Jak zautomatyzować prognozowanie szeregów wykorzystując R?

Adam Zagdański

QuantUp
http://www.quantup.pl

7 marca 2014

Spis treści

1 Wprowadzenie

1.1 Potrzeba automatycznej konstrukcji prognoz

- Podejmowanie ważnych decyzji biznesowych, dotyczących planowania, kupna/sprzedaży, zatrudnienia, itp. wymaga często automatycznej konstrukcji prognoz dla dużej liczby (jednowymiarowych) szeregów czasowych.
- Przykładem może być np. konieczność prognozowania (przynajmniej raz w miesiącu) wielkości sprzedaży dla ponad tysiąca produktów.
- Nawet wtedy, gdy prognozowanie potrzebne jest dla dużo mniejszej liczby szeregów, konstrukcja optymalnych prognoz wymaga zatrudnienia pracowników (analityków) posiadających odpowiednią wiedzę i doświadczenie w zakresie analizy szeregów czasowych.
- Wybór nieoptymalnego modelu może mieć poważne konsekwencje dla dokładności prognoz!
- Istnieje więc zapotrzebowanie na narzędzia/algorytmy umożliwiające całkowicie lub przynajmniej częściowo automatyczne prognozowanie szeregów czasowych.

1.2 Pożądane własności algorytmów automatycznego prognozowania

- Algorytmy pozwalające na automatyczną konstrukcję prognoz powinny realizować wszystkie etapy analizy, tzn.:

  1. wybór optymalnego modelu dla danych,
  2. estymacja parametrów,
  3. konstrukcja prognoz (punktowych i przedziałowych).

- W poszukiwaniu optymalnego modelu ważne jest zastosowanie odpowiednich kryteriów, zabezpieczających przed przeparametryzowaniem (tzn. zbyt dobrym dopasowaniem modelu dla danych konstrukcyjnych/uczących, co może prowadzić do złej jakości prognoz dla nowych okresów).
- Algorytmy powinny być odporne w przypadku występowania w analizowanych szeregach czasowych niestandardowych wzorców/zachowań.
- Powinniśmy móc zastosować je dla dużej liczby szeregów bez konieczności ingerencji ze strony analityka.
- W przypadku dużej liczby szeregów nie bez znaczenia jest też złożoność obliczeniowa algorytmów.

1.3 Narzędzia dostępne w R – pakiet *forecast*.

- Efektywne algorytmy umożliwiające automatyczną konstrukcję prognoz są udostępnione m.in. w pakiecie *forecast* (Hyndman i Khandakar (2008)).
- Pakiet pozwala na automatyczne prognozowanie z wykorzystaniem dwóch najbardziej popularnych metod:

  - **modele ARIMA** – klasyczna rodzina modeli dla szeregów czasowych, wprowadzona przez Boxa-Jenkinsa (1976)
  - **modele ETS** (ExponenTialSmoothing lub ErrorTrendSeason) – rodzina modeli opracowana przez Hyndmana i innych (2002), która stanowi uogólnienie algorytmów wygładzania wykładniczego

- Oba dostępne algorytmy (ARIMA i ETS) mogą być zastosowane dla szeregów, w których wy stępuje trend długoterminowy i/lub sezonowość.
- Istnieją również rozwiązania przyspieszające działanie algorytmów (takie jak możliwość specyfikacji ograniczeń dla wybranych parametrów czy krokowa metoda wyboru najlepszego modelu).
- W dalszej części zilustrujemy działanie obu algorytmów dla czterech wybranych szeregów czasowych.
- Dodatkowe informacje nt. możliwości pakietu *forecast* można znaleźć w artykule: Rob J. Hyndman & Yeasmin Khandakar (2008), *Automatic Time Series Forecasting: The forecast Package for R*, Journal of Statistical Software, American Statistical Association, vol. 27(i03). http://www.jstatsoft.org/v27/i03/paper

2 Dane

- W analizie wykorzystamy cztery szeregi czasowe, o różnej częstotliwości (dane miesięczne, kwartalne i dzienne) i pochodzących z różnych obszarów zastosowań:
  - **bezrobocie** - stopa bezrobocia w Polsce w latach 1990-2013, dane miesięczne,
  - **ceny benzyny** - średnie ceny benzyny w Polsce w latach 2006-2013, dane miesięczne,
  - **kurs EUR/PLN** - dane dzienne w okresie: od 2013-01-02 do 2013-08-23,
  - **Produkt Krajowy Brutto (PKB)** - dane kwartalne w latach: 1998-2013.

2.1 Wczytanie danych

- Analizę rozpoczynamy od wczytania danych, zapisanych uprzednio w plikach *RData* (format binarny R'a).
- Wszystkie szeregi czasowe były zapisane jako obiekty klasy *ts* lub *xts* (dedykowanych tego typu danych), co ułatwi nam ich wizualizację i analizę.

```
library(xts) #potrzebne do wykresu danych PLN/EUR

# wczytanie danych
load("../data/bezrobocie.ts.RData")
load("../data/cena.benzyny.ts.RData")
load("../data/kurs.EUR.PLN.xts.RData")
load("../data/PKB.RData")
```

- Przyjrzyjmy się poszczególnym szeregom.

- Aby poprawić przejrzystość prezentacji, dla każdego z analizowanych szeregów wypiszemy tylko wartości dla kilku początkowych lat/okresów.

```
print(bezrobocie.ts)
```

```
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1990 0.3 0.8 1.5 1.9 2.4 3.1 3.8 4.5 5.0 5.5 5.9 6.5
## 1991 6.6 6.8 7.1 7.3 7.7 8.4 9.4 9.8 10.5 10.8 11.1 12.2
## 1992 12.1 12.4 12.1 12.2 12.3 12.6 13.1 13.4 13.6 13.5 13.5 14.3
## 1993 14.2 14.4 14.4 14.4 14.3 14.8 15.4 15.4 15.4 15.3 15.5 16.4
```

```
print(cena.benzyny.ts)
```

```
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2006 3.77 3.72 3.72 4.02 4.21 4.21 4.32 4.40 4.25 3.97 3.80 3.65
## 2007 3.66 3.78 4.04 4.25 4.35 4.46 4.46 4.40 4.36 4.33 4.43 4.42
## 2008 4.33 4.29 4.36 4.34 4.48 4.65 4.72 4.59 4.46 4.35 4.06 3.66
## 2009 3.46 3.81 3.88 3.95 4.11 4.45 4.59 4.49 4.42 4.30 4.36 4.29
```

```
print(kurs.EUR.PLN.xts)
```

```
## [,1]
## 2013-01-02 4.067
## 2013-01-03 4.077
## 2013-01-04 4.125
## 2013-01-07 4.122
## 2013-01-08 4.126
```

```
print(PKB)
```

```
## Qtr1 Qtr2 Qtr3 Qtr4
## 1998 106.5 105.3 104.9 102.9
## 1999 101.6 103.0 105.0 106.2
## 2000 106.0 105.2 103.3 102.4
## 2001 102.0 100.9 100.8 100.2
```

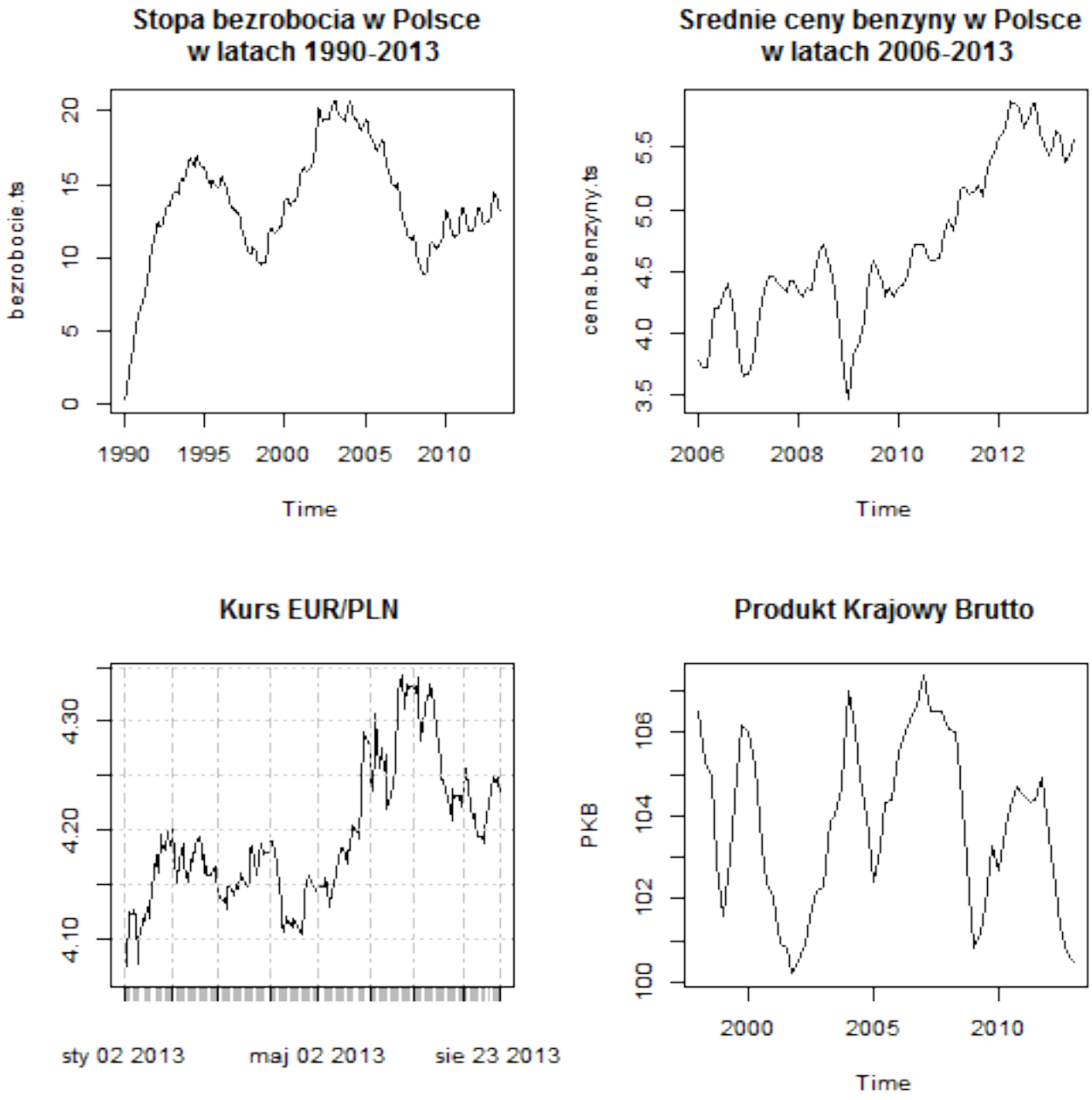
2.2 Wykres danych i ACF

- Pierwszym krokiem w analizie szeregu czasowego jest zwykle przedstawienie danych na wykresie.
- Analiza wykresu może pomóc nam w zidentyfikowaniu regularnych wzorców występujących w danych, takich jak trendy (tendencje długoterminowe) lub efekty sezonowe (sezonowość).

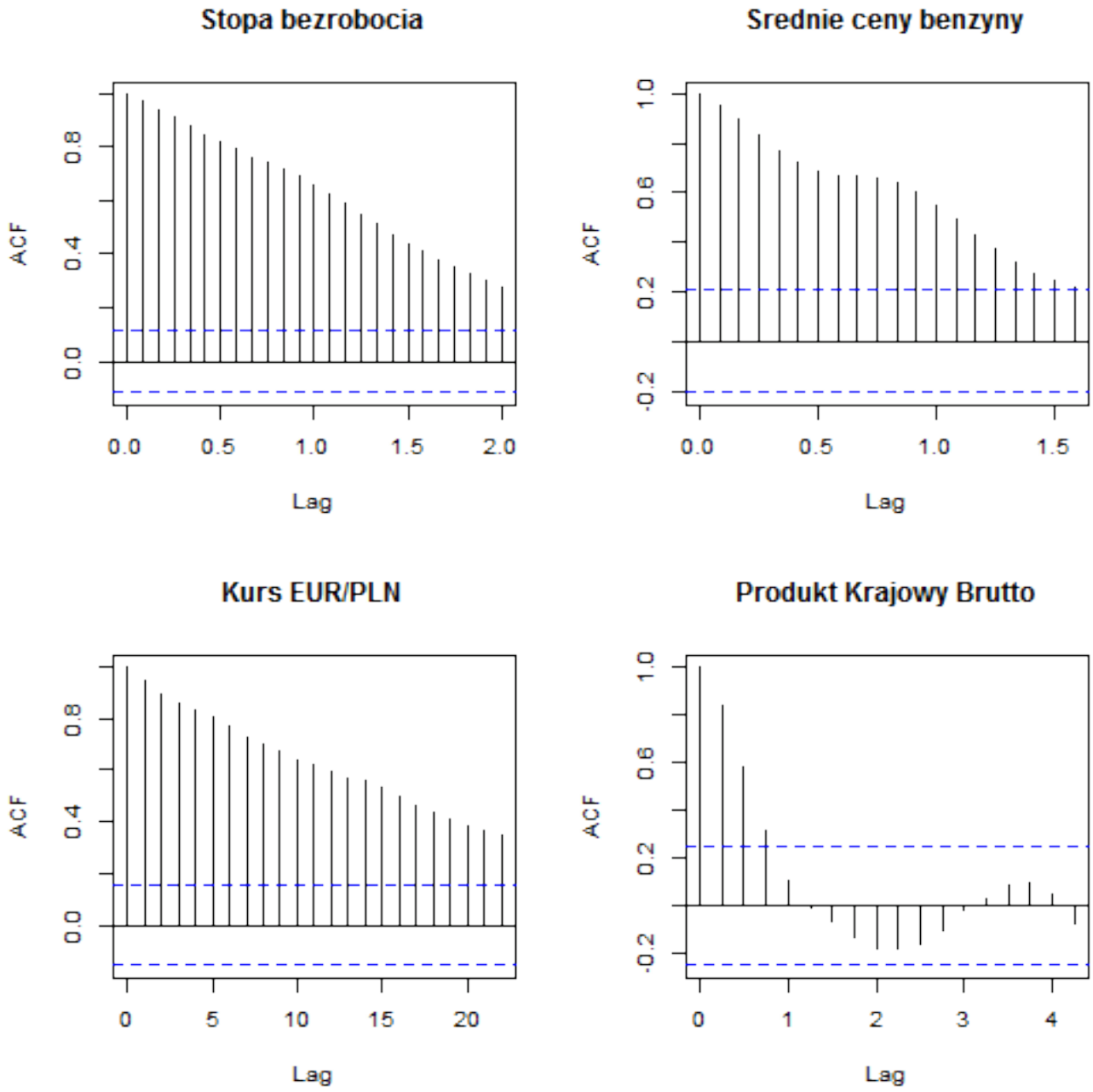
```
# wykres danych
par(mfrow = c(2, 2))
plot(bezrobocie.ts, main = "Stopa bezrobocia w Polsce \n w latach 1990-2013")
plot(cena.benzyny.ts, main = "Srednie ceny benzyny w Polsce \n w latach 2006-2013")
plot(kurs.EUR.PLN.xts, main = "Kurs EUR/PLN")
plot(PKB, main = "Produkt Krajowy Brutto")
```

- Aby zilustrować siłę i charakter zależności/korelacji czasowej wyznaczymy dodatkowo wykresy funkcji autokorelacji (ACF) dla poszczególnych szeregów.

```
# ACF
par(mfrow = c(2, 2))
acf(bezrobocie.ts, main = "Stopa bezrobocia")
acf(cena.benzyny.ts, main = "Srednie ceny benzyny")
acf(kurs.EUR.PLN.xts, main = "Kurs EUR/PLN")
acf(PKB, main = "Produkt Krajowy Brutto")
```



Rysunek 1: Wykresy wyjściowych szeregów czasowych



Rysunek 2: Wykresy funkcji autokorelacji (ACF)

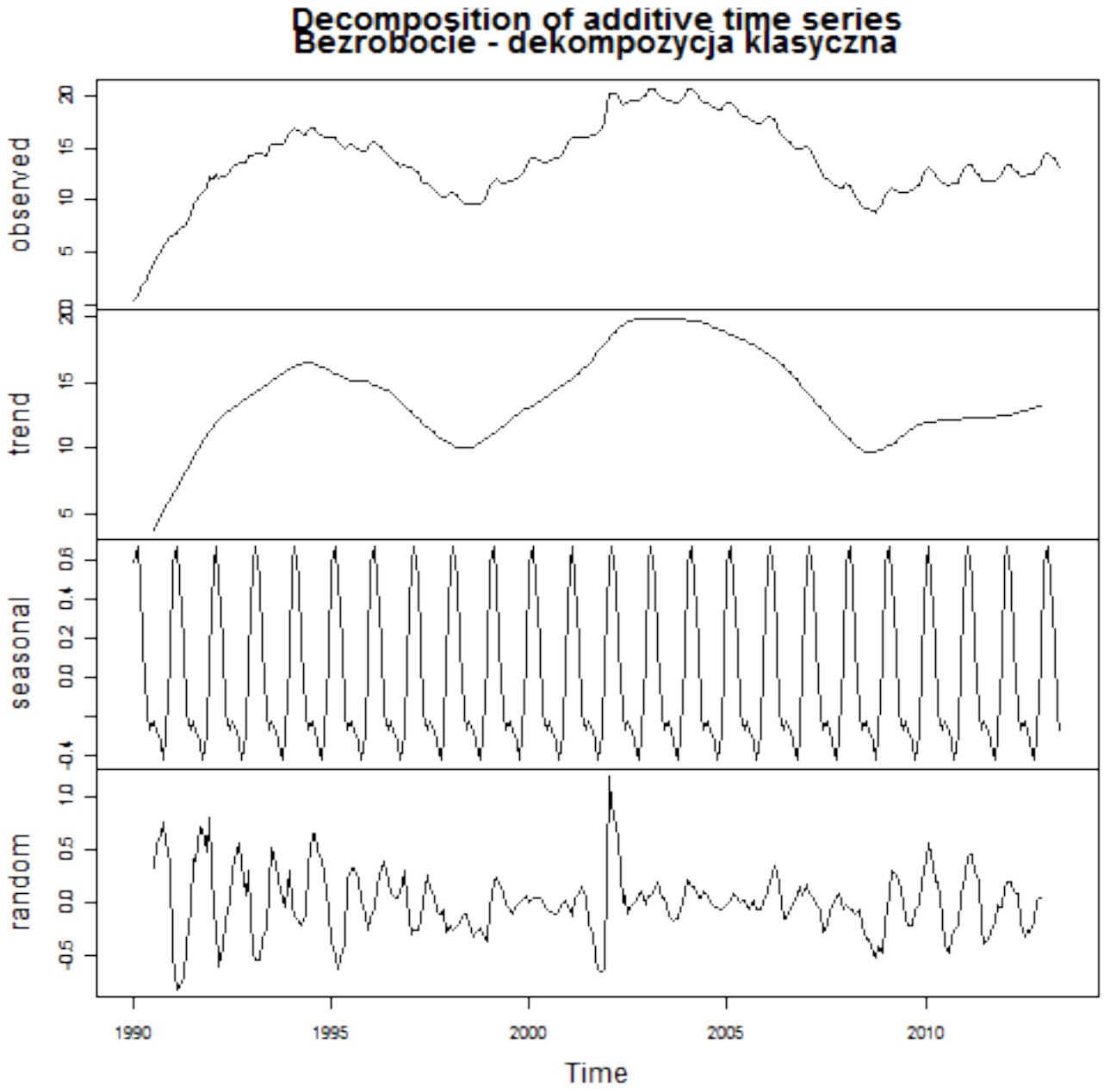
3 Dekompozycja

- Dekompozycję szeregów na: **a)** trend długoterminowy, **b)** sezonowość i **c)** losowe fluktuacje, przeprowadzimy z wykorzystaniem funkcji *decompose()* (metoda klasycznej dekompozycji).
- Dekompozycję zastosujemy tylko dla szeregów, w których występuje sezonowość, tzn:
  - bezrobocie,
  - średnie ceny benzyny.

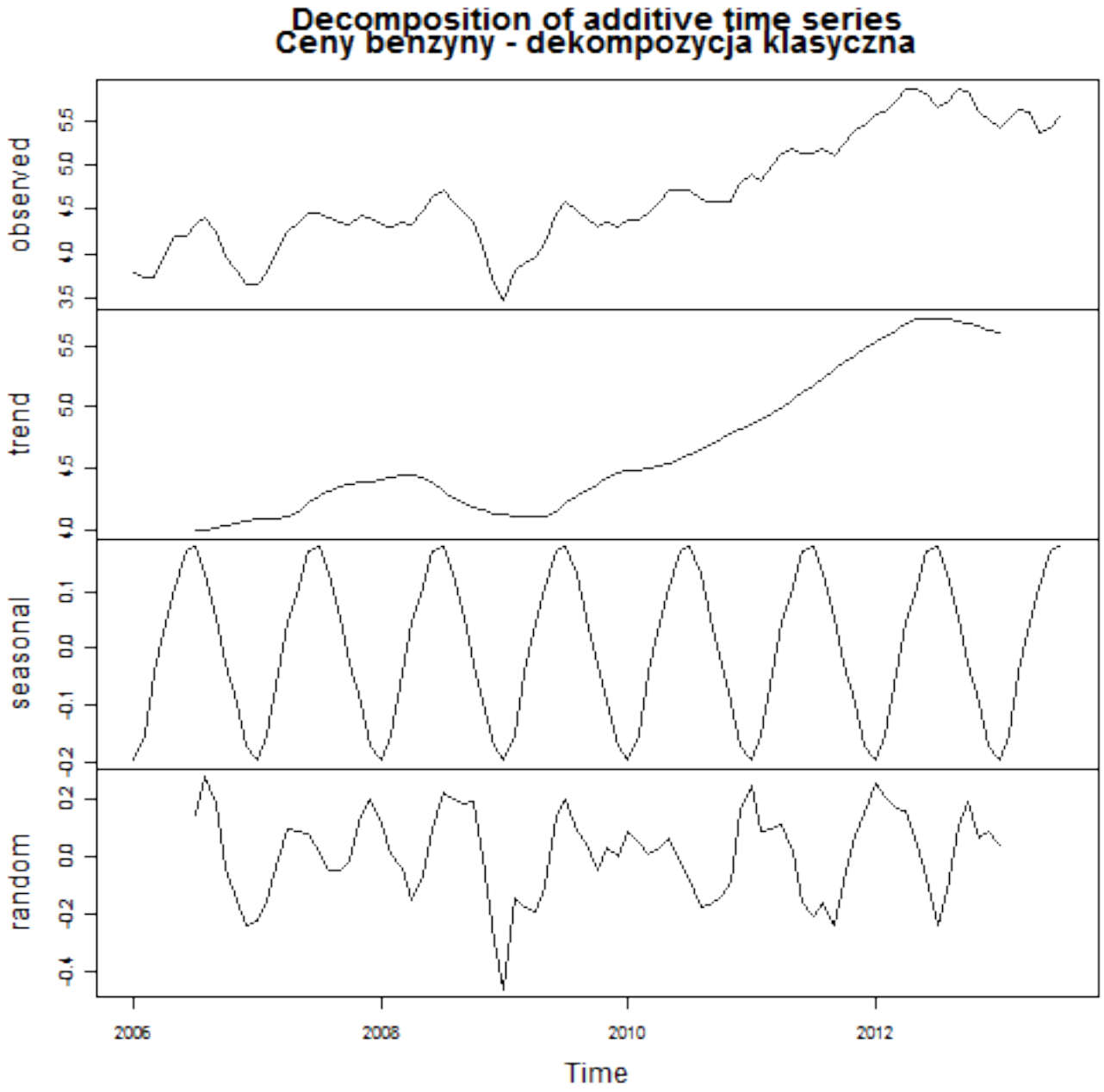
```
# dekompozycja klasyczna
bezrob.dekomp <- decompose(bezrobocie.ts)
benzyna.dekomp <- decompose(cena.benzyny.ts)
```

```
plot(bezrob.dekomp)
title("Bezrobocie - dekompozycja klasyczna")
```

```
plot(benzyna.dekomp)
title("Ceny benzyny - dekompozycja klasyczna")
```



Rysunek 3: Dekompozycja dla szeregu 'bezrobocie'



Rysunek 4: Dekompozycja dla szeregu cen benzyny

4 Prognozowanie

- Zastosujemy dwie metody konstrukcji prognoz:
  - **ARIMA** – klasyczna rodzina modeli, wprowadzona przez Boxa-Jenkinsa (1976),
  - **ETS** (ExponenTialSmoothing lub ErrorTrendSeason) – uniwersalna rodzina modeli dla szeregów czasowych zaproponowana przez Hyndmana i innych (2002) (uogólnienie metod wygładzania wykładniczego).
- Optymalne modele dla poszczególnych szeregów wybierane są całkowicie automatycznie, z wykorzystaniem kryteriów statystycznych oceniających dokładność dopasowania.
- Oprócz prognoz punktowych wyznaczymy także przedziały predykcyjne dla dwóch poziomów ufności: 80% i 95%.

4.1 Podział danych na część uczącą (konstrukcyjną) i testową

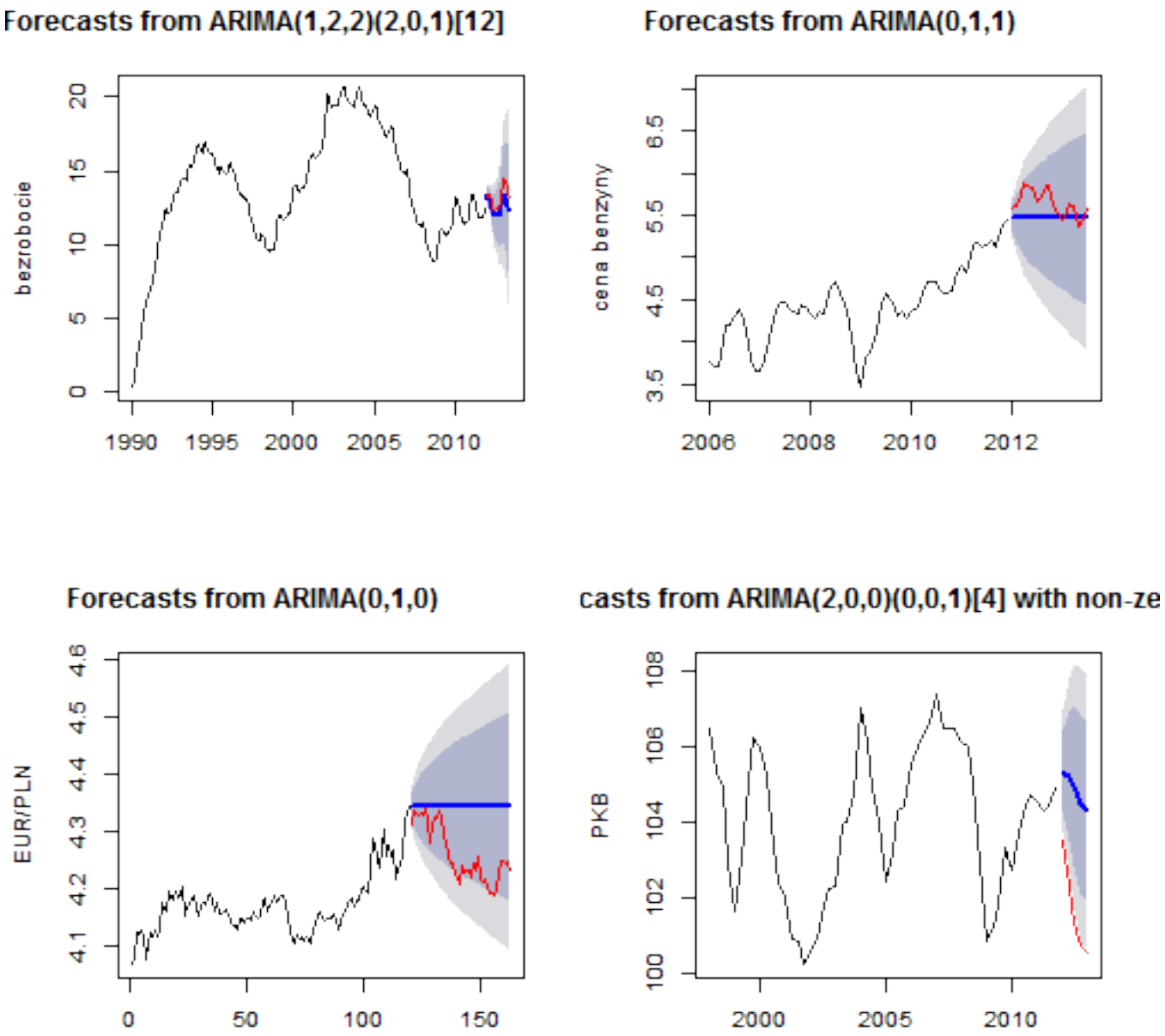
- Aby w wiarygodny sposób porównać dokładność prognoz podzielimy każdy z analizowanych szeregów na dwie części:
  - **zbiór uczący (konstrukcyjny)** – podzbiór danych, który będzie wykorzystany do dopasowania modelu i konstrukcji prognoz,
  - **zbiór testowy** – dane dla ostatnich kilku/kilkunastu okresów, który będzie wykorzystany do oceny dokładności prognoz.

```
bezrobocie.ts.learn <- window(bezrobocie.ts, end = c(2011, 12))
bezrobocie.ts.test <- window(bezrobocie.ts, start = c(2012, 1))

cena.benzyny.ts.learn <- window(cena.benzyny.ts, end = c(2011, 12))
cena.benzyny.ts.test <- window(cena.benzyny.ts, start = c(2012, 1))

kurs.EUR.PLN.ts.learn <- window(as.ts(kurs.EUR.PLN.xts), end = 120)
kurs.EUR.PLN.ts.test <- window(as.ts(kurs.EUR.PLN.xts), start = 121)

PKB.ts.learn <- window(PKB, end = c(2011, 4))
PKB.ts.test <- window(PKB, start = c(2012, 1))
```

4.2 ARIMA

- Zaczynamy od prognozowania na bazie klasycznych modeli z klasy ARIMA (AutoRegressive Integrated Moving Average).
- Dla każdego z szeregów (niezależnie) dopasujemy model ARIMA(p,d,q)(P,D,Q)[s].
- Aby znaleźć optymalny model wykorzystamy funkcję *auto.arima()* (pakiet *forecast*) dla domyślnych parametrów (takich jak kryterium dokładności dopasowania modelu, itp.).

```
library(forecast)

# ARIMA
bezr.arima <- auto.arima(bezrobocie.ts.learn)
benzyna.arima <- auto.arima(cena.benzyny.ts.learn)
EUR.arima <- auto.arima(kurs.EUR.PLN.ts.learn)
PKB.arima <- auto.arima(PKB.ts.learn)



bezr.arima.forecast <- forecast(bezr.arima, h = length(bezrobocie.ts.test))
benzyna.arima.forecast <- forecast(benzyna.arima, h = length(cena.benzyny.ts.test))
EUR.arima.forecast <- forecast(EUR.arima, h = length(kurs.EUR.PLN.ts.test))
PKB.arima.forecast <- forecast(PKB.arima, h = length(PKB.ts.test))

par(mfrow = c(2, 2))
plot(bezr.arima.forecast, ylab = "bezrobocie")
lines(bezrobocie.ts.test, col = "red")
plot(benzyna.arima.forecast, ylab = "cena benzyny")
lines(cena.benzyny.ts.test, col = "red")
plot(EUR.arima.forecast, ylab = "EUR/PLN")
lines(kurs.EUR.PLN.ts.test, col = "red")
plot(PKB.arima.forecast, ylab = "PKB")
lines(PKB.ts.test, col = "red")
```

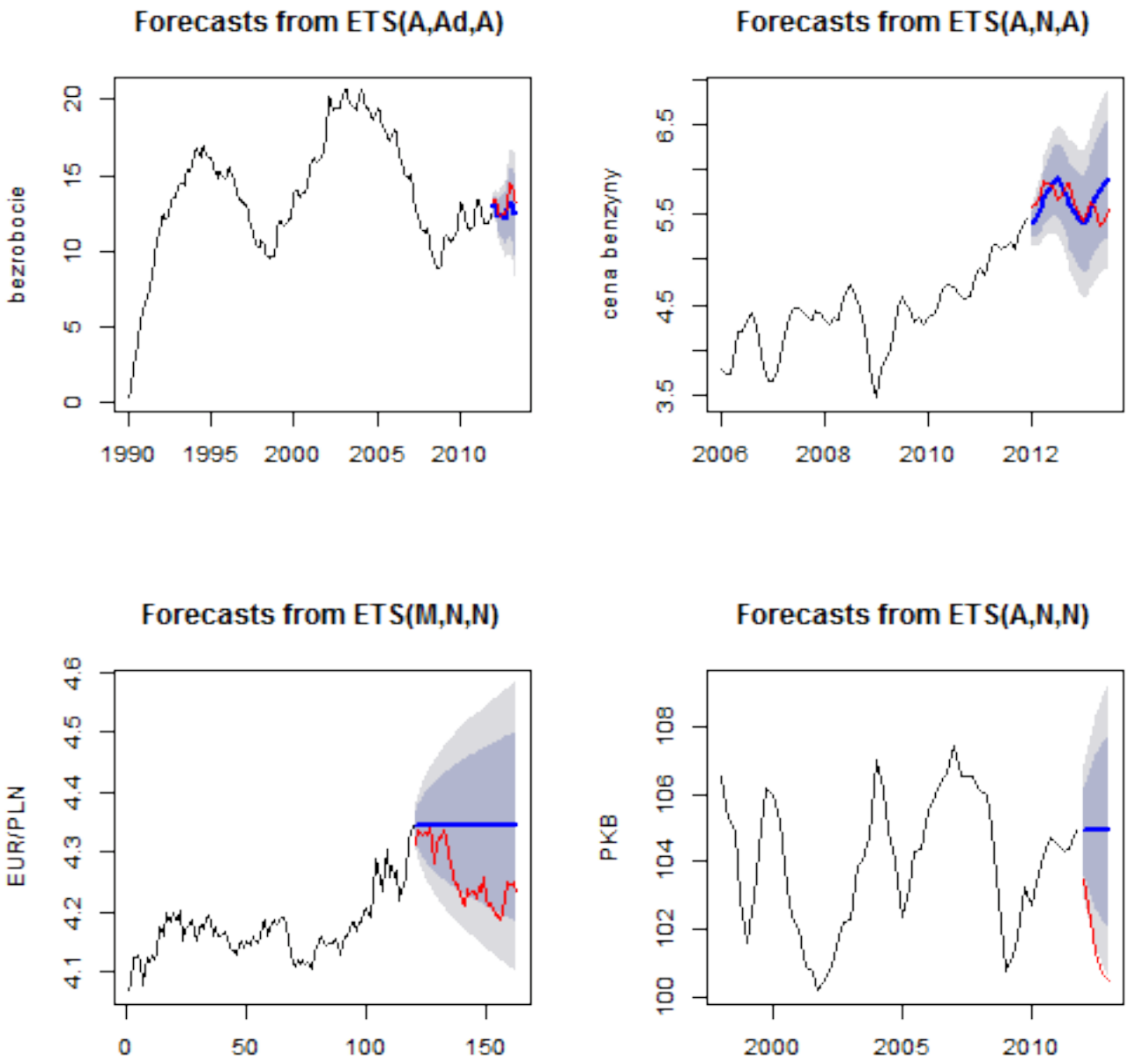


Rysunek 5: Prognozy na bazie modeli ARIMA

4.3 ETS (ExponenTialSmoothing / ErrorTrendSeason)

- W kolejnym kroku do konstrukcji prognoz zastosujemy odpowiednie warianty wygładzania wykładniczego (ang. exponential smoothing).
- Aby znaleźć optymalny model wykorzystamy funkcję *ets()* (pakiet *forecast*) dla domyślnych parametrów (takich jak kryterium dokładności dopasowania modelu, itp.).

```
library(forecast)

# ETS
bezr.ets <- ets(bezrobocie.ts.learn)
benzyna.ets <- ets(cena.benzyny.ts.learn)
EUR.ets <- ets(kurs.EUR.PLN.ts.learn)
PKB.ets <- ets(PKB.ts.learn)

bezr.ets.forecast <- forecast(bezr.ets, h = length(bezrobocie.ts.test))
benzyna.ets.forecast <- forecast(benzyna.ets, h = length(cena.benzyny.ts.test))
EUR.ets.forecast <- forecast(EUR.ets, h = length(kurs.EUR.PLN.ts.test))
PKB.ets.forecast <- forecast(PKB.ets, h = length(PKB.ts.test))

par(mfrow = c(2, 2))
plot(bezr.ets.forecast, ylab = "bezrobocie")
lines(bezrobocie.ts.test, col = "red")
plot(benzyna.ets.forecast, ylab = "cena benzyny")
lines(cena.benzyny.ts.test, col = "red")
plot(EUR.ets.forecast, ylab = "EUR/PLN")
lines(kurs.EUR.PLN.ts.test, col = "red")
plot(PKB.ets.forecast, ylab = "PKB")
lines(PKB.ts.test, col = "red")
```



Rysunek 6: Prognozy na bazie modeli ETS

4.4 ARIMA vs ETS

- Porównamy teraz prognozy uzyskane dla obu metod.
- Dla poszczególnych szeregów przedstawione są wykresy porównujące prognozy z rzeczywistymi wartościami.
- Dodatkowo, wyznaczymy miary dokładności prognoz na zbiorze uczącym (training set) i zbiorze testowym (test set).
- Wyznaczone zostaną wartości wszystkich miar dostępnych aktualnie w pakiecie *forecast*, tzn.: ME, RMSE, MAE, MPE, MAPE, MASE, ACF1 (patrz np.: Hyndman and Koehler, *Another look at measures of forecast accuracy*, International Journal of Forecasting 22 (2006) 679–688).

Prognozy dla szeregu *bezrobocie*

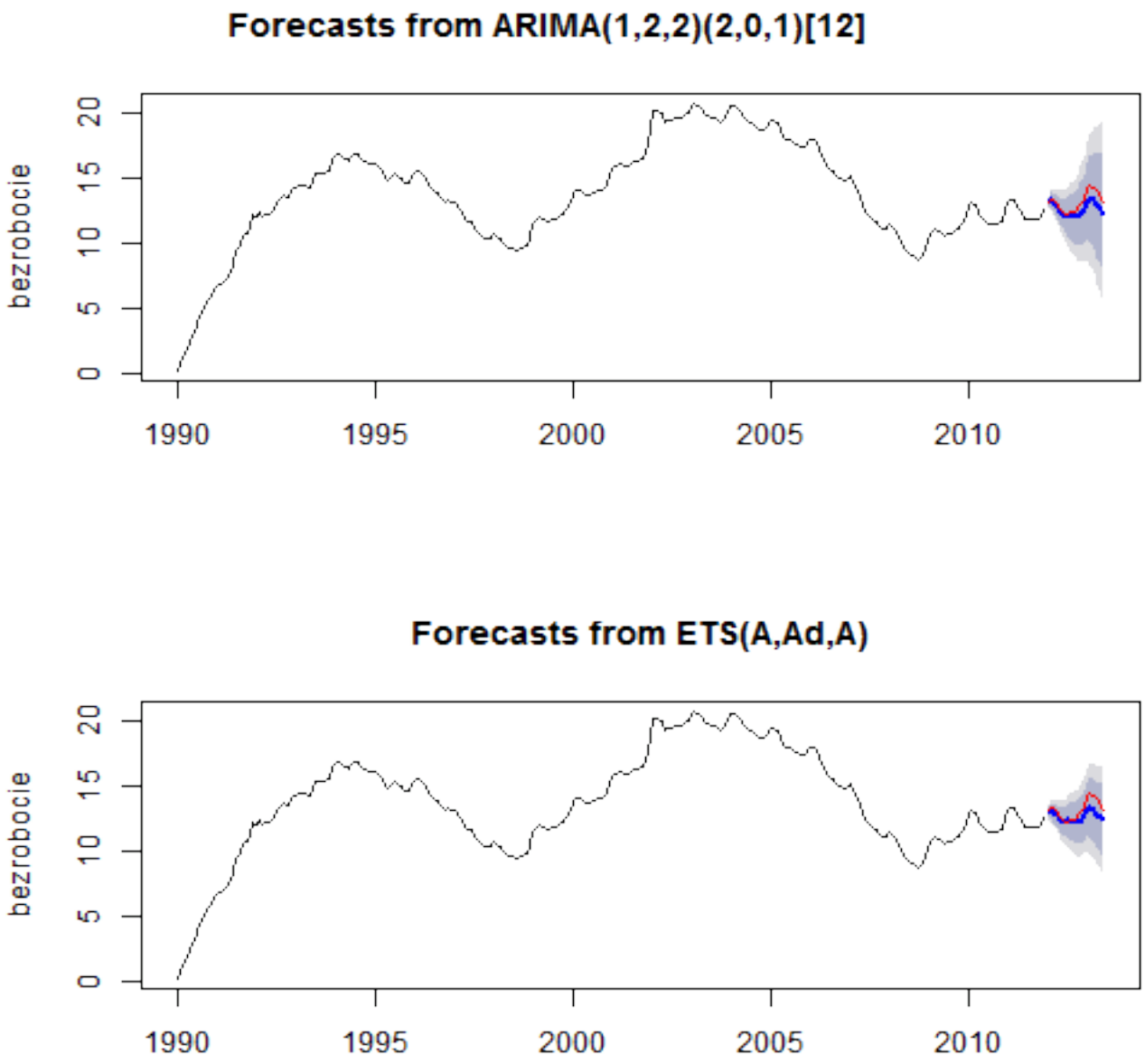
```
library(forecast)
par(mfrow = c(2, 1))

# porównanie dokładności prognoz: bezrobocie
plot(bezr.arima.forecast, ylab = "bezrobocie")
lines(bezrobocie.ts.test, col = "red")
plot(bezr.ets.forecast, ylab = "bezrobocie")
lines(bezrobocie.ts.test, col = "red")
```

```
accuracy(bezr.arima.forecast, bezrobocie.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1
## Training set 0.002761 0.2267 0.1416 0.07223 1.123 0.06979 0.02437
## Test set 0.556897 0.6721 0.5569 4.14115 4.141 0.27447 0.89223
## Theil's U
## Training set NA
## Test set 2.001
```

```
accuracy(bezr.ets.forecast, bezrobocie.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1
## Training set 0.0004695 0.2450 0.1606 -0.1228 1.949 0.07915 0.2344
## Test set 0.5429571 0.6769 0.5444 3.9909 4.002 0.26829 0.9073
## Theil's U
## Training set NA
## Test set 1.983
```



Rysunek 7: Prognozy dla szeregu 'bezrobocie': porównanie modeli ARIMA i ETS

Prognozy dla cen benzyny

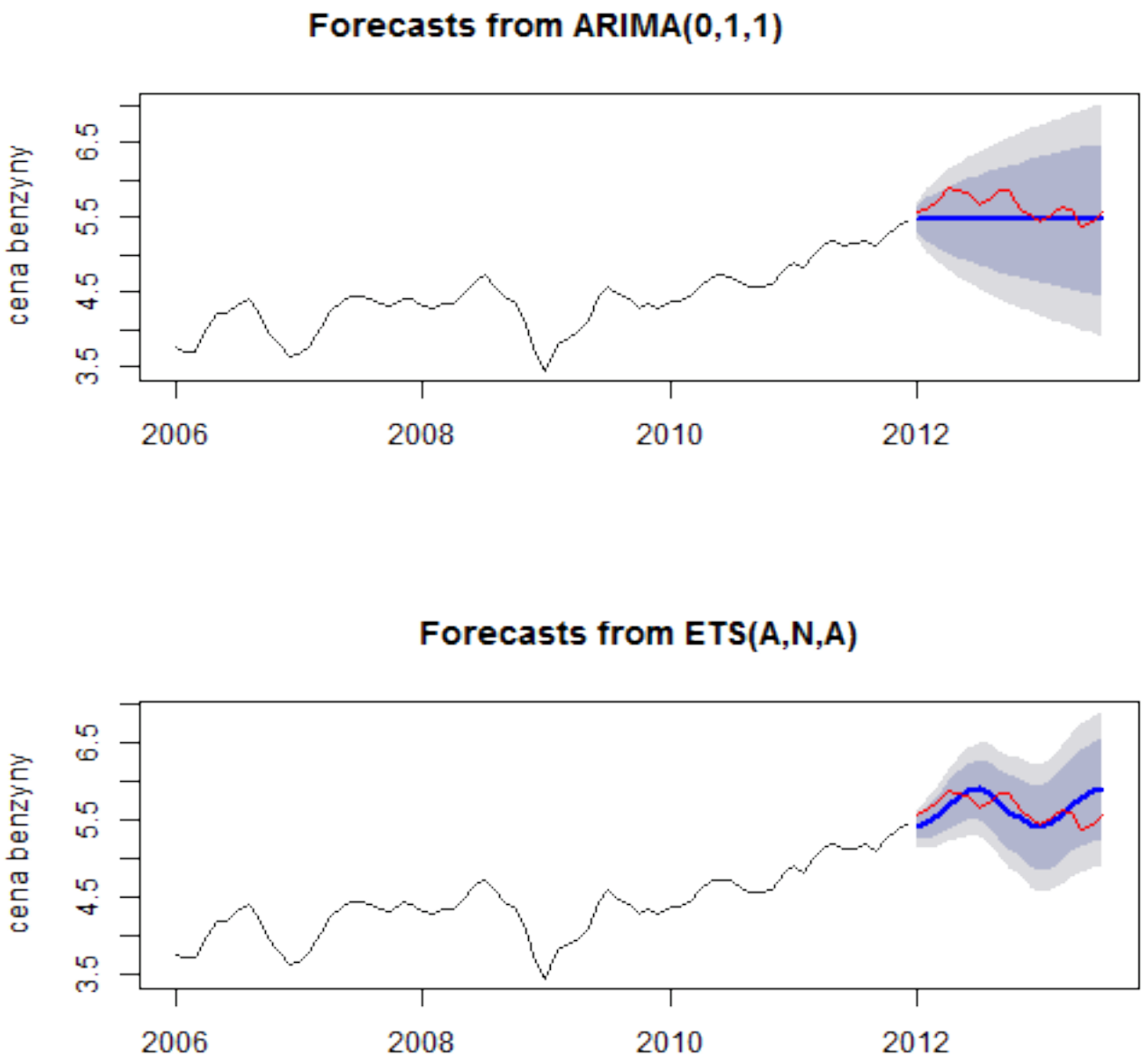
```
par(mfrow = c(2, 1))

# porównanie dokładności prognoz: cena benzyny
plot(benzyna.arima.forecast, ylab = "cena benzyny")
lines(cena.benzyny.ts.test, col = "red")
plot(benzyna.ets.forecast, ylab = "cena benzyny")
lines(cena.benzyny.ts.test, col = "red")
```

```
accuracy(benzyna.arima.forecast, cena.benzyny.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U
## Training set 0.0155 0.1202 0.09156 0.3197 2.137 0.2454 0.06566 NA
## Test set 0.1808 0.2357 0.19691 3.1321 3.430 0.5277 0.70062 2.013
```

```
accuracy(benzyna.ets.forecast, cena.benzyny.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1
## Training set 0.021162 0.1183 0.09376 0.3902 2.159 0.2512 0.3437
## Test set -0.006572 0.2013 0.16625 -0.1702 2.963 0.4455 0.6992
## Theil's U
## Training set NA
## Test set 1.75
```



Rysunek 8: Prognozy dla szeregu cen benzyny: porównanie modeli ARIMA i ETS

Prognozy dla kursu EUR/PLN

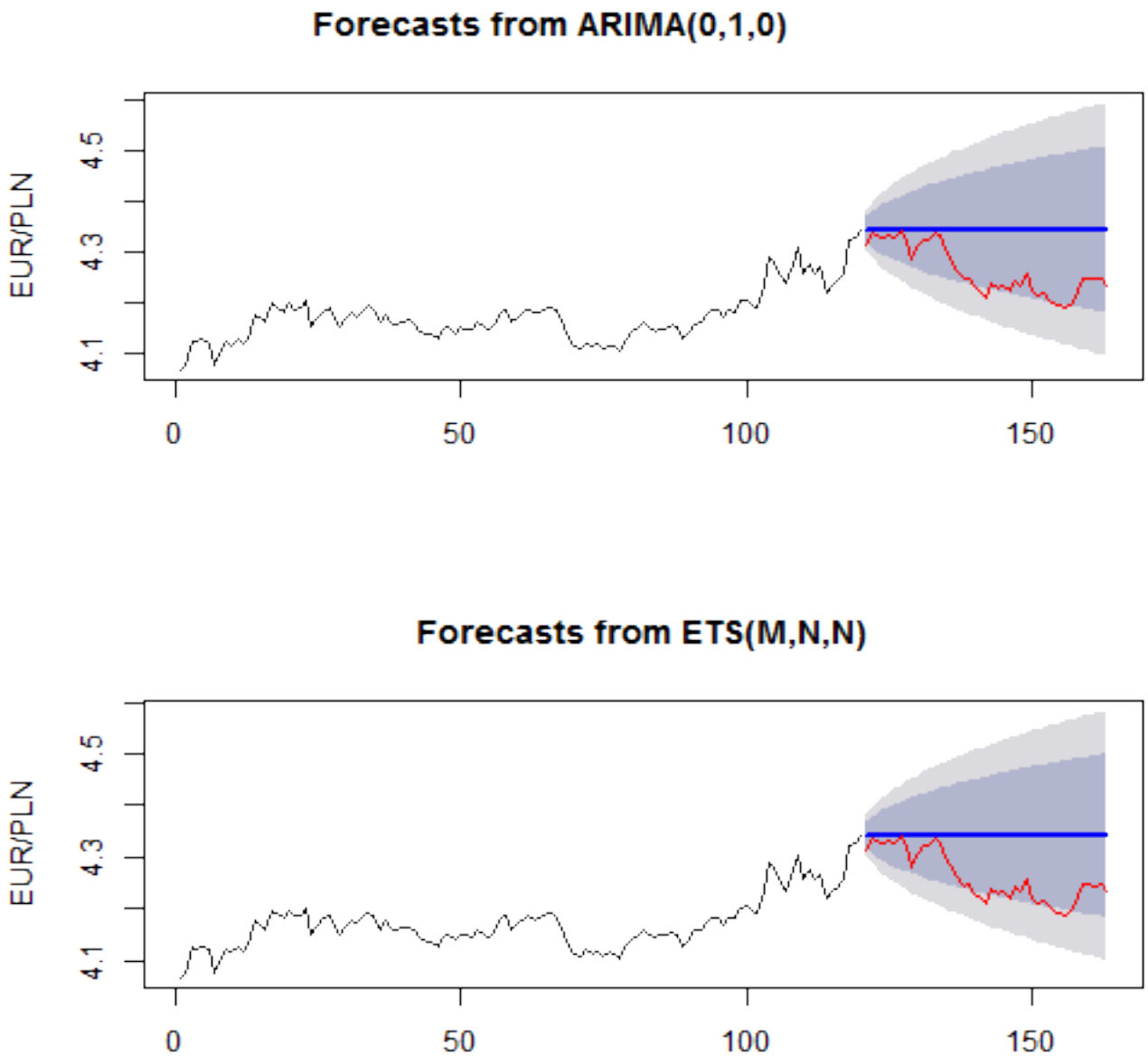
```
par(mfrow = c(2, 1))

# porównanie dokładności prognoz: kurs EUR/PLN
plot(EUR.arima.forecast, ylab = "EUR/PLN")
lines(kurs.EUR.PLN.ts.test, col = "red")
plot(EUR.ets.forecast, ylab = "EUR/PLN")
lines(kurs.EUR.PLN.ts.test, col = "red")
```

```
accuracy(EUR.arima.forecast, kurs.EUR.PLN.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1
## Training set 0.002335 0.01941 0.01443 0.0545 0.3448 0.994 -0.05958
## Test set -0.081935 0.09522 0.08193 -1.9359 1.9359 5.643 0.92503
## Theil's U
## Training set NA
## Test set 5.804
```

```
accuracy(EUR.ets.forecast, kurs.EUR.PLN.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1
## Training set 0.002437 0.01938 0.01428 0.05685 0.3411 0.9835 -0.008287
## Test set -0.080990 0.09441 0.08099 -1.91377 1.9138 5.5777 0.925025
## Theil's U
## Training set NA
## Test set 5.755
```



Rysunek 9: Prognozy dla kursu EUR/PLN: porównanie modeli ARIMA i ETS

Prognozy dla PKB

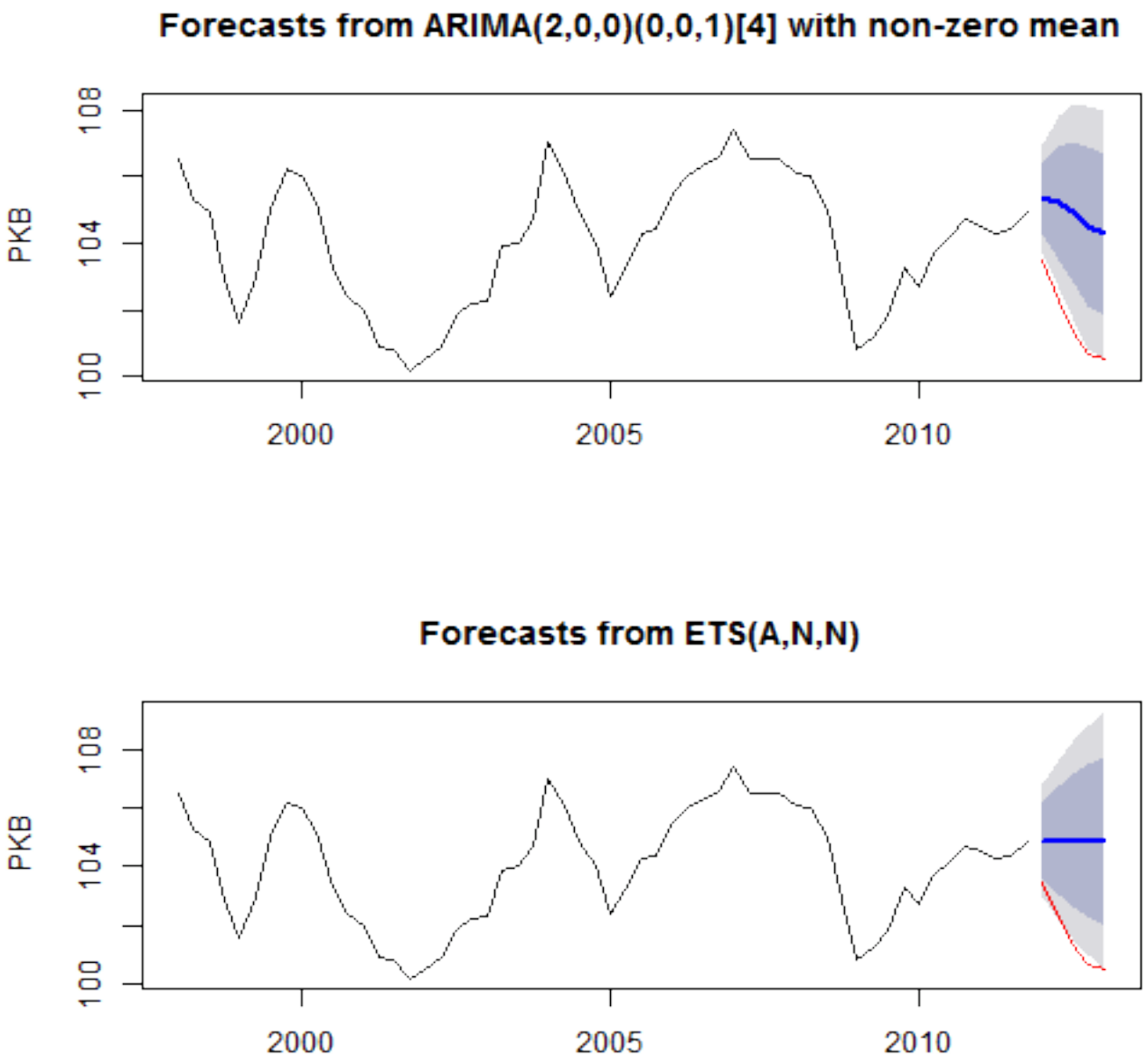
```
par(mfrow = c(2, 1))

# porównanie dokładności prognoz: PKB
plot(PKB.arima.forecast, ylab = "PKB")
lines(PKB.ts.test, col = "red")
plot(PKB.ets.forecast, ylab = "PKB")
lines(PKB.ts.test, col = "red")
```

```
accuracy(PKB.arima.forecast, PKB.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1
## Training set -0.01893 0.824 0.6856 -0.02487 0.6597 0.3209 -0.03805
## Test set -3.18451 3.270 3.1845 -3.14067 3.1407 1.4905 0.31001
## Theil's U
## Training set NA
## Test set 4.241
```

```
accuracy(PKB.ets.forecast, PKB.ts.test)
```

```
## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U
## Training set -0.02854 1.006 0.7965 -0.03169 0.7673 0.3728 0.3861 NA
## Test set -3.23995 3.426 3.2399 -3.19933 3.1993 1.5164 0.3886 4.521
```



Rysunek 10: Prognozy dla PKB: porównanie modeli ARIMA i ETS

5 Podsumowanie

- Zastosowanie narzędzi dostępnych w środowisku R (pakiet *forecast*) pozwoliło na automatyczną konstrukcję prognoz dla wybranych szeregów czasowych (zróżnicowanych pod względem własności).
- Dodatkowo, porównaliśmy dokładność prognoz wyznaczonych dwoma metodami: modele ARIMA oraz ETS (wygładzanie wykładnicze), wykorzystując najbardziej popularne kryteria (miary) oceny dokładności prognoz.
- W przypadku szeregów zawierających jednocześnie trend i sezonowość możliwa była także łatwa identyfikacja tych regularnych wzorców dzięki wykorzystaniu dostępnych w R metod dekompozycji.
- Prognozowanie na bazie wygładzania wykładniczego (modele ETS) dawało zazwyczaj lepsze prognozy w porównaniu do prognoz na bazie metodologii ARIMA. W niektórych przypadkach dokładność obu metod były jednak porównywalna.
- Zgodnie z oczekiwaniami, błędy prognoz na zbiorze testowym są większe w porównywaniu do błędów uzyskanych dla zbioru uczącego.
- Dla większości szeregów prognozy możemy uznać za satysfakcjonujące (przynajmniej w krótkim horyzoncie czasowym).
- Wyjątek stanowi szereg PKB, dla którego skonstruowane prognozy w znacznym stopniu odbiegają od wartości rzeczywistych. W tym przypadku warto byłoby przyjrzeć się dokładniej dopasowanym modelom, biorąc pod uwagę dodatkowe kryteria oceny jakości dopasowania.