Business Intelligence Case Challenge

Piotr Migdałek

Politechnika Wrocławska

Maj 2022

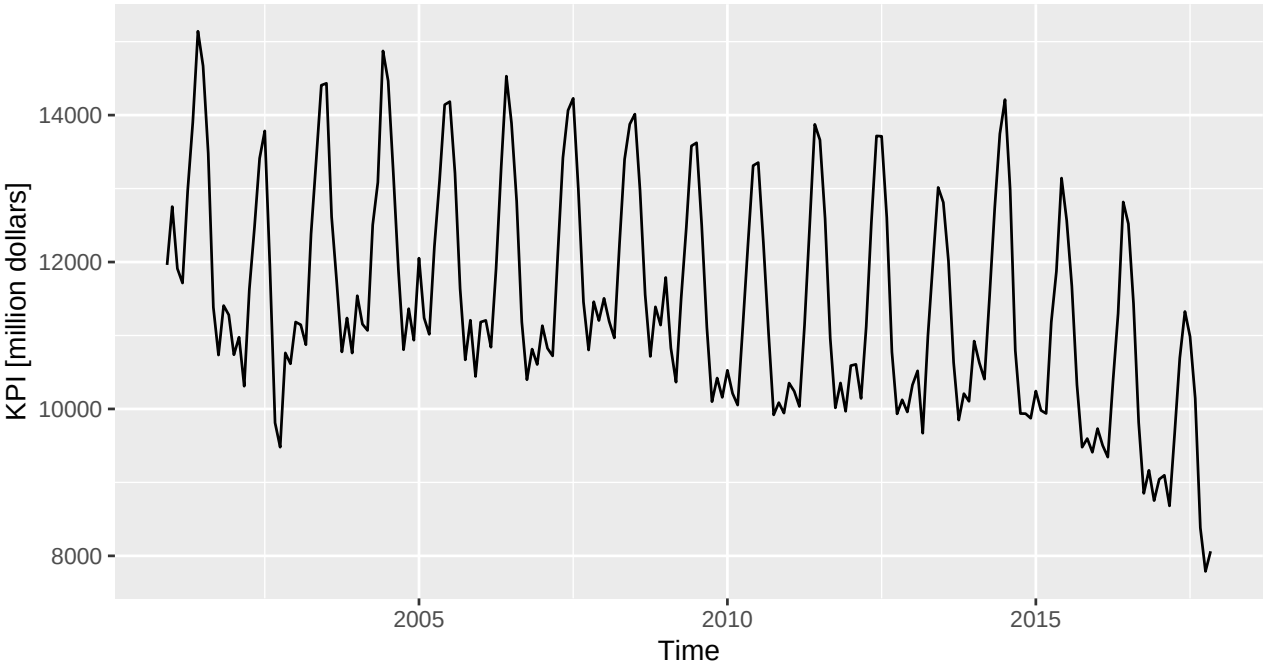
Spis treści

1 Wstęp

Wyzwanie zaproponowane przez Koło Naukowe Analizy Danych działające na Uniwersytecie Ekonomicznym w Krakowie dotyczy prognozowania zysku ze sprzedaży detalicznej prądu. Podczas przeprowadzanej analizy będę korzystał z pakietu `R`, dokładniej z biblioteki `forecast` oraz `prophet`, służących do analizy szeregów czasowych, `ggplot2` do wizualizacji kolejnych etapów analizy oraz `knitr` do sporządzenia raportu w formacie pdf z wykrozystaniem oprogramowania do zautomatyzowanego składu tekstu – LaTeX.

Na sam początek wczytam dane z pliku csv oraz zwizualizuję ich postać, by móc lepiej zbadać ich specyfikę pod kątem dopasowania odpowiednich modeli statystycznych.

```
BICC <- read.csv("Data_BICC.csv", header = T, row.names=NULL, sep = ";")[,2]
ts.BICC <- ts(BICC, frequency = 12, start = c(2000, 12))
```

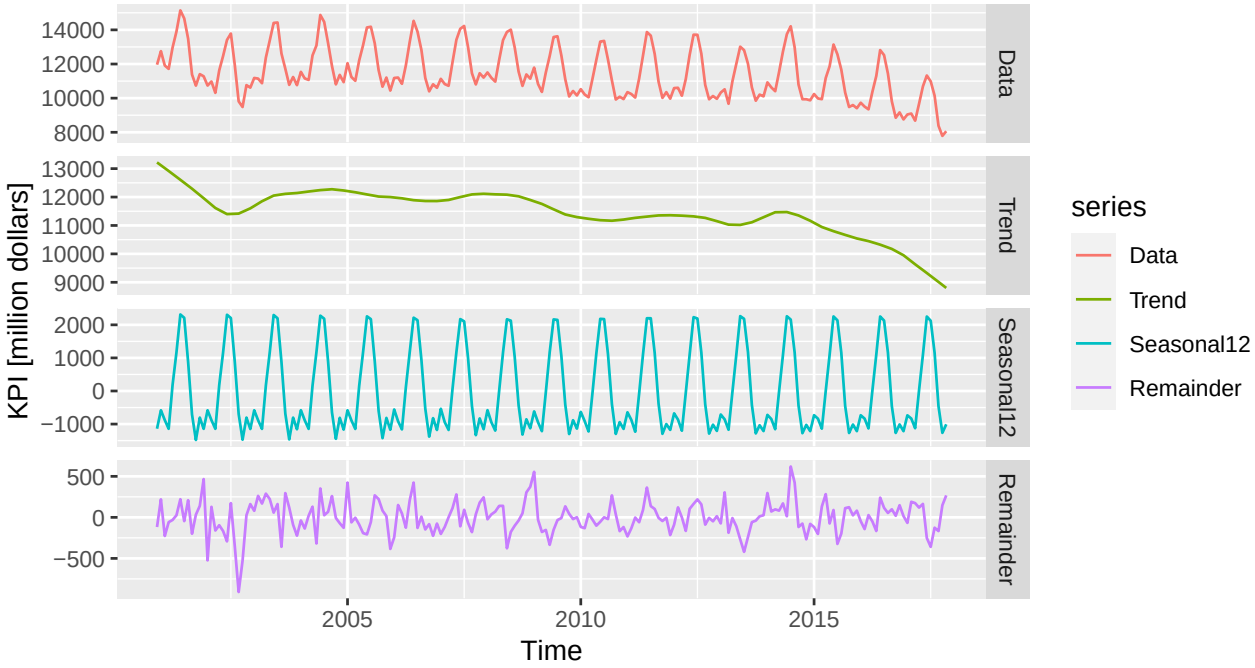


Rysunek 1: Analizowany szereg czasowy zysku ze sprzedaży detalicznej prądu

Dane charakteryzują się wyraźną sezonowością o okresie 12 miesięcy, której modelowanie będzie kluczowym aspektem prognozowania. W danych występuję również trend i ma on charakter nieliniowy, gdyż po spadku w początkowych latach później otrzymujemy niewielki wzrost, następnie znowu spadek, po którym następuję mały wzrost, którego następstwem jest znacząca tendencja spadkowa.

By lepiej przyjrzeć się poszczególnym komponetnom szeregu można zastosować technikę dekompozycji jak np. dekompozycję STL [1], która rozbija postać szeregu czasowego na sumę składników: estymowany trend, estymowana sezonowość oraz stacjonarne zakłócenie.

```
ts.BICC %>% mstl() %>% autoplot(ts.BICC) + labs(y = "KPI [million dollars]")
```



Rysunek 2: Rozbicie szeregu na poszczególne komponentu używając dekompozycji STL

Powyższy wykres potwierdza wnioski wysnute z pierwotnej postaci danych, czyli 12 miesięczną sezonowość oraz nieliniowy trend o charakterze malejącym.

2 Przedstawienie metodologii

Do wyznaczenia prognoz dla analizowanego szeregu czasowego użyję kombinacji prognoz wyznaczonych przez kilka modeli. W przełomowym artykule, Bates i Granger [2] ponad 50 lat temu wykazali, że uśrednienie prognoz często prowadzi do poprawy ich dokładności. Wielu praktyków analizowało ważone średnie i bardziej skomplikowane metody uzwględniania kilku wariantów prognoz jednocześnie, w praktyce zwykła średnia okazuje się najczęściej najlepszym rozwiązaniem.

2.1 Dynamiczna harmoniczna regresja

Klasycznym podejściem w modelowaniu szeregów czasowym jest modelowanie ARIMA zaproponowane przez Boxa i Jenkinsa prawie pół wieku temu. Modele ARIMA (*AutoRegressive Integrated Moving Average*) składają się z części autoregresyjnej, która jest poniekąd regresją liniową dla opóźnionych wartości szeregu oraz części ruchomej średniej, która w ten sam sposób modeluje opóźnione reszty modelu. Człon *Integrated* odpowiada za sprowadzenie szeregu do postaci stacjonarnej, gdyż modele ARMA są modelami stacjonarnymi. Postać stacjonarną uzyskuje się przez operację różnicowania, która pozbywa się pierwiastka jednostkowego z wyjściowych danych poprzez odjęcie od każdej obserwacji w chwili t jej poprzednika w chwili $t-1$. Do wybrania optymalnego rzędu różnicowania (czasami operację trzeba powtórzyć kilka razy, by uzyskać postać stacjonarną szeregu) wykrozystuję się rozmaite testy statycztyczne np. test KPSS [3]. Do dobrania odpowiednich rzędów wielomianów autoregresyjnego oraz ruchomej

średniej można użyć procedury iteracyjnej dobierającej odpowiedni model, minimalizując wybrane kryterium informacyjne (AIC, AICc lub BIC) [4] w tym celu można wykorzystać funkcję `auto.arima`. Model ARIMA może również mieć analogiczną strukturę do modelowania części sezonowej, jeśli takowa występuję w analizowanych danych.

Zwykłe modele sezonowe SARIMA(p,d,q)(P,D,Q)[12] gorzej poradzą sobie z analizowanym zadaniem predykcyjnym z racji na swój liniowy charakter. W drodze modyfikacji klasycznego podejścia warto roważyć model dynamicznej harmonicznej regresji, która używa komponentów fourierowskich ($\sin\frac{2\pi tk}{12}$ oraz $\cos\frac{2\pi tk}{12}$) jako *dummy variables* do modelowania sezonowości, natomiast inne dynamiki danych wychwytywane są przez reszty ARIMA.

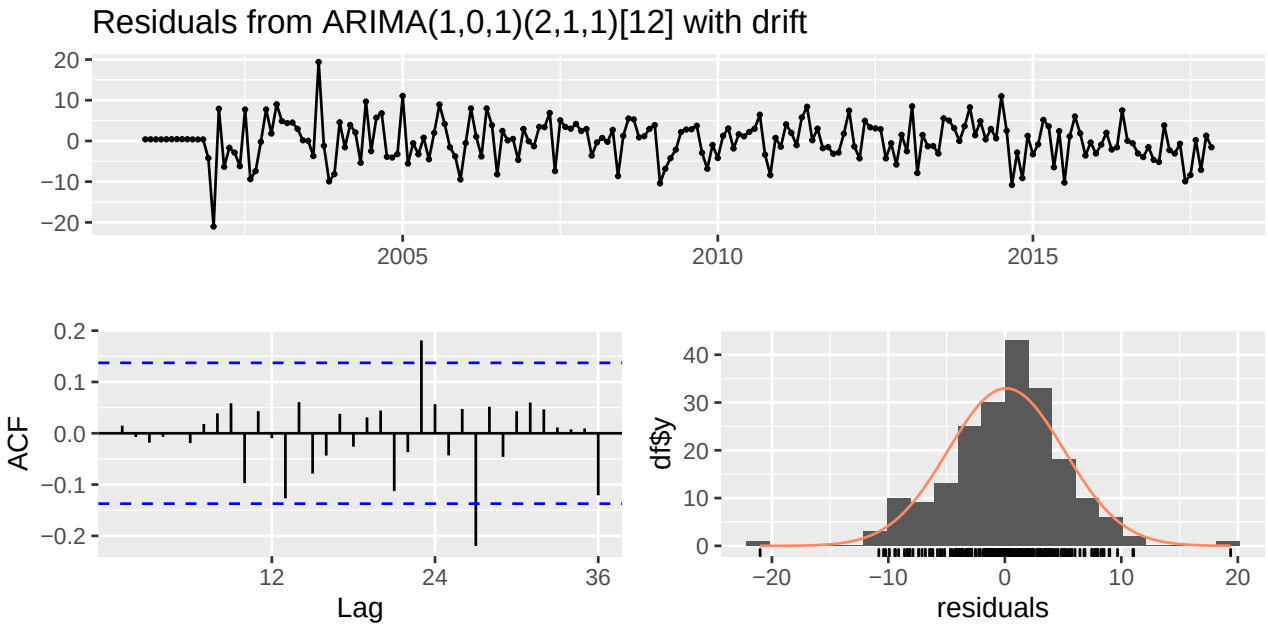
```
#domyślnie kryterium informacyjne to AICc;
#test stacjonarności to domyślnie KPSS
arima <- auto.arima(ts.BICC, lambda = "auto")
ARIMA <- arima %>% forecast(h =50)

dhr <- auto.arima(
  ts.BICC,
  lambda = "auto",
  xreg = fourier(ts.BICC, 6),
  seasonal = F
)
DHR <- dhr %>%
  forecast(h = 50, xreg = fourier(ts.BICC, 6, h = 50))
```

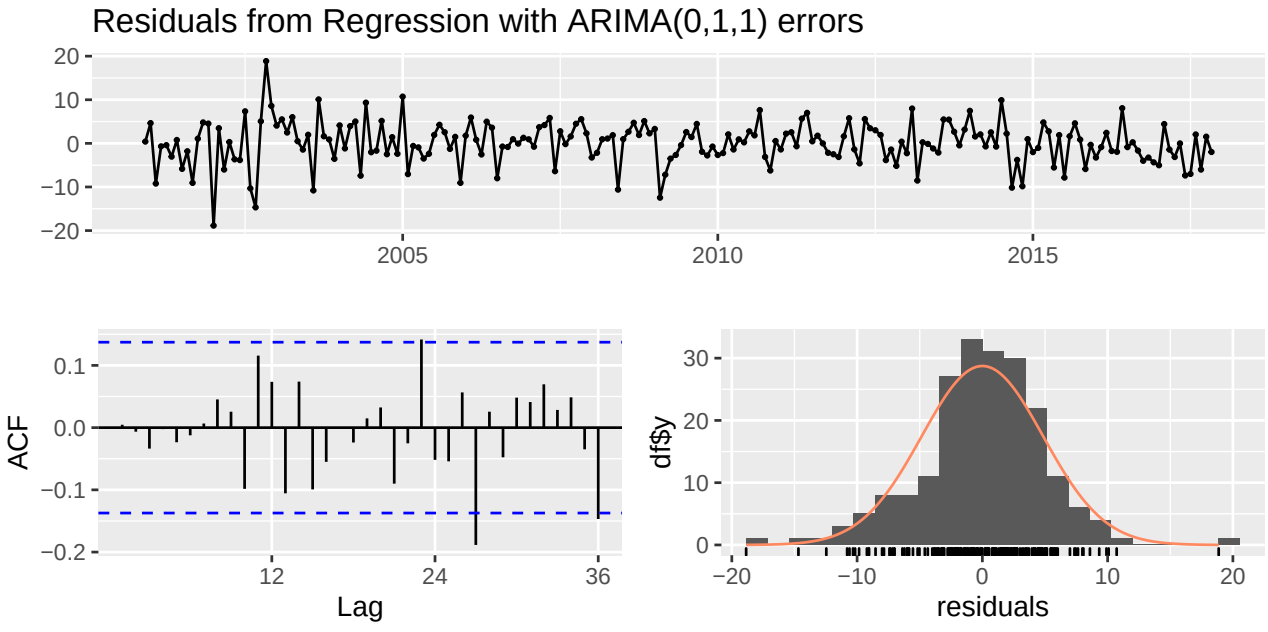
```
## Series: ts.BICC
## ARIMA(1,0,1)(2,1,1)[12] with drift
## Box Cox transformation: lambda= 0.5763918
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1    drift
##       0.9557  -0.2934  0.0969  -0.0469  -0.8696  -0.3118
## s.e.  0.0308   0.0773  0.1044   0.1045   0.1013   0.1095
##
## sigma^2 estimated as 27.17:  log likelihood=-594.53
## AIC=1203.07   AICc=1203.68   BIC=1225.87
## Series: ts.BICC
## Regression with ARIMA(0,1,1) errors
## Box Cox transformation: lambda= 0.5763918
##
## Coefficients:
##           ma1    drift     S1-12     C1-12    S2-12   C2-12   S3-12  C3-12
##       -0.3225  -0.4135  -18.3252  -23.0845  15.0591  0.1167  0.0294  0.208
## s.e.   0.0682   0.2315    0.6914    0.6863   0.4269  0.4257  0.3580  0.358
##         S4-12   C4-12   S5-12   C5-12   C6-12
##       -2.3400  2.0251  0.2302  3.4323  0.1927
## s.e.   0.3318  0.3323  0.3210  0.3217  0.2251
##
```

```
## sigma^2 estimated as 25.19:  log likelihood=-608.87
## AIC=1245.74   AICc=1247.97   BIC=1292.12
```

Korzystając z funkcji `auto.arima` do danych zostały dopasowane modele ARIMA(1,0,1)(2,1,1)[12] oraz ARIMA(0,1,1) + składowe fourierowskie. Przed modelowaniem wykonana została transformacja potęgowa Boxa-Coxa rzędu $\lambda = 0.5763918$. Natomiast dla modelu regresji ustalony został parametr $K = 6$, regulujący liczbę regresorów, po to by dostać mniej wygładzoną, bardziej zniuansowaną postać sezonowości. Poniżej znajdują się wykresy reszt wyznaczonych modeli.

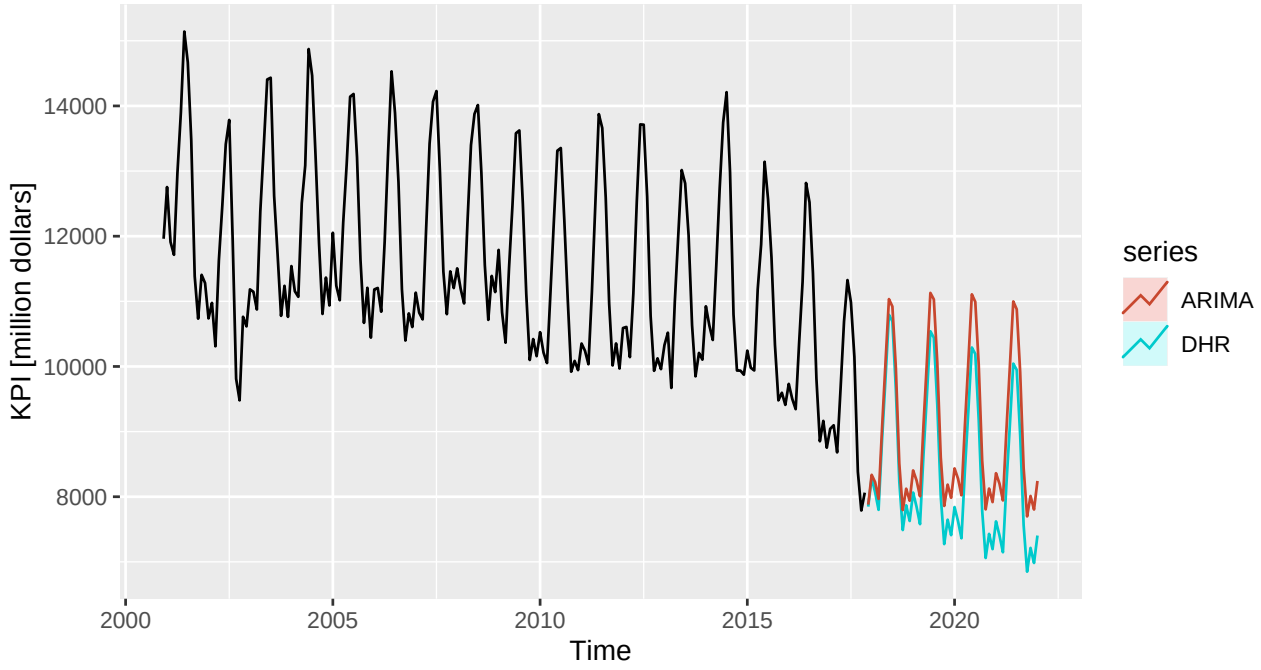


Rysunek 3: Reszty dla modelu ARIMA



Rysunek 4: Reszty dla modelu DHR

Reszty modeli również sugerują, że minimalnie lepiej dopasowany jest model z komponentami fourierowskimi, natomiast obydwu modelom nie udało się uzwględnić pełnej informacji w danych, co widać na wykresie funkcji ACF (wartości dla kilku opóźnień przekraczają poziom istotności dla białego szumu). Poniższy wykres przedstawia wyznaczone prognozy za pomocą dopasowanych modeli (wszystkie tranformacje dla danych pakiet odwraca automatycznie przy wyznaczaniu prognoz).



Rysunek 5: Prognozy dla modelu DHR oraz ARIMA

Prognoza dla modelu ARIMA wydaję się być bardzo podobna do metody sezonowej naiwnej, natomiast dynamiczny model regresji wyłapał dynamikę opadającego nieliniowego trendu, dlatego też prognozę dla tego modelu uzwględnie w finalnej kombinacji prognoz.

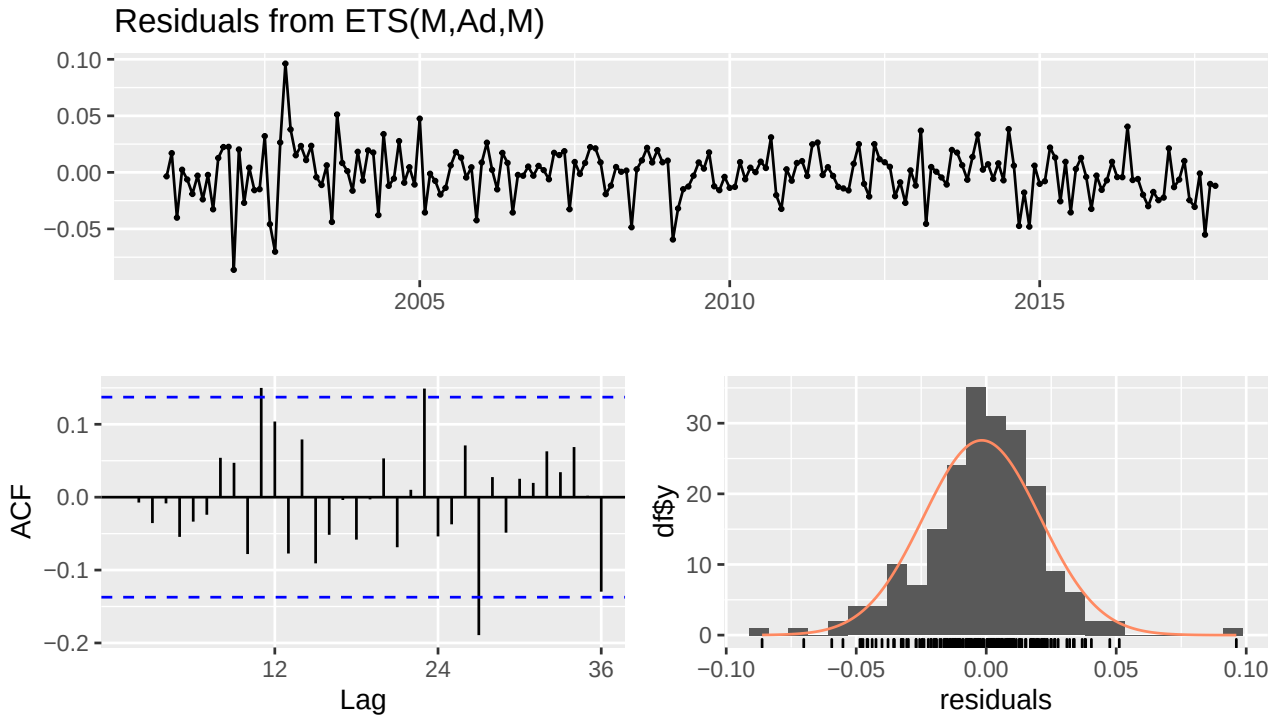
2.2 ETS

Modele wygładzania wykładniczego to druga popularna rodzina modeli statystycznych, zawierająca zarówno modele liniowe jak i nieliniowe. W najabrdziej ogólnej postaci mówimy o modelu ETS (*Exponential smoothing state space model*), gdzie E – errors, T – trend, S – seasonality. Optymalne parametry modeli wybierane są automatycznie, używając funkcji `ets`, która dobiera odpowiednie komponenty przestrzeni stanów oraz parametry wygładzania za pomocą kryteriów informacyjnych (domyślnie AICc).

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = ts.BICC)
##
##   Smoothing parameters:
```

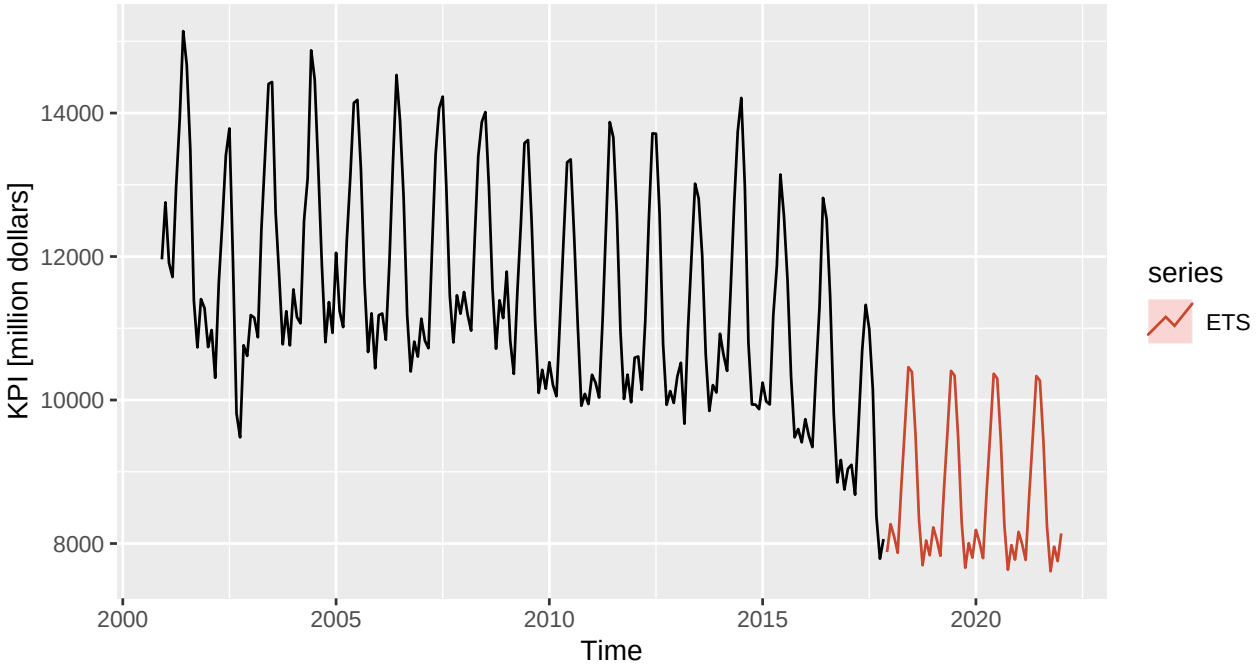
```
##     alpha = 0.7323
##     beta  = 0.0016
##     gamma = 1e-04
##     phi   = 0.9795
##
##   Initial states:
##     l = 13389.4087
##     b = -32.4577
##     s = 0.9218 0.8819 0.9536 1.0904 1.1889 1.1961
##           1.0981 1.0042 0.8988 0.9241 0.9435 0.8986
##
##   sigma:  0.0234
##
##      AIC     AICc      BIC
## 3382.792 3386.489 3442.518
```

Dopasowany został model ETS(M,Ad,M) o multiplikatywnej strukturze sezonowości oraz błędów oraz o addytywnym wyagasającym trendzie. Powyższy wynik zwraca również parametry wygładzające dla poszczególnych parametrów modelu. Poniżej wykres dla reszt modelu.



Rysunek 6: Reszty dla modelu ETS

W dalszym ciągu nie ma mowy o perfekcyjnym dopasowaniu, natomiast tak jak to miało miejsce dla poprzednich modeli reszty są losowe (nie występuję istotna autokorelacja); model wygląda na dobrze dopasowany.



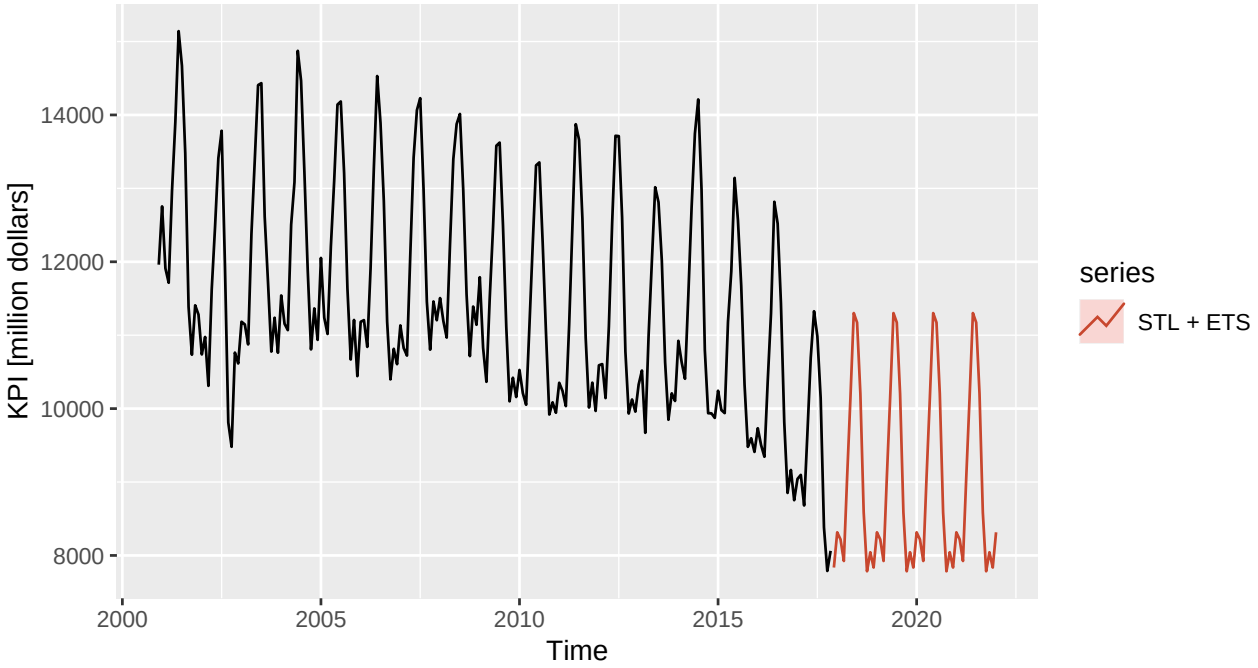
Rysunek 7: Prognozy dla modelu ETS

Uzyskana prognoza dla modelu ETS(M,Ad,M) wydaję się być podobna do prognozy dla modelu ARIMA(1,0,1)(2,1,1)[12], niemniej w dalszym ciągu warto uwzlględnić ją w kombinacji prognoz.

2.3 ETS + STL

Wyżej wspomniany model może być również używany w połączeniu ze wspomnianą wcześniej dekompozycją STL.

```
STL <- stlf(ts.BICC, h = 50)
```



Rysunek 8: Prognozy dla modelu STL + ETS

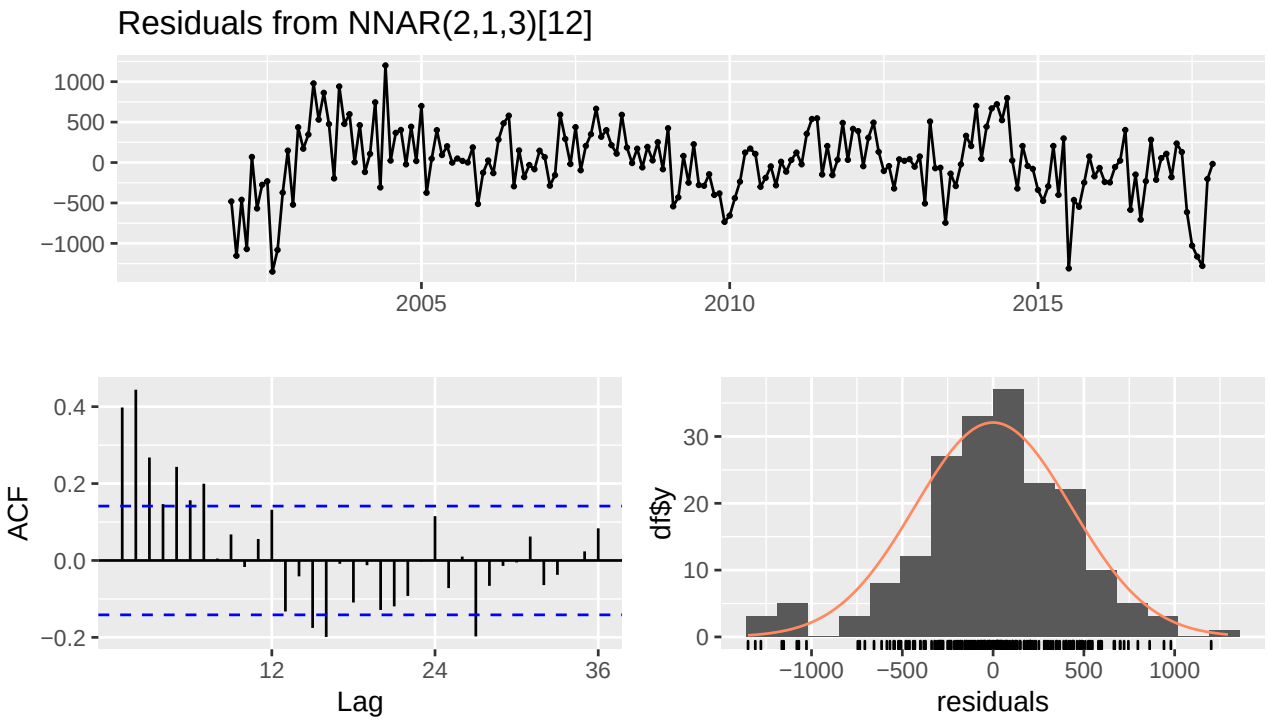
Dla stacjonarnej pozostałości dopasowany został model ETS(M, N, N), który odpowiada prostemu wygładzaniu wykładniczemu z multiplikatywnymi błędami; estymowany trend oraz sezonowość zostały ekstrapolowane. Prognoza wyznaczona przez model STL + ETS charakteryzuję się wiekszą amplitudą sezonową niż prognoza modelu ETS(M,Ad,M).

2.4 NNAR

NNAR (*Neural network autoregression*) to model łączący komponent autoregresyjny z architekturą prostej sieci neuronowej z jedną ukrytą warstwą modelującą nieliniowe zależności szeregu czasowego. Używając funkcji `nnetar` można dopasować odpowiedni model sezonowy NNAR(p, P, k), gdzie p oraz P to rzędy wielomianów autoregresyjnych, natomiast k odpowiada za liczbę neuronów w warstwie ukrytej. Model NNAR(p, P, 0) to po prostu model ARIMA(p,0,0)(P,0,0).

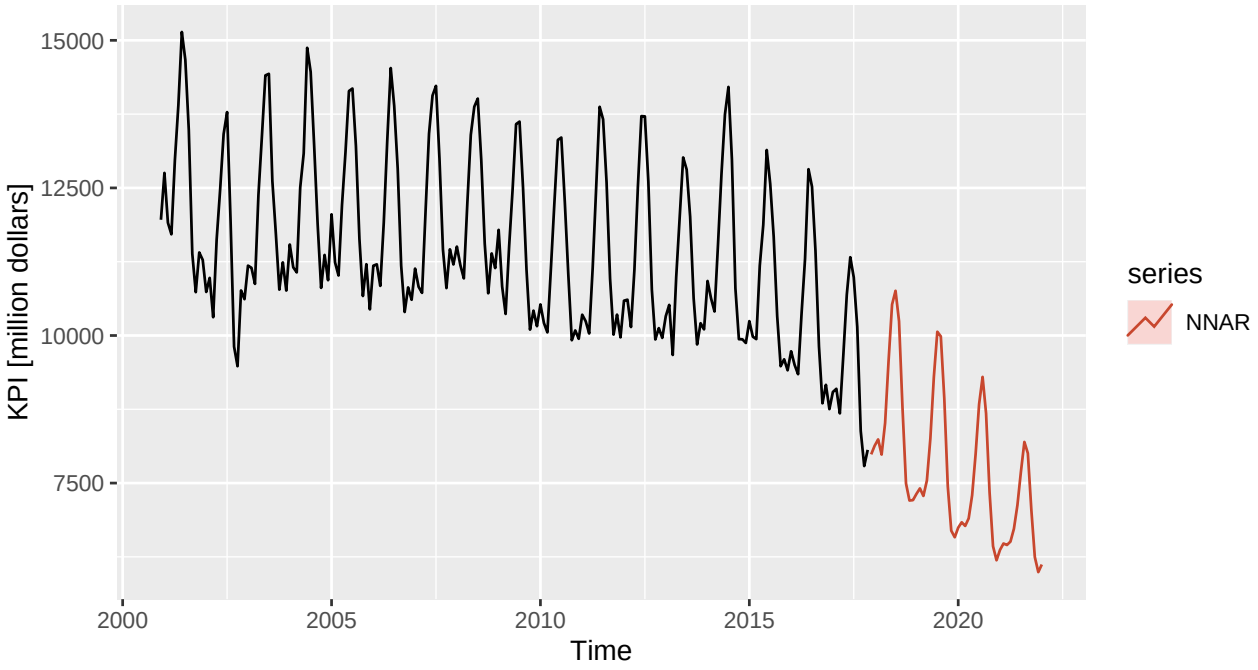
```
## Series: ts.BICC
## Model:  NNAR(2,1,3)[12]
## Call:   nnetar(y = ts.BICC, size = 3, repeats = 100)
##
## Average of 100 networks, each of which is
## a 3-3-1 network with 16 weights
## options were - linear output units
##
## sigma^2 estimated as 185090
```

Do danych został dopasowany model NNAR(2,1,3), gdzie rząd $p = 2$ został wybrany za pomocą kryterium AIC, reszta parametrów została ustalona.



Rysunek 9: Reszty dla modelu NNAR

Model wydaję się pomijać sporą część informacji dla początkowych opóźnień, niemniej z racji na jego odmienny charakter i nielinową specyfikę warto również rozważyć dodanie go jako składnika kombinacji.



Rysunek 10: Prognozy dla modelu NNAR

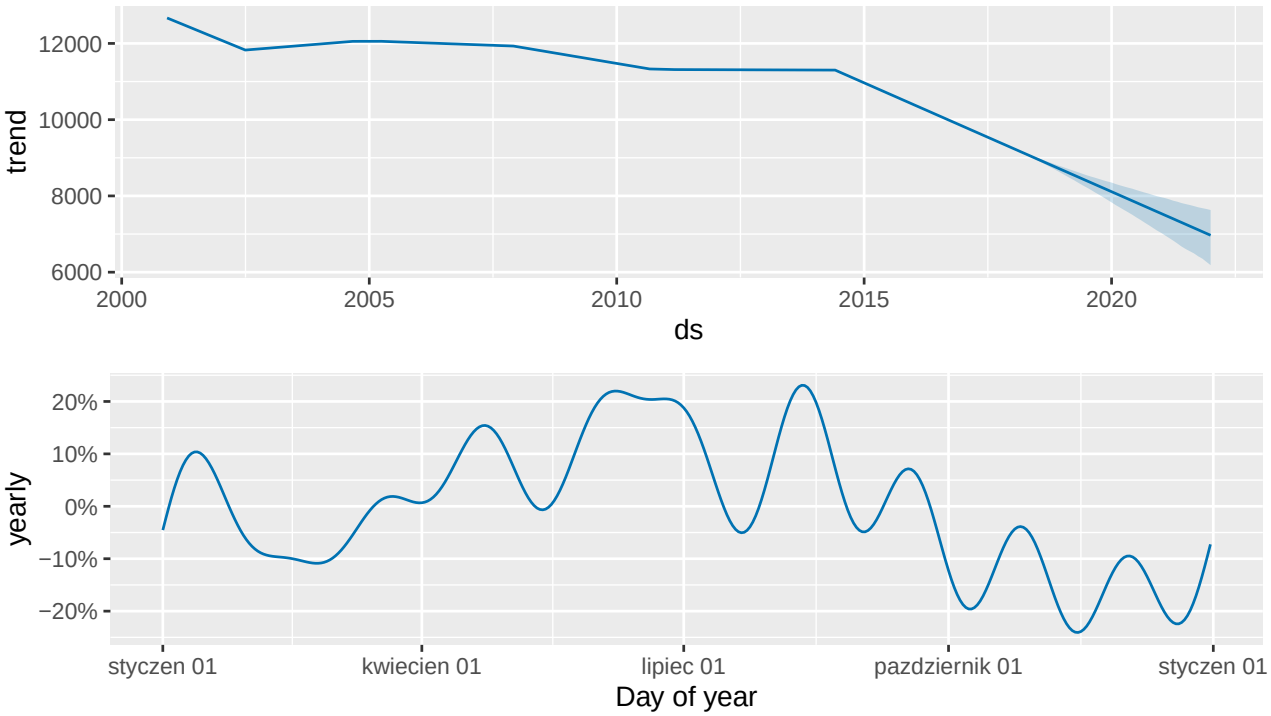
2.5 Prophet

Ciekawym modelem nieliowym jest automatyczny algorytm zaproponowany w 2018 przez analityków Facebooka, czyli Prophet [5]. Prophet sprowadza szereg czasowy do postaci:

$$y_t = g(t) + h(t) + s(t) + \epsilon_t, \tag{1}$$

gdzie $g(t)$ reprezentuje liniowy trend (lub nieliniowy składnik wzrostu), $h(t)$ modeluje efekty świąt używając odpoweidnich *dummy variables*, natomiast $s(t)$ to składnik sezonowy modelowany przy pomocy zmiennych fourierowskich; do tego wszystkiego dodawane jest losowe zakłócenie. Model estymowany jest przy pomocy metod bayesowskich, by umożliwić automatyczne wykrywanie punktów zmian trendu oraz wyżej wymienionych komponentów.

```
df.BICC <- data.frame(ds = zoo::as.Date(time(ts.BICC)), y = matrix(ts.BICC))
prophet <- prophet(df.BICC, seasonality.mode = 'multiplicative')
future <- make_future_dataframe(prophet, periods = 50, freq = 'month')
predict <- predict(prophet, future)
PROPHET <- ts(tail(predict[["yhat"]], 50), start = c(2017, 12), frequency = 12)
```



Rysunek 11: Komponenty modelu Prophet

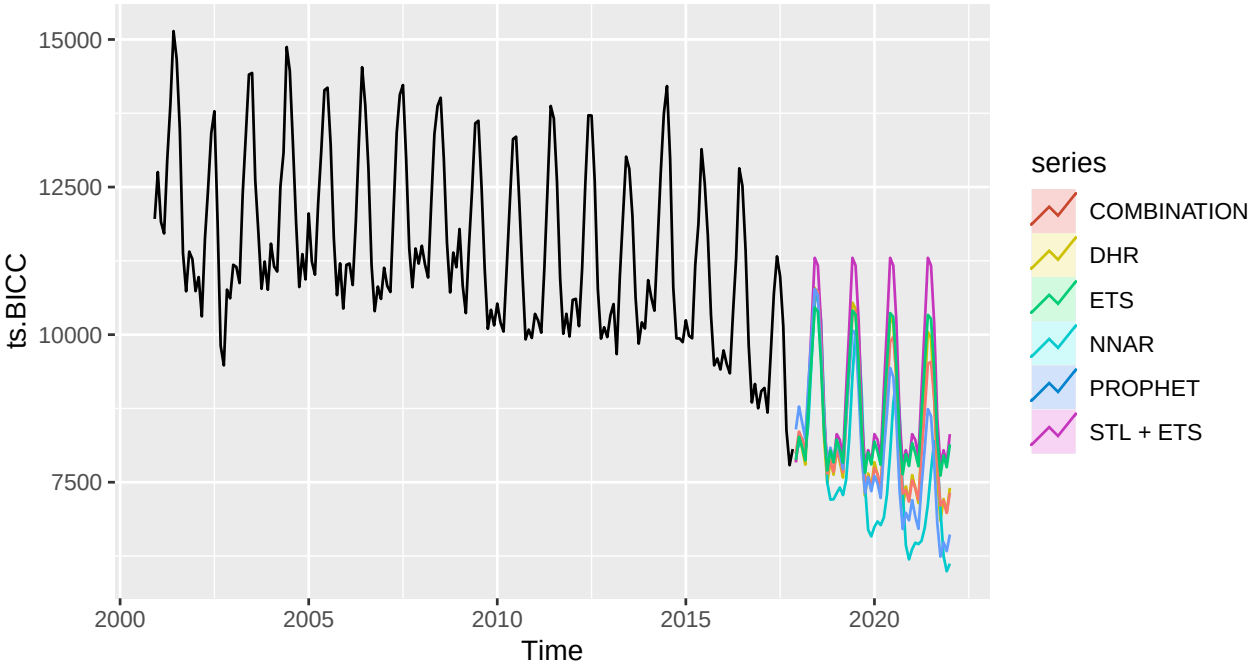
Rysunek 12: Prognozy dla modelu Prophet

Prophet przy wyznaczaniu prognoz również esktrapolował końcową tendencję spadkową trendu wraz z multiplikatywną strukturą sezonowości. Prognoza wydaję się być adekwatna, dlatego uwzględnie ją w finalnej kombinacji.

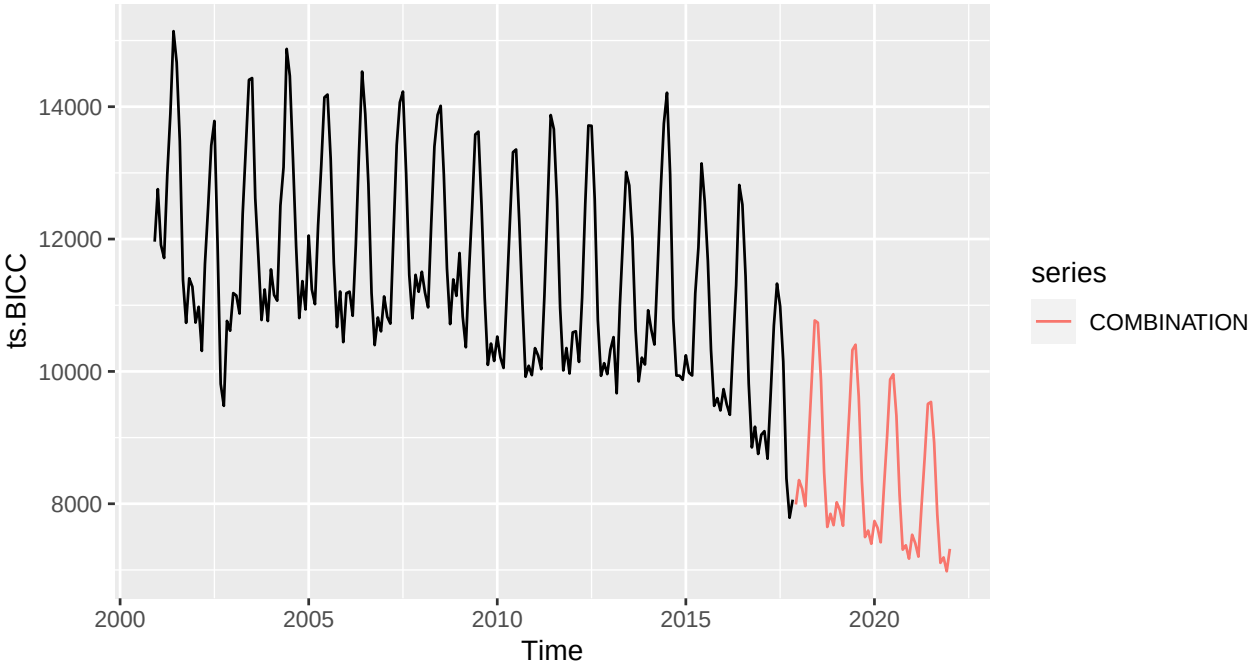
2.6 Kombinacja

Poniższe wykresy reprezentują prognozy dla omówionych modeli oraz prognozę wyznaczoną na podstawie kombinacji ich przewidywań.

```
COMBINATION <- (DHR[["mean"]] + NNAR[["mean"]] +
                  ETS[["mean"]] + PROPHET + STL[["mean"]])/5
```



Rysunek 13: Porównanie otrzymanych prognoz



Rysunek 14: Prognoza dla kombinacji metod

3 Wyznaczenie prognozy

Poniżej wyznaczona (podlegająca ocenie) prognoza dla metody kombinacyjnej – czyli zwykłej średniej arytmetycznej dla prognoz modeli: DHR, ETS, STL + ETS, NNAR, PROPHET.

```
##           Jan       Feb       Mar       Apr       May       Jun       Jul
## 2017                                                                      
## 2018 8358.045  8226.873  7964.398  8882.689  9823.137 10770.347 10736.347
## 2019 8022.848  7904.555  7667.740  8517.475  9380.627 10324.021 10403.802
## 2020 7740.310  7635.737  7417.817  8221.377  8992.775  9878.381  9957.398
## 2021 7532.537  7402.169  7200.994  7976.744  8705.258  9509.169  9538.963
## 2022 7318.762                                                             
##           Aug       Sep       Oct       Nov       Dec
## 2017                                          7989.059
## 2018 9862.396  8471.895  7649.321  7848.783  7676.410
## 2019 9621.110  8344.692  7496.134  7595.449  7395.603
## 2020 9330.048  8132.045  7306.163  7372.672  7170.410
## 2021 8924.309  7840.253  7106.618  7189.294  6979.407
## 2022
```

4 Podsumowanie

W powyższej pracy zaproponowałem metodę prognozowania opartą na kombinacji wyników kilku metod statystycznych w celu poprawienia finalnej skuteczności. Wszystkie użyte w kombinacji modele mają charakter nieliniowy, by lepiej radzić sobie z charakterystyką badanych danych. Metody wykorzystywały różne metody modelowania 12-miesięcznej sezonowości takie jak różnicowanie sezonowe, komponenty fourierowskie, wygładzanie wykładnicze oraz dekompozycja STL. W pracy została wprowadzona szeroka taksonomia metod, by podejść do badanego zagadnienia w możliwe najpełniejszy sposób. Uzyskana finalna prognoza wydaje się dobrze odzwierciedlać dynamikę oraz sezonowość zawartą w danych historycznych.

Literatura

[1] Cleveland RB, Cleveland WS, McRae JE, Terpenning I. STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion). Journal of Official Statistics. 1990;6:3–73.

[2] Bates JM, Granger CWJ. The Combination of Forecasts. Journal of the Operational Research Society. 1969 dec;20(4):451–468.

[3] Kokoszka P, Young G. KPSS test for functional time series. Statistics. 2016 feb;50(5):957–973.

[4] Hyndman RJ, Khandakar Y. Automatic time series forecasting: the forecast package for R. Journal of Statistical Software. 2008;26(3):1–22. Available from: https://www.jstatsoft.org/article/view/v027i03.

[5] Taylor SJ, Letham B. Forecasting at Scale. The American Statistician. 2018 jan;72(1):37–45.

[6] Hyndman R, Athanasopoulos G. Forecasting: Principles and Practice. 2nd ed. Australia: OTexts; 2018.