czynnikami prowadzącymi do poprawy dokładności detekcji w tym przypadku będzie odpowiedni dobór zmiennych oraz algorytmów wykrywania anomalii dla danych wielowymiarowych.

Dodatek: Wykorzystane oprogramowanie

Do przeprowadzonych analiz oraz wizualizacji wyników zamieszczonych w pracy zostały wykorzystane rozmaite biblioteki pakietu `R`. Tabela 4.8 przedstawia najważniejsze wykorzystane biblioteki uwzględniając podział względem ich funkcjonalności.

Tabela 4.8: Najważniejsze biblioteki pakietu `R` wykorzystane w niniejszej pracy.

biblioteka	cytowanie
Wykrywanie punktowych anoamlii	
`forecast`	[22]
`tsoutliers`	[15]
`tsrobprep`	[29]
`AnomalyDetection`	[36]
Wykrywanie anomalnych sekwencji	
`jmotif`	[33]
`tsmp`	[5]
Tabele i rysunki	
`ggplot2`	[37]
`patchwork`	[30]
`kableExtra`	[40]

Do przeprowadzenia studium przypadku zamieszczonego w podrozdziale 2.4 wykorzystany został pakiet `forecast`, w szczególności funkcje `auto.arima` – automatyczny wybór optymalnego modelu ARIMA, `forecast` oraz `accuracy` do konstrukcji oraz porównania dokładności prognoz oraz pakiet `tsoutliers`, w którym znajduje się funkcja `tso` będącą implementacją algorytmu połączonej estymacji parametrów modelu ARIMA oraz efektów obserwacji odstających autorstwa Chena i Liu (patrz podrozdział 2.3.2).

W analizie porównawczej dokładności wykrywania anomalii punktowych (patrz podrozdział 3.5) oprócz metod zaimplementowanych w bibliotekach zamieszczonych w tabeli 4.8 zostały zaproponowane autorskie algorytmy `OSWM` oraz `TSWM` (patrz podrozdział 3.1.2), których dokumentacja znajduje się poniżej.

`One-sided window method`

Description

Simple function which detects point anomalies in univariate time series data. The procedure uses smoothing function via sliding window to compute predicted values of observations which are then compared to their actual values. Observation is labeled as anomalous if absolute value of the difference of actual and predicted values exceeds threshold defined by thershold function and alpha coefficent.

Usage

```
OSWM(
    ts,
    m = 10,
    alpha = 3,
    t.func = sd,
    p.func = mean,
    treshold = NULL,
    decomp = F,
    ...
)
```

Arguments

`ts`	an univariate numeric time series object or a numeric vector.
`m`	size of a sliding window.
`alpha`	parameter defining the multiple of `t.func` in a threshold computation.
`t.func`	function used for threshold computation.
`p.func`	function applied via sliding window.
`threshold`	if not `NULL` the provided threshold value is used instead of `alpha`*`t.func`.
`decomp`	if `TRUE` the reminder of the mstl decomposition performed on ts will be used.
`...`	additional arguments passed to mstl.

Examples

```
## Creating toy dataset
data <- rnorm(100)
ind <- sample(11:100, 3)
data[ind] <- rep(5, 3) #adding anomalies

## Detection
OSWM(data)$anom.vec #1 if anomalous
OSWM(data)$times #indices of anomalous points
```

`Two-sided window method`

Description

Simple function which detects point anomalies in univariate time series data. The procedure applies smoothing function on neighbouring observations to compute estimated values of observations which are then compared to their actual values. Observation is labeled as anomalous if absolute value of the difference of actual and estimated values exceeds threshold defined by thershold function and alpha coefficent.

Usage

```
TSWM(
    ts,
    k = 5,
    alpha = 3,
    t.func = sd,
    e.func = mean,
    treshold = NULL,
    decomp = F,
    ...
)
```

Arguments

`ts`	an univariate numeric time series object or a numeric vector.
`m`	number of proceeding and exceeding values given to .
`alpha`	parameter defining the multiple of `t.func` in a threshold computation.
`t.func`	function used for threshold computation.
`e.func`	function applied via sliding window.
`threshold`	if not `NULL` the provided threshold value is used instead of `alpha`*`t.func`.
`decomp`	if `TRUE` the reminder of the mstl decomposition performed on ts will be used.
`...`	additional arguments passed to mstl.

Examples

```
## Creating toy dataset
data <- rnorm(100)
ind <- sample(11:100, 3)
data[ind] <- rep(5, 3) #adding anomalies

## Detection
TSWM(data)$anom.vec #1 if anomalous
TSWM(data)$times #indices of anomalous points
```

Bibliografia

[1] Yahoo! webscope dataset ydata-labeled-time-series-anomalies-v1_0. `http://labs.yahoo.com/Academic_Relations`. Accessed: 2022-03-28.

[2] Ahmad, S., Lavin, A., Purdy, S., Agha, Z. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing 262* (Nov. 2017), 134–147.

[3] Bandara, K., Hyndman, R. J., Bergmeir, C. MSTL: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns, 2021.

[4] Basu, S., Meckesheimer, M. Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems 11*, 2 (Aug. 2006), 137–154.

[5] Bischoff, F. *tsmp: Time Series with Matrix Profile*, 2020. R package version 0.4.14.

[6] Blázquez-García, A., Conde, A., Mori, U., Lozano, J. A. A review on outlier/anomaly detection in time series data, 2020.

[7] Box, G. E. P., Tiao, G. C. Intervention analysis with applications to economic and environmental problems. *Journal of the American Statistical Association 70*, 349 (Mar. 1975), 70–79.

[8] Braei, M., Wagner, S. Anomaly detection in univariate time-series: A survey on the state-of-the-art, 2020.

[9] Brockwell, P., Davis, R. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer New York, 2006.

[10] Chang, I., Tiao, G. C., Chen, C. Estimation of time series parameters in the presence of outliers. *Technometrics 30*, 2 (may 1988), 193–204.

[11] Chen, C., Liu, L.-M. Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association 88*, 421 (Mar. 1993), 284.

[12] Cheung, Y.-W., Lai, K. S. Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics 13*, 3 (1995), 277–280.

[13] Cleveland, R. B., Cleveland, W. S., McRae, J. E., Terpenning, I. Stl: A seasonal-trend decomposition procedure based on loess (with discussion). *Journal of Official Statistics 6* (1990), 3–73.

[14] Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association 74*, 368 (Dec. 1979), 829–836.

[15] de Lacalle, J. L. *tsoutliers: Detection of Outliers in Time Series*, 2019. R package version 0.6-8.

[16] Fraley, C., Raftery, A. E. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association 97*, 458 (June 2002), 611–631.

[17] Friedman, J. H. *SMART User's Guide*, vol. Technical Report No. 1. Stanford, Laboratory for Computational Statistics, Stanford University, 1984.

[18] Grubbs, F. E. Sample criteria for testing outlying observations. *The Annals of Mathematical Statistics 21*, 1 (Mar. 1950), 27–58.

[19] Guerrero, V. M. Time-series analysis supported by power transformations. *Journal of Forecasting 12*, 1 (Jan. 1993), 37–48.

[20] Hannah Ritchie, M. R., Rosado, P. co2.

[21] Hochenbaum, J., Vallis, O., Kejariwal, A. Automatic anomaly detection in the cloud via statistical learning. *ArXiv abs/1704.07706* (2017).

[22] Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., Yasmeen, F. *forecast: Forecasting functions for time series and linear models*, 2022. R package version 8.16.

[23] Hyndman, R. J., Khandakar, Y. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software 27*, 3 (2008), 1–22.

[24] Jager, F., Taddei, A., Moody, G. B., Emdin, M., Antolic, G., Dorn, R., Smrdel, A., Marchesi, C., Mark, R. G. The long-term st database, 1995.

[25] Kaiser, R., Maravall, A. Seasonal outliers in time series. *Estadistica 53* (02 1999).

[26] Keogh, E., Lin, J., Fu, A. HOT SAX: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, IEEE.

[27] Lin, J., Keogh, E., Wei, L., Lonardi, S. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery 15*, 2 (Apr. 2007), 107–144.

[28] Mueen, A., Zhu, Y., Yeh, M., Kamgar, K., Viswanathan, K., Gupta, C., Keogh, E. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance, August 2017. `http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html`.

[29] Narajewski, M., Kley-Holsteg, J., Ziel, F. tsrobprep — an R package for robust preprocessing of time series data. *SoftwareX 16* (2021), 100809.

[30] Pedersen, T. L. *patchwork: The Composer of Plots*, 2020. R package version 1.1.1.

[31] Rosner, B. Percentage points for a generalized ESD many-outlier procedure. *Technometrics 25*, 2 (may 1983), 165–172.

[32] Scrucca, L., Fop, M., Murphy, T. B., Raftery, A. E. mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal 8*, 1 (2016), 289–317.

[33] Senin, P. *jmotif: Time Series Analysis Toolkit Based on Symbolic Aggregate Discretization, i.e. SAX*, 2020. R package version 1.1.1.

[34] Tietjen, G. L., Moore, R. H. Some grubbs-type statistics for the detection of several outliers. *Technometrics 14*, 3 (Aug. 1972), 583–597.

[35] Tukey, J. W. *Exploratory Data Analysis*. Mass: Addison-Wesley Pub. Co, Reading, 1977.

[36] Vallis, O., Hochenbaum, J., Kejariwal, A., Rudis, B., Tang, Y. *Anomaly-Detection: Anomaly Detection Using Seasonal Hybrid Extreme Studentized Deviate Test*, 2018. R package version 2.0.1.

[37] Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

[38] Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Silva, D. F., Mueen, A., Keogh, E. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)* (2016), pp. 1317–1322.

[39] Zagdański, A., Suchwałko, A. *Analiza i prognozowanie szeregów czasowych. Praktyczne wprowadzenie na podstawie środowiska R*. PWN, 2015. ISBN 978-83-01-18356-1.

[40] Zhu, H. *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*, 2021. R package version 1.3.4.

[41] Zhu, Y., Yeh, C.-C. M., Zimmerman, Z., Kamgar, K., Keogh, E. Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds. In *2018 IEEE International Conference on Data Mining (ICDM)* (Nov. 2018), IEEE.

[42] Zhu, Y., Zimmerman, Z., Senobari, N. S., Yeh, C.-C. M., Funning, G., Mueen, A., Brisk, P., Keogh, E. Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins. In *2016 IEEE 16th International Conference on Data Mining (ICDM)* (2016), pp. 739–748.