

Dopasowanie modeli AR i porównania dokładności prognoz

Analiza szeregów czasowych

Ksawery Józefowski, 277513

2026-02-04

Spis treści

1	Dopasowanie modeli i prognozy szeregów	2
1.1	Wstęp	2
1.2	Test stacjonarności	2
1.3	Wybór optymalnego rzędu dla modelu $AR(p)$	6
1.4	Porównanie estymacji współczynników modeli $AR(p)$ metodą Yule’a-Walkera i największej wiarygodności	9
1.5	Ocena istotności współczynników	11
1.6	Ocena dopasowania modeli	12
1.7	Prognozowanie szeregu	17
1.8	Wnioski	19
2	Porównanie dokładności prognoz	20
2.1	Wstęp	20
2.2	Podział na zbiór testowy i uczący	20
2.3	Dopasowanie odpowiedzi modeli do danych	20
2.4	Wyznaczenie prognoz dla zbioru testowego	28
2.5	Wnioski	31

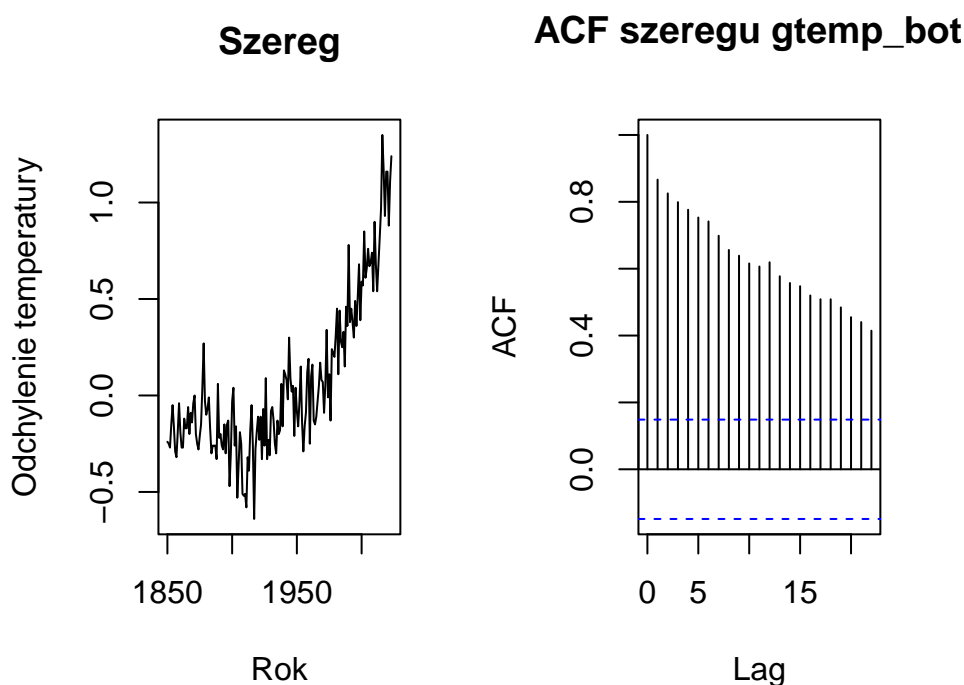
1 Dopasowanie modeli i prognozy szeregów

1.1 Wstęp

Celem tej części analizy jest dopasowanie modelu AR do szeregu `gtemp both`. Dane te przedstawiają odchylenia średniej rocznej temperatury (na powierzchni lądów i oceanów) w latach 1850-2023, mierzone względem średniej temperatury globalnej wyznaczonej dla okresu 1991-2020.

1.2 Test stacjonarności

W celu dopasowania modelu stacjonarnego do wybranego szeregu, musimy zbadać jego stacjonarność. Na obrazku 1 prezentujemy jak wygląda szereg oraz jego funkcja autokorelacji. Zauważalny jest silny trend wzrostowy. Sezonowość jest najprawdopodobniej nieobecna.



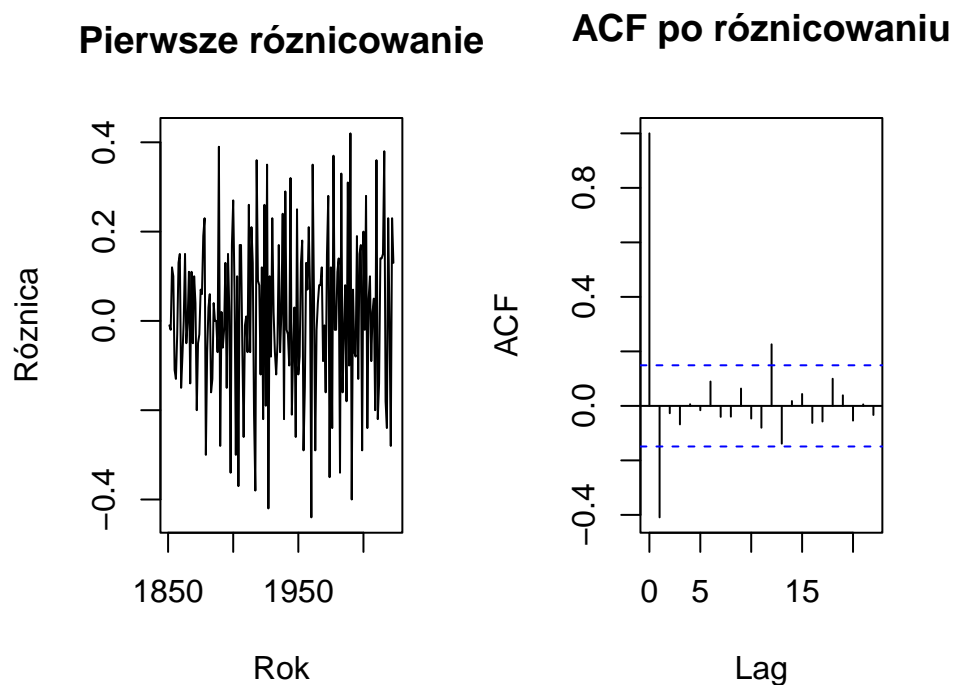
Rysunek 1: Wizualizacja szeregu oraz wykres ACF

Do sprowadzenia szeregu do postaci stacjonarnej posłużymy się dwoma metodami:

- Różnicowaniem
- Eliminacją trendu wielowymiarowego

Na danych wywołujemy dwie funkcje, `ndiffs` oraz `nsdiffs`. `ndiffs` stwierdza, że powinniśmy skorzystać z 1 krotnego różnicowania, a `nsdiffs` zwraca błąd potwierdzając tym samym, że w szeregu nie występuje sezonowość.

```
# ndiffs(dane) # 1 krotne roznicowanie
# nsdiffs(dane) # stwierdza ze nie ma sezonu
dane.diff.1 <- diff(dane)
```

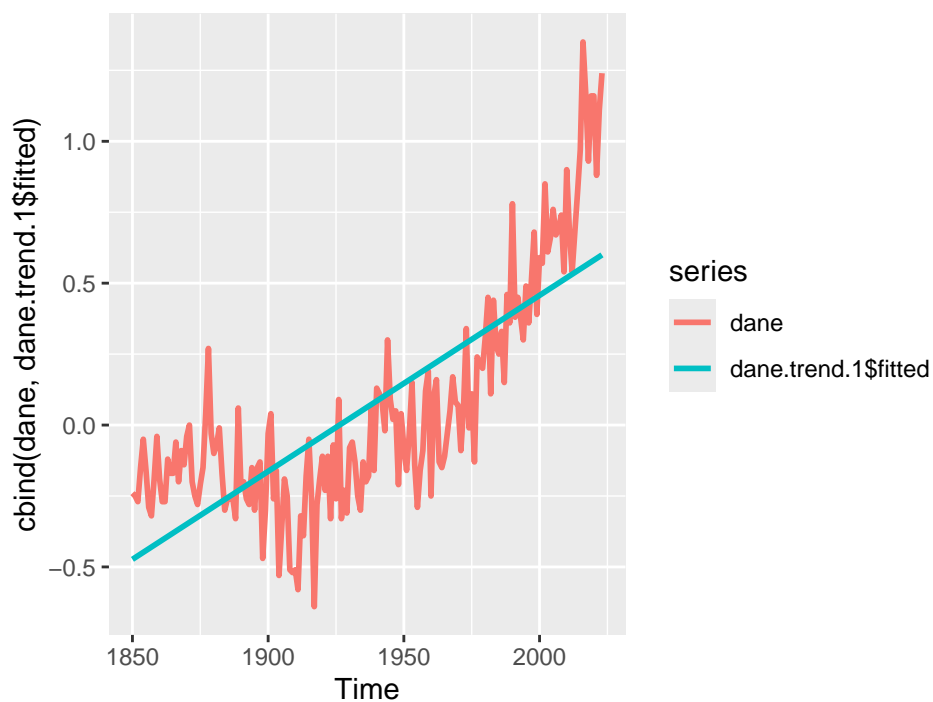


Rysunek 2: Wizualizacja szeregu oraz wykres ACF po różnicowaniu

Na obrazu 2 prezentujemy szereg i jego ACF po 1 krotnym różnicowaniu. Bazując na wykresach można stwierdzić, że szereg jest teraz stacjonarny.

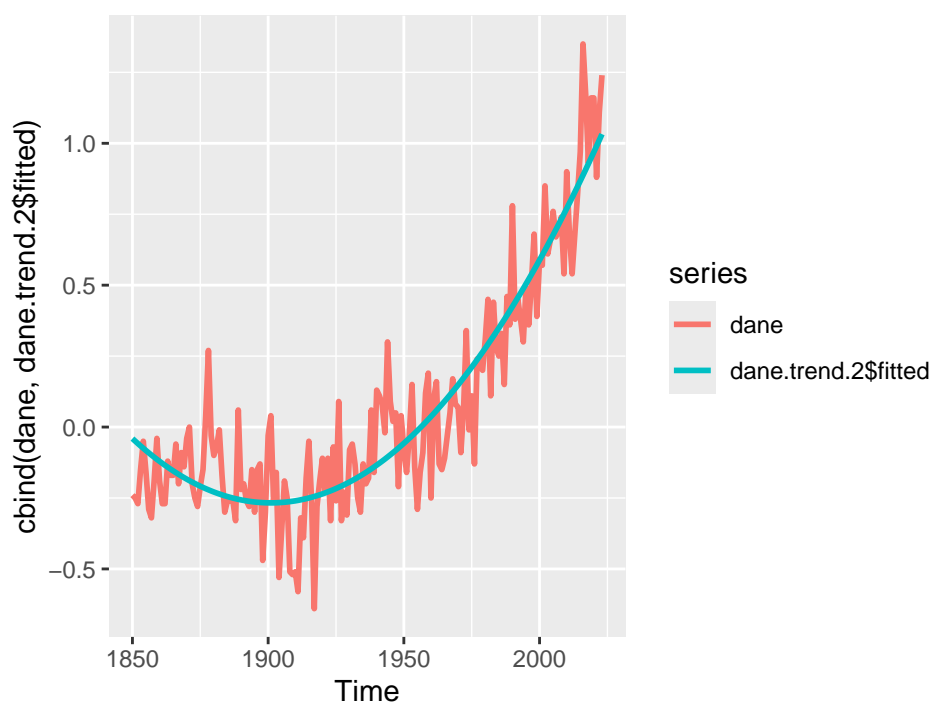
Następnie badamy trend wielowymiarowy. Z rysunku 3 wnioskujemy, że trend liniowy nie jest dobrym dopasowaniem do szeregu, tym samym zbadamy teraz trend kwadratowy.

```
dane.trend.1 <- tslm(dane ~ trend)
```



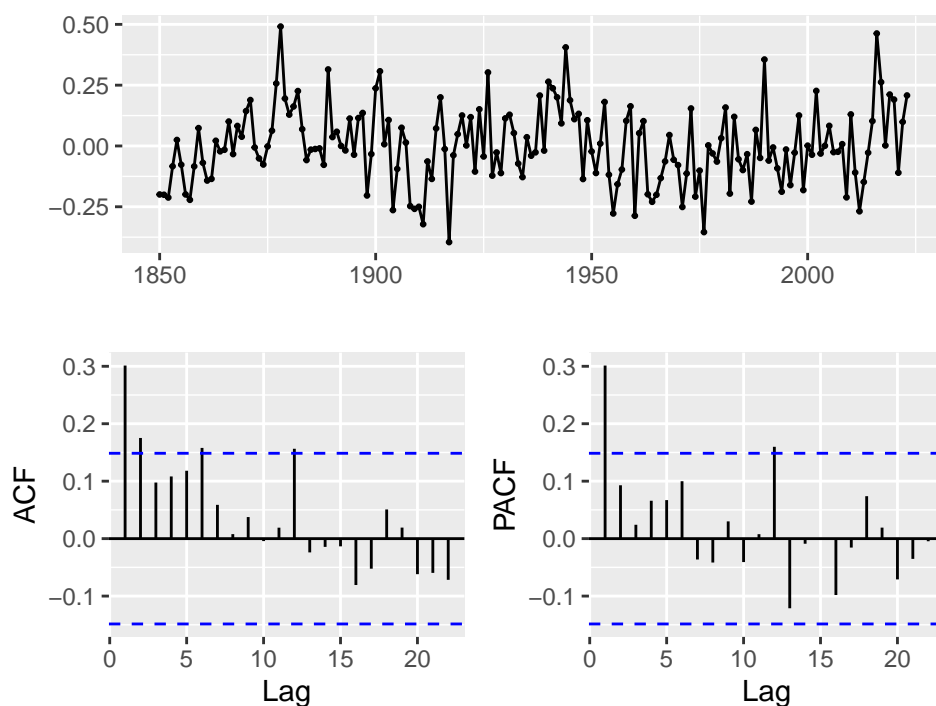
Rysunek 3: Estymacja szeregu po eliminacji trendu liniowego

```
dane.trend.2 <- tslm(dane ~ trend + I(trend^2))
# summary(dane.trend.2)
```



Rysunek 4: Estymacja szeregu po eliminacji trendu kwadratowego

Trend kwadratowy jak widać na obrazuk 4 estymuje idealnie nasz szereg, więc dalszą analizę będziemy kontynuować po jego eliminacji

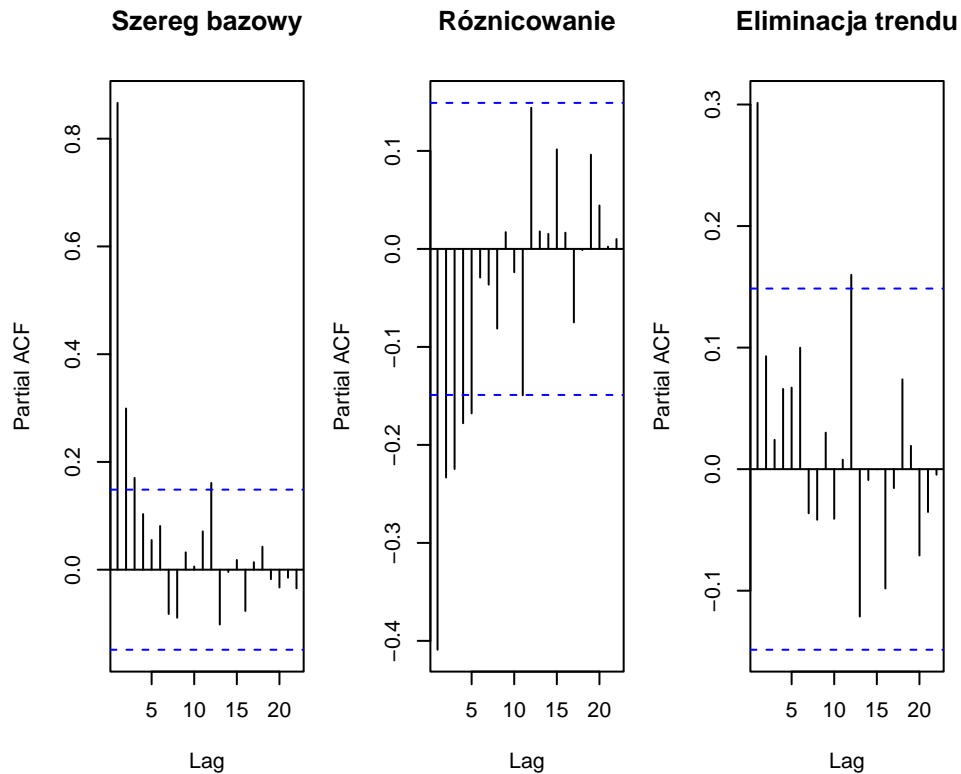


Rysunek 5: Szereg, ACF i PACF po eliminacji trendu kwadratowego

Jak widzimy na obrazku 5 eliminacja trendu kwadratowego pozwoliła na przybranie przez szereg postaci stacjonarnej.

1.3 Wybór optymalnego rzędu dla modelu $AR(p)$

Następnym krokiem analizy jest wybór optymalnego p . Do tego użyjemy wykresów PACF szeregów jak i również kryteriów informacyjnych AIC i FPE. Na rysunku 6 poniżej prezentujemy wykresy PACF dla szeregu bazowego i jego wersji stacjonarnych. Wstępnie możemy wywnioskować, że dla wersji po różnicowaniu optymalny parametr $p = 5$, a po eliminacji trendu $p = 12$.

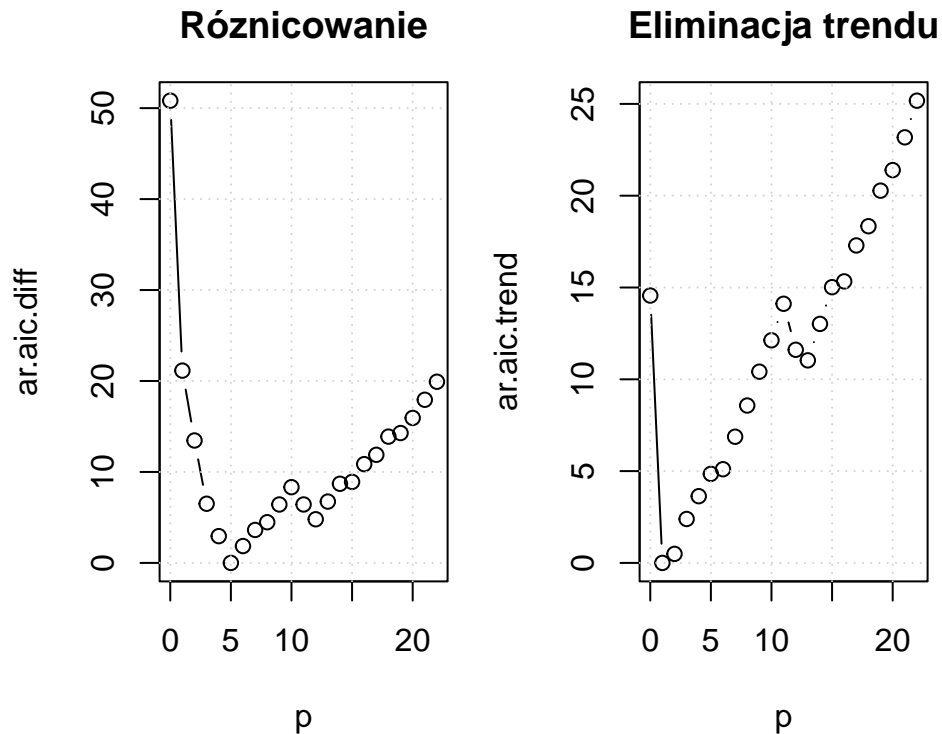


Rysunek 6: Wykresy PACF bazowego szeregu i jego postaci po transformacjach

Dla potwierdzenia naszych wniosków prezentujemy wykresy 7 kryterium AIC. Potwierdza ono, że $p = 5$ jest odpowiednim parametrem dla szeregu po różnicowaniu, jednakże dla wersji z eliminacją trendu stwierdza, że powinniśmy wybrać $p = 1$.

```
n <- length(dane)
p.max.AIC <- floor(10*log10(n))

ar.aic.diff <- ar(dane.diff.1, aic = TRUE)$aic
ar.aic.trend <- ar(dane.reszty.2, aic = TRUE)$aic
```



Rysunek 7: Wykresy AIC dla szeregu po różnicowaniu i eliminacji trendu

Dla pewności dodatkowo zbadamy kryterium FPE. Na rysunku 8 prezentujemy wykresy kryterium FPE. Po tym badaniu jesteśmy jednoznacznie stwierdzić, że szereg po różnicowaniu powinien za parametr przyjąć $p = 5$, a szereg po eliminacji trendu $p = 1$.

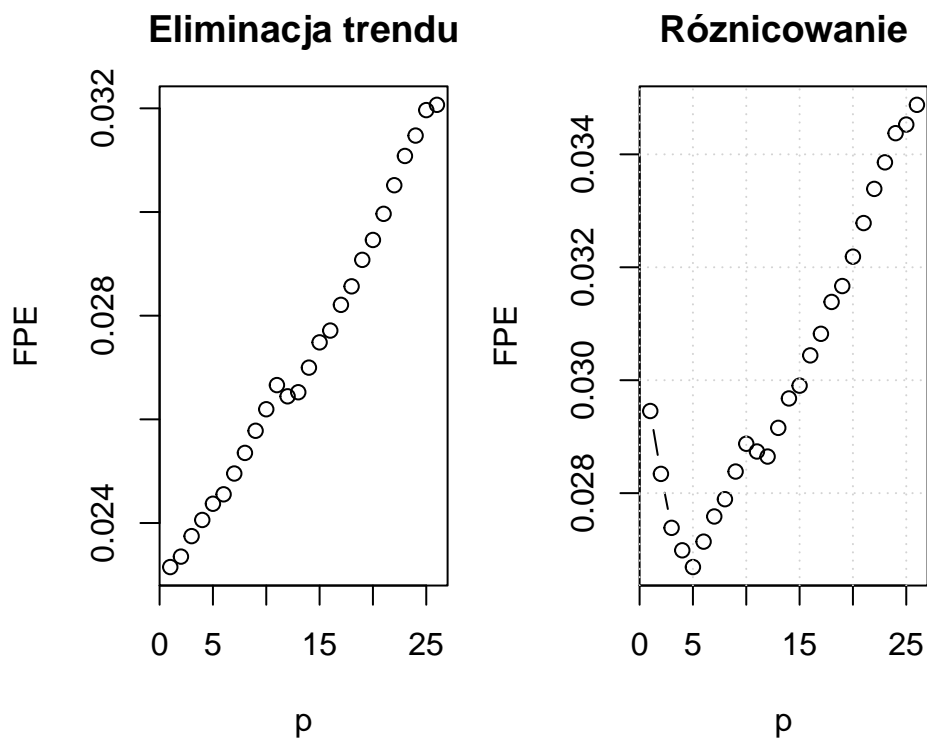
```
dane.reszty <- dane.reszty.2
N <- length(dane.reszty)
p.max.FPE <- floor(2 * sqrt(N))
fpe.tslm <- numeric(p.max.FPE)

for (p in 1:p.max.FPE) {
  c.MN <- (N+p) / (N-p)
  fpe.value <- ar(dane.reszty, aic=FALSE, method="yw", order.max=p)$var.pred
  fpe.tslm[p] <- c.MN * fpe.value
}

reszty_diff <- dane.diff.1
N <- length(reszty_diff)
p.max.FPE <- floor(2 * sqrt(N))
fpe.diff <- numeric(p.max.FPE)

for (p in 1:p.max.FPE) {
  c.MN <- (N + p) / (N - p)
```

```
fpe.diff[p] <- ar(reszty_diff, aic = FALSE, method = "yw", order.max = p)$var.pred * c
}
```



Rysunek 8: Wykresy FPE dla szeregu po różnicowaniu i eliminacji trendu

1.4 Porównanie estymacji współczynników modeli AR(p) metodą Yule'a-Walkera i największej wiarygodności

Dla obu przekształceń dopasujemy modele AR(p) o wybranym rzędzie ($p = 5$ dla szeregu różnicowanego oraz $p = 1$ dla reszt po eliminacji trendu), stosując dwie metody estymacji współczynników:

- Yule'a-Walkera - metoda estymacji wstępnej oparta na układzie równań normalnych.
- Największej wiarygodności - estymacja oparta na maksymalizacji funkcji wiarygodności.

Przeprowadzimy następujące operacje dla obu przekształceń:

- Dopasowanie modelu AR(p) metodą YW i MLE.
- Wyznaczenie współczynników modelu (`coef.diff.yw`, `coef.diff.mle`, `coef.trend.yw`, `coef.trend.mle`).
- Oszacowanie wariancji składnika losowego (`var.wn.diff.yw`, `var.wn.diff.mle`, `var.wn.trend.yw`, `var.wn.trend.mle`).

- Porównanie współczynników i wariancji dla obu metod.

Wyniki pokazują, że współczynniki i wariancje są podobne niezależnie od zastosowanej metody estymacji, zarówno dla szeregu różnicowanego, jak i dla reszt po eliminacji trendu, co potwierdza spójność estymacji metodą Yule'a-Walkera i MLE.

```
ar.fit.diff.yw <- ar(dane.diff.1, order.max=5, aic=FALSE, method="yule-walker")
coef.diff.yw <- ar.fit.diff.yw$ar
var.wn.diff.yw <- ar.fit.diff.yw$var.pred
cov.diff.yw <- ar.fit.diff.yw$asy.var.coef
order.diff.yw <- ar.fit.diff.yw$order

ar.fit.diff.mle <- ar(dane.diff.1, order.max=5, aic=FALSE, method="mle")
coef.diff.mle <- ar.fit.diff.mle$ar
var.wn.diff.mle <- ar.fit.diff.mle$var.pred
order.diff.mle <- ar.fit.diff.mle$order

ar.fit.trend.yw <- ar(dane.reszty.2, order.max = 1, aic = FALSE, method = "yule-walker")
coef.trend.yw <- ar.fit.trend.yw$ar
var.wn.trend.yw <- ar.fit.trend.yw$var.pred
cov.trend.yw <- ar.fit.trend.yw$asy.var.coef
order.trend.yw <- ar.fit.trend.yw$order

ar.fit.trend.mle <- ar(dane.reszty.2, order.max = 1, aic = FALSE, method = "mle")
coef.trend.mle <- ar.fit.trend.mle$ar
var.wn.trend.mle <- ar.fit.trend.mle$var.pred
order.trend.mle <- ar.fit.trend.mle$order

## YW dla eliminacji trendu
## [1] 0.3013202
## [1] 0.02288595
## MLE dla eliminacji trendu
##      ar1
## 0.3054151
## [1] 0.02257886
## YW dla różnicowania
## [1] -0.6266831 -0.4625600 -0.3921596 -0.2780332 -0.1678407
## [1] 0.02519167
## MLE dla różnicowania
##      ar1      ar2      ar3      ar4      ar5
## -0.6311944 -0.4698950 -0.3939761 -0.2812103 -0.1719101
```

```
## [1] 0.02416992
```

Porównując wyniki estymacji współczynników modeli AR(p) metodą Yule’a-Walkera oraz największej wiarygodności, możemy stwierdzić, że dla obu rodzajów przekształceń danych, współczynniki są bardzo zbliżone, podobnie jak wariancje składnika losowego. Dla szeregu różnicowanego współczynniki AR są ujemne i malejące w kolejnych lagach, co wskazuje na silną autokorelację krótkookresową, natomiast dla reszt po eliminacji trendu współczynnik AR(1) jest dodatni i znacznie mniejszy, sugerując słabszą dynamikę krótkookresową. Wariancja składnika losowego dla szeregu różnicowanego jest nieco większa niż dla reszt po eliminacji trendu, co jest zgodne z większą zmiennością pierwszych różnic w porównaniu do zdekorelowanych reszt. Ogólnie można stwierdzić, że modele AR(p) zostały dobrze dopasowane, a wybór metody estymacji nie wpływa istotnie na wartości oszacowanych parametrów ani na wariancję reszt dlatego, też analizę będziemy kontynuować przy użyciu estymacji metodą Yule’a-Walkera.

1.5 Ocena istotności współczynników

W celu oceny statystycznej istotności współczynników wybranego modelu AR(p) zastosujemy twierdzenie o asymptotycznej normalności estymatorów. Dla każdej wartości parametru θ_i skonstruujemy asymptotyczny 95% przedział ufności według wzoru $\hat{\theta}_i \pm z_{1-\alpha/2} \cdot \text{SE}(\hat{\theta}_i)$, gdzie $\hat{\theta}_i$ oznacza estymator współczynnika, a $\text{SE}(\hat{\theta}_i)$ jego błąd standardowy.

Najpierw przeanalizuję szereg różnicowany przy użyciu metody Yule’a-Walkera. Wyznaczone przedziały ufności pozwolą sprawdzić hipotezę zerową $H_0 : \theta_i = 0$ przeciwko alternatywie $H_1 : \theta_i \neq 0$. Współczynniki, dla których przedział ufności zawiera zero, uznamy za nieistotne statystycznie i potencjalnie możliwe do pominięcia w modelu. Analogicznie, przeprowadzimy analizę dla reszt po eliminacji trendu wielomianowego, również z wykorzystaniem estymacji Yule’a-Walkera.

```
model <- ar.fit.diff.yw
hat.theta <- model$ar
se <- sqrt(diag(model$asy.var.coef))
TL_1 <- hat.theta - ( qnorm(1-0.05/2) * se )
TU_1 <- hat.theta + ( qnorm(1-0.05/2) * se )

model <- ar.fit.trend.yw
hat.theta <- model$ar
se <- sqrt(diag(model$asy.var.coef))
TL_2 <- hat.theta - (qnorm(1 - 0.05/2) * se)
TU_2 <- hat.theta + (qnorm(1 - 0.05/2) * se)
```

```
## Różnicowanie
```

```
## [1] -0.7761981 -0.6346370 -0.5682125 -0.4501102 -0.3173557
```

```
## [1] -0.47716803 -0.29048301 -0.21610675 -0.10595627 -0.01832563
```

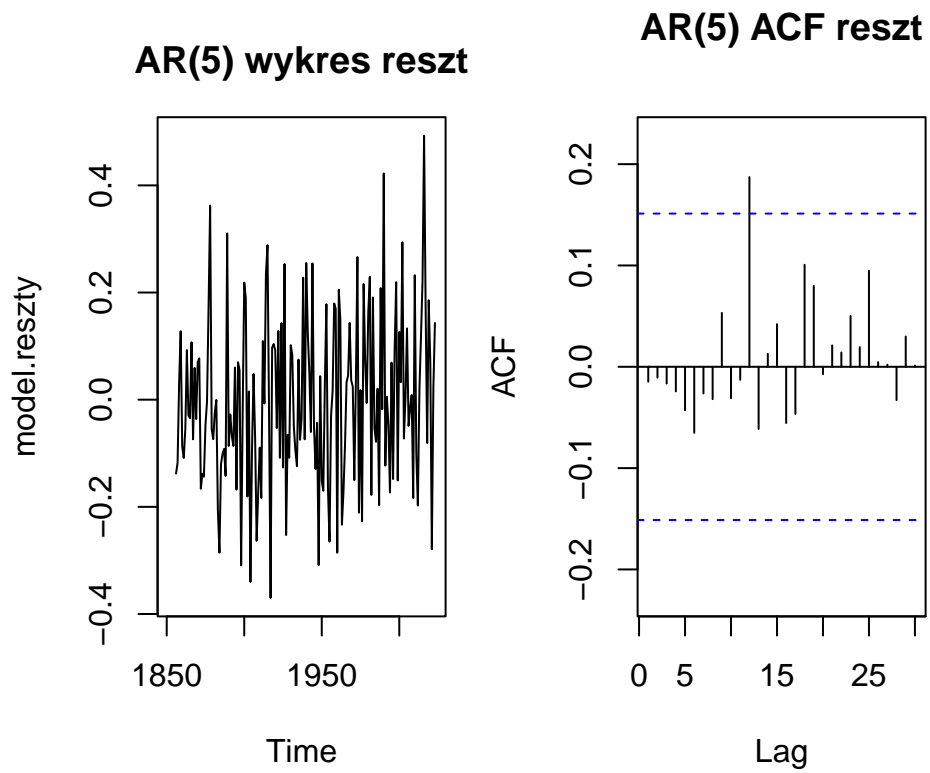
```
## integer(0)
```

```
##  
## Eliminacja trendu  
## [1] 0.1588201  
## [1] 0.4438202  
## integer(0)
```

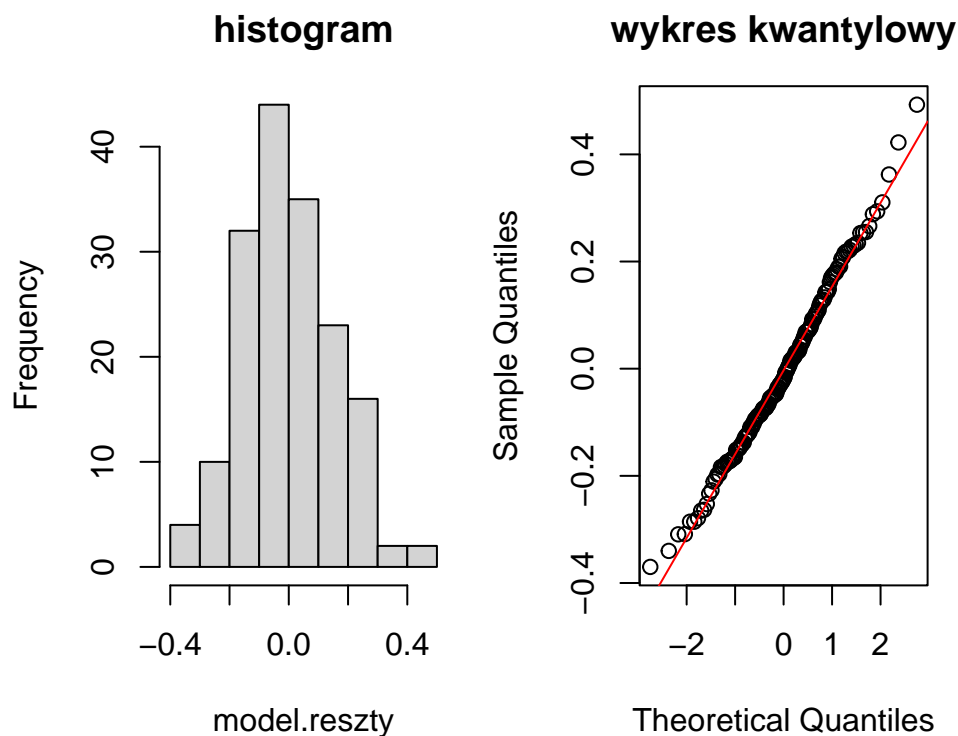
Analiza przedziałów ufności dla parametrów modelu $AR(p)$ wykazała, że dla szeregu różnicowanego wszystkie współczynniki mają przedziały ufności, które nie obejmują zera, co oznacza, że wszystkie są statystycznie istotne. Oznacza to, że model $AR(5)$ dla szeregu różnicowanego dobrze opisuje krótkookresową dynamikę szeregu. W przypadku reszt po eliminacji trendu współczynnik $AR(1)$ również okazał się istotny statystycznie, a w modelu nie stwierdzamy żadnych współczynników nieistotnych. Ogólnie możemy stwierdzić, że wybrane modele $AR(p)$ są statystycznie uzasadnione, a oszacowane współczynniki powinny zostać uwzględnione w modelu.

1.6 Ocena dopasowania modeli

W celu oceny dopasowania wybranych modeli $AR(p)$ przeprowadzimy analizę reszt. Rozważymy dwa modele: $AR(5)$ dla szeregu różnicowanego oraz $AR(1)$ dla reszt po eliminacji trendu. Analiza obejmie wykresy reszt, funkcję autokorelacji, testy losowości Ljung-Box'a oraz sprawdzenie normalności rozkładu reszt za pomocą histogramu, wykresu kwantylowego oraz testu Shapiro-Wilka. Celem jest weryfikacja, czy reszty zachowują się jak biały szum, co jest warunkiem poprawnego dopasowania modelu autoregresyjnego.



Rysunek 9: Wykresy reszt i ich ACF dla AR(5)



Rysunek 10: Histogram i wykres kwantylowy reszt dla AR(5)

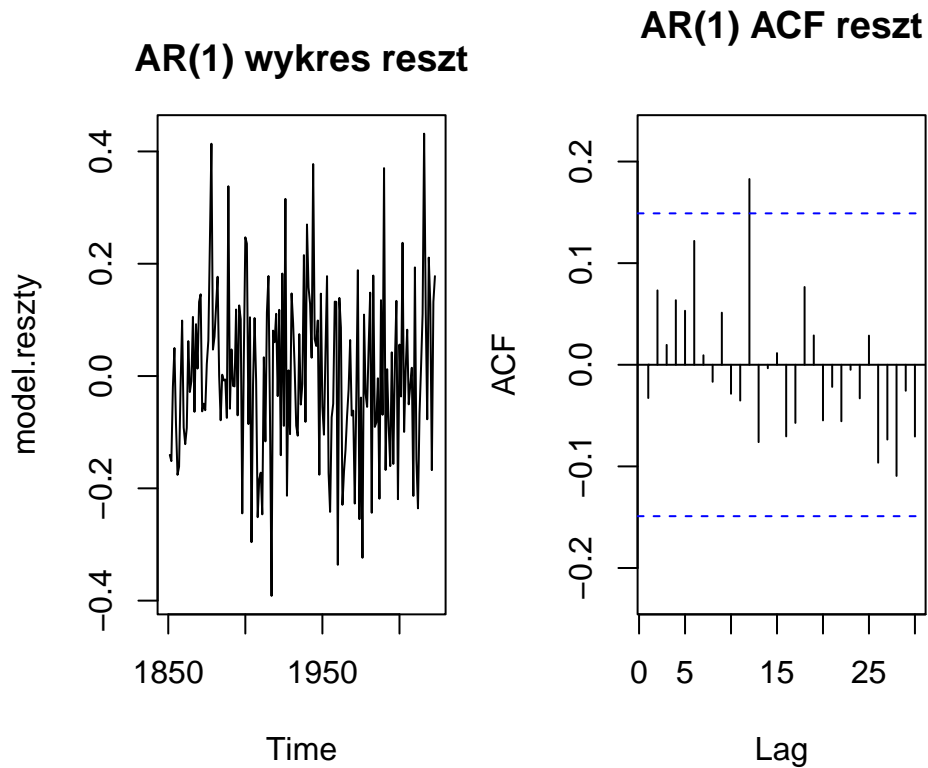
```
##
## Box-Ljung test
##
## data: model.reszty
## X-squared = 0.037001, df = 1, p-value = 0.8475

##
## Box-Ljung test
##
## data: model.reszty
## X-squared = 8.7111, df = 7, p-value = 0.2741

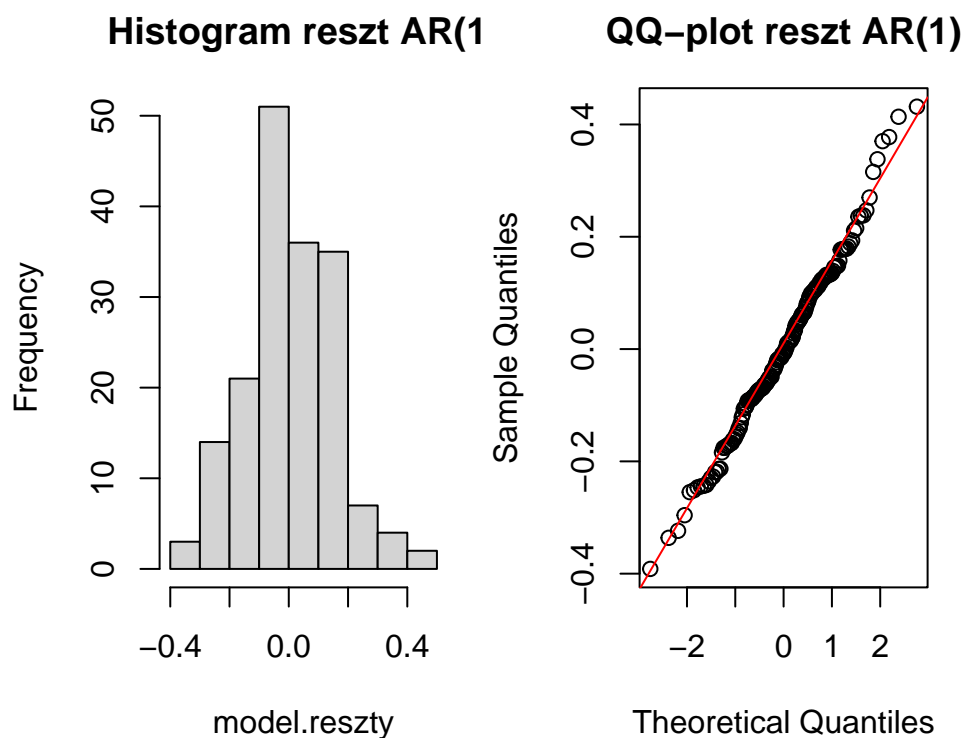
##
## Shapiro-Wilk normality test
##
## data: model.reszty
## W = 0.9923, p-value = 0.5108
```

Przeprowadziliśmy szczegółową analizę reszt modelu AR(5) dopasowanego do szeregu różnicowanego. Wykresy reszt 9 nie wykazują żadnych regularnych wzorców, co sugeruje brak niepożądanych zależności wizualnych. Funkcja autokorelacji pokazuje jedynie pojedynczą niewielką autokorelację przy lag = 12, która nie wydaje się istotna. Test Ljunga-Boxa dla lag = 1 oraz lag = 12 nie odrzuca hipotezy o losowości reszt, co potwierdza brak istotnej

autokorelacji. Test Shapiro-Wilka jak i wykresy rozkładu 10 wskazują również, że rozkład reszt jest zbliżony do normalnego. Wnioskujemy z tego, że model AR(5) dobrze dopasowuje dane po różnicowaniu, a reszty zachowują się jak biały szum.



Rysunek 11: Wykresy reszt i ich ACF dla AR(1)



Rysunek 12: Histogram i wykres kwantylowy reszt dla AR(1)

```
##
## Box-Ljung test
##
## data: model.reszty
## X-squared = 0.18853, df = 1, p-value = 0.6641
##
## Box-Ljung test
##
## data: model.reszty
## X-squared = 12.347, df = 11, p-value = 0.3381
##
## Shapiro-Wilk normality test
##
## data: model.reszty
## W = 0.99237, p-value = 0.4962
```

Dla modelu AR(1) dopasowanego do reszt po eliminacji trendu obserwujemy podobne zależności. Wykres reszt 11 nie pokazuje regularnych wzorców, a ACF nie ujawnia istotnych autokorelacji. Testy Ljunga-Boxa dla różnych lagów (lag = 1 oraz lag = 12) wskazują brak istotnej autokorelacji. Rozkład reszt na wykresach 12 jest również zbliżony do normalnego, co potwierdza test Shapiro-Wilka. Możemy więc stwierdzić, że model AR(1) dla reszt po

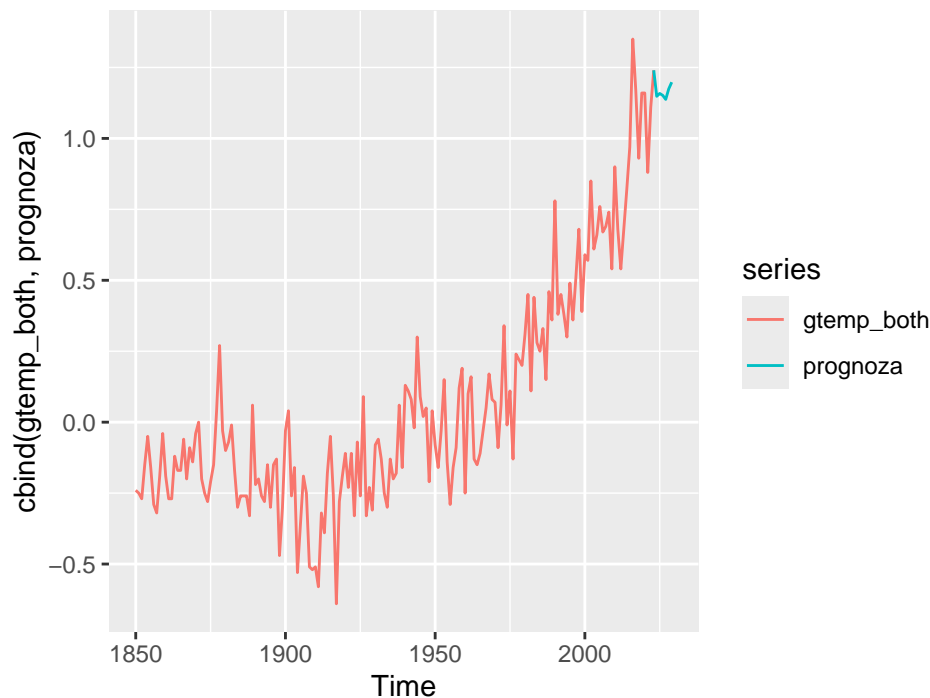
eliminacji trendu jest dobrze dopasowany, a reszty zachowują się jak biały szum, co umożliwia wiarygodną interpretację i prognozowanie szeregu.

1.7 Prognozowanie szeregu

W kolejnym etapie przeprowadzimy prognozowanie wartości szeregu temperatur globalnych na podstawie wybranych modeli AR(p). Dla modelu AR(5) dopasowanego do szeregu różnicowanego (wybranego na podstawie PACF, AIC oraz FPE i oszacowanego metodą Yule'a-Walkera) skonstruowano prognozy na 6 okresów do przodu. Prognozy te zostały następnie przekształcone do poziomu oryginalnego szeregu poprzez odwrócenie różnicowania.

Dodatkowo wykonaliśmy prognozę dla modelu AR(1) dopasowanego do reszt po eliminacji trendu deterministycznego. W tym przypadku prognoza została skonstruowana jako suma przewidywanego trendu oraz prognozowanych reszt, co pozwala uzyskać prognozy w pierwotnej skali szeregu. Otrzymane wartości prognoz przedstawiono na wykresach wraz z danymi historycznymi.

```
model <- ar.fit.diff.yw
model.prognoza <- predict(model, n.ahead = 6)$pred
prognoza <- diffinv(model.prognoza, lag=1, xi = dane[n])
```

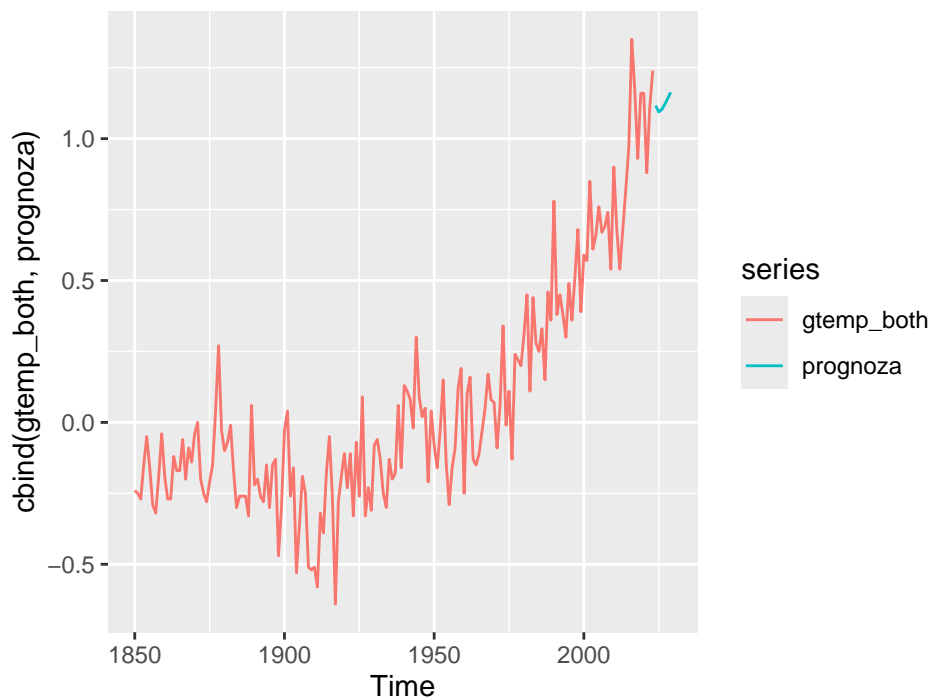


Rysunek 13: Bazowy szereg + jego prognoza dla AR(5)

```

reszty.prog <- predict(ar.fit.trend.yw, n.ahead = 6)$pred
T <- length(dane)
t.future <- (T+1):(T+6)
trend.prog <- predict(dane.trend.2,
                      newdata = data.frame(trend = t.future))
prognoza <- trend.prog + reszty.prog

```



Rysunek 14: Bazowy szereg + jego prognoza dla AR(1)

Analizując prognozy obu modeli AR(p) na wykresach 13 i 14, zauważamy, że model AR(5) dopasowany do szeregu różnicowanego generuje prognozy, które dobrze odzwierciedlają poziom i zmienność oryginalnego szeregu temperatur. W przypadku modelu AR(1) opartego na resztach po eliminacji trendu prognozy nieco odstają od obserwowanego szeregu, co sugeruje, że model ten lepiej nadaje się do uchwycenia ogólnych trendów niż do dokładnego odtwarzania krótkookresowych fluktuacji.

1.8 Wnioski

Podsumowując całą analizę, przeprowadziliśmy pełną procedurę dopasowania modeli autoregresyjnych $AR(p)$ do danych. Dokonaliśmy transformacji szeregu poprzez różnicowanie oraz eliminację trendu, wybraliśmy rzędy modeli na podstawie PACF, AIC i FPE, oszacowaliśmy parametry metodami Yule'a-Walkera i MLE, a następnie oceniliśmy istotność współczynników za pomocą asymptotycznych przedziałów ufności. Analiza reszt, testy Ljunga-Boxa oraz testy normalności potwierdziły poprawność dopasowania modeli. Prognozy wykazały, że model $AR(p)$ po różnicowaniu najlepiej odtwarza krótkookresowe wahania szeregu, natomiast model po eliminacji trendu dobrze uchwycił główny kierunek zmian, ale mniej precyzyjnie oddaje poziom szeregu. Wnioski te sugerują, że w praktyce warto uwzględniać zarówno transformacje różnicowe, jak i trendowe, w zależności od celu analizy.

2 Porównanie dokładności prognoz

2.1 Wstęp

Głównym celem zadania jest porównanie dokładności prognoz skonstruowanych na bazie modeli (S)ARIMA, modeli dekompozycji oraz algorytmów wygładzania wykładniczego dla danych euretail.

2.2 Podział na zbiór testowy i uczący

W celu oceny jakości prognoz szereg czasowy `euretail` podzielimy na zbiór uczący oraz zbiór testowy. Zbiór uczący obejmuje początkową część szeregu, natomiast zbiór testowy składa się z ostatnich 16 obserwacji, które posłużą do weryfikacji dokładności prognoz. Takie podejście pozwala na ocenę modeli w sposób odzwierciedlający rzeczywiste prognozowanie przyszłych wartości szeregu.

```
data(euretail)

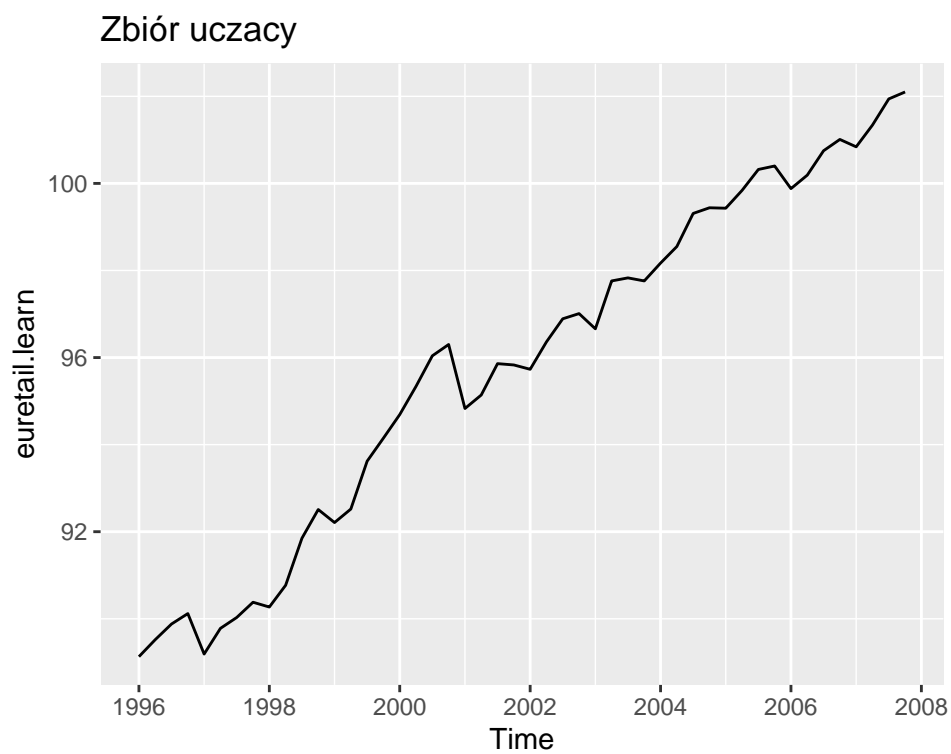
n_test <- 16
euretail.learn <- window(euretail, end =
                          time(euretail)[length(euretail) - n_test])
euretail.test <- window(euretail, start =
                        time(euretail)[length(euretail) - n_test + 1])
h <- length(euretail.test)
```

2.3 Dopasowanie odpowiedniach modeli do danych

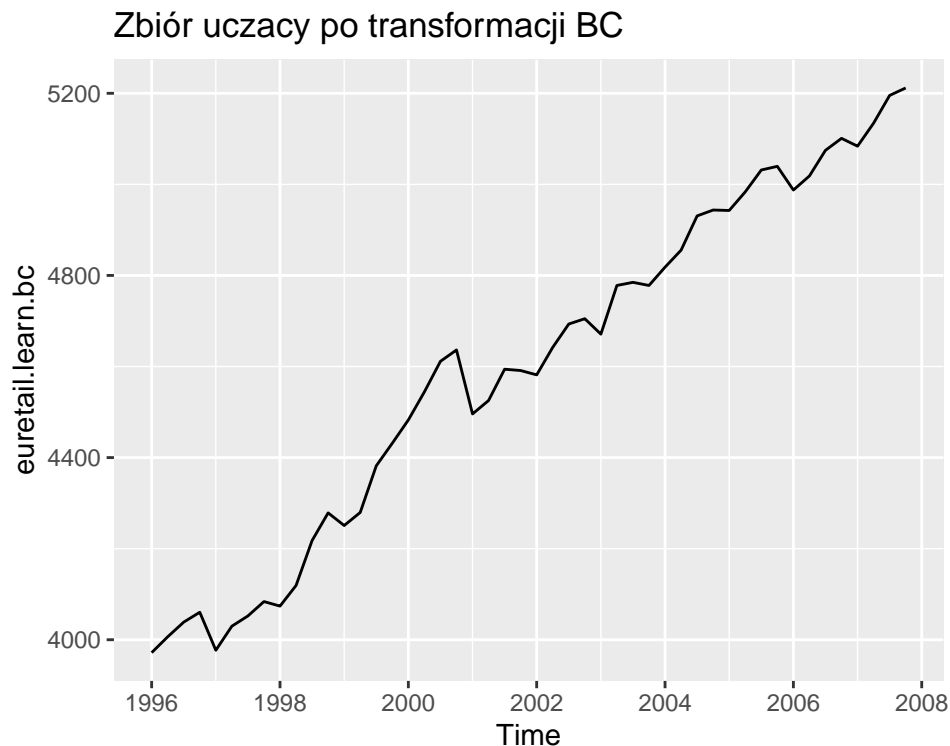
W niniejszej części pracy przeprowadzimy identyfikację oraz dopasowanie różnych klas modeli szeregów czasowych dla danych `euretail`. Analiza obejmie wstępne przekształcenia danych, badanie stacjonarności, identyfikację modeli ARIMA, modeli trendu oraz algorytmów wygładzania wykładniczego. Celem jest wybór modeli najlepiej opisujących dynamikę badanego szeregu i nadających się do dalszego prognozowania.

Na start zrobimy wizualną analizę szeregu czasowego oraz zastosujemy transformację Boxa-Coxa w celu stabilizacji wariancji.

```
par(mfrow = c(1,2), mar=c(4,4,3,1))
lambda <- BoxCox.lambda(euretail.learn, method = "loglik")
euretail.learn.bc <- BoxCox(euretail.learn, lambda = lambda)
```



Rysunek 15: Bazowy zbiór uczący i jego wersja po transformacji Boxa-Coxa



Rysunek 16: Bazowy zbiór uczący i jego wersja po transformacji Boxa-Coxa

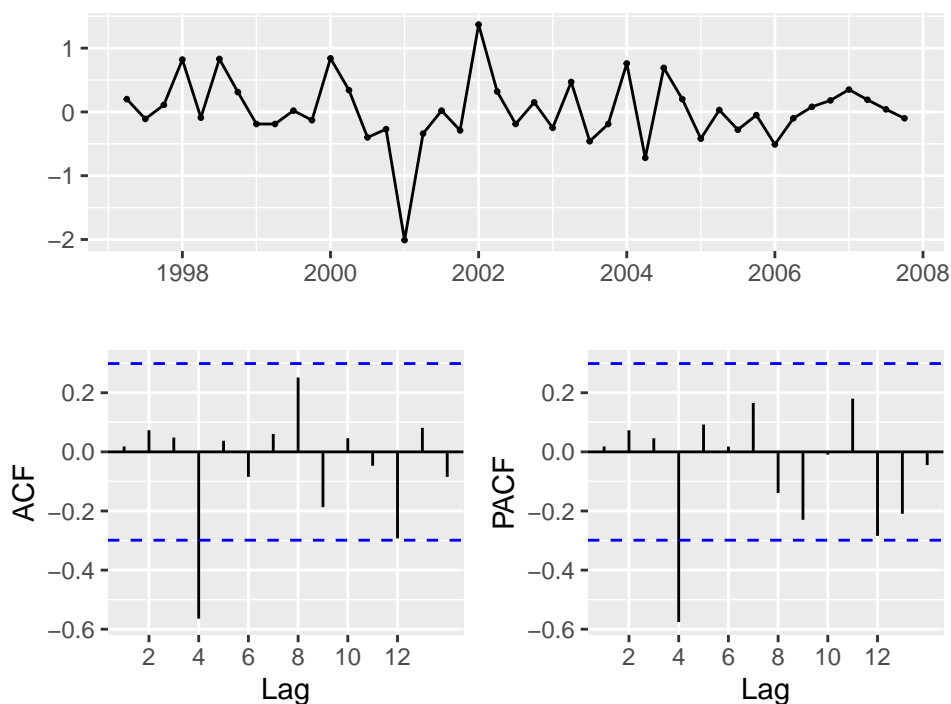
Na obrazku 16 możemy zauważyć, że transformacja Boxa-Coxa nic nie wnosi, więc nie będziemy jej używać w dalszej części.

Następnie sprawdzimy stopień niestacjonarności szeregu przy użyciu liczby wymaganych różnic zwykłych oraz sezonowych. Na tej podstawie przeprowadzimy odpowiednie różnicowanie w celu uzyskania szeregu stacjonarnego. Funkcja `ndiffs` zwraca, że szereg powinniśmy różnicować 1 raz, a funkcja `nsdiffs` stwierdza, że różnicowanie sezonowe powinno być przeprowadzone 1 razy.

```
ndiffs(euretail.learn) # 1
nsdiffs(euretail.learn) # 1

euretail.diff <- diff(euretail.learn, lag = 4)
euretail.diff <- diff(euretail.diff, lag = 1)

mean(euretail.diff)
```



Rysunek 17: Szereg po różnicowaniu i różnicowaniu sezonowym

Wykres 17 potwierdza, że szereg po wcześniej wspomnianych krotnościach różnicowań jest stacjonarny, również pozwala on na identyfikację potencjalnych modeli ARIMA. Rozważymy model autoregresyjny oraz model średniej ruchomej. Średnia różnicowanego szeregu wynosi 0.0239535, jest ona różna od 0, więc zbadamy modele z dryfem.

```
model.ar <- Arima(
  euretail.learn,
  order = c(4,2,0),
  seasonal = c(0,1,0),
  include.drift = TRUE
)
```

```
model.ma <- Arima(
  euretail.learn,
  order = c(0,2,9),
  seasonal = c(0,1,0),
  include.drift = TRUE
)
```

```
summary(model.ar)
```

```
## Series: euretail.learn
```

```
## ARIMA(4,2,0)(0,1,0)[4]
##
## Coefficients:
##          ar1      ar2      ar3      ar4
##      -0.4839  -0.2179  -0.0649  -0.4869
## s.e.   0.1311   0.1535   0.1504   0.1284
##
## sigma^2 = 0.2729:  log likelihood = -34.91
## AIC=79.82   AICc=81.49   BIC=88.51
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.00356198 0.4648136 0.3614478 -0.003625689 0.3764106 0.3227867
##              ACF1
## Training set -0.1478629
```

```
summary(model.ma)
```

```
## Series: euretail.learn
## ARIMA(0,2,9)(0,1,0)[4]
##
## Coefficients:
##          ma1      ma2      ma3      ma4      ma5      ma6      ma7      ma8
##      -0.9091  -0.0329  0.2071  -1.2720  0.8635  0.0895  -0.1202  0.3331
## s.e.   0.2124   0.2736  0.3215   0.3737  0.3374  0.2563   0.3682  0.3881
##          ma9
##      -0.1314
## s.e.   0.2198
##
## sigma^2 = 0.163:  log likelihood = -26.06
## AIC=72.13   AICc=79.22   BIC=89.5
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.05072199 0.3347288 0.2398156 -0.05254765 0.2503834 0.2141646
##              ACF1
## Training set -0.08625189
```

Dodatkowo zastosujemy funkcję `auto.arima`, przeprowadzając dokładne przeszukiwanie przestrzeni modeli z wykorzystaniem kryteriów informacyjnych AICc oraz BIC.

```
model.auto.aicc <- auto.arima(
  euretail.learn,
  stepwise = FALSE,
  approximation = FALSE,
  ic = "aicc"
```

```
)

model.auto.bic <- auto.arima(
  euretail.learn,
  stepwise = FALSE,
  approximation = FALSE,
  ic = "bic"
)

summary(model.auto.aicc)
```

```
## Series: euretail.learn
## ARIMA(1,0,0)(1,1,0)[4] with drift
##
## Coefficients:
##          ar1      sar1    drift
##          0.8515  -0.5440  0.2538
## s.e.    0.0784   0.1225  0.0625
##
## sigma^2 = 0.1799:  log likelihood = -24.14
## AIC=56.27   AICc=57.3   BIC=63.41
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.0229874 0.3919481 0.2850336 0.02506754 0.2992295 0.254546
##              ACF1
## Training set 0.1433959
```

```
summary(model.auto.bic)

## Series: euretail.learn
## ARIMA(1,0,0)(1,1,0)[4]
##
## Coefficients:
##          ar1      sar1
##          0.9659  -0.5528
## s.e.    0.0287   0.1201
##
## sigma^2 = 0.186:  log likelihood = -26.01
## AIC=58.02   AICc=58.62   BIC=63.37
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.08256266 0.4034748 0.2889494 0.08867202 0.3022119 0.2580429
##              ACF1
```

```
## Training set 0.09979758
```

W kolejnym kroku dopasujemy model trendu ze składową sezonową oraz nieliniowym trendem czasowym, wykorzystując regresję dla szeregów czasowych.

```
model.tslm <- tslm(euretail.learn ~ season + trend + I(trend^2))
summary(model.tslm)
```

```
##
## Call:
## tslm(formula = euretail.learn ~ season + trend + I(trend^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88518 -0.29972 -0.04017  0.19661  1.26550
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  87.8780432   0.2766735  317.624  < 2e-16 ***
## season2      0.2212937   0.2221878    0.996  0.324966
## season3      0.5403197   0.2224060    2.429  0.019479 *
## season4      0.4862449   0.2227674    2.183  0.034696 *
## trend        0.3691680   0.0231412   15.953  < 2e-16 ***
## I(trend^2)   -0.0017829   0.0004578   -3.895  0.000347 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5441 on 42 degrees of freedom
## Multiple R-squared:  0.9836, Adjusted R-squared:  0.9816
## F-statistic: 502.9 on 5 and 42 DF,  p-value: < 2.2e-16
```

Rozważymy również modele oparte na wygładzaniu wykładniczym, w tym proste wygładzanie, modele Holta, Holta-Wintersa oraz automatyczny model ETS.

```
model.ses <- ses(euretail.learn, initial = "optimal")

model.holt <- holt(euretail.learn)
model.holt.damped <- holt(euretail.learn, damped = TRUE)

model.hw.add <- hw(euretail.learn, seasonal = "additive")
model.hw.mult <- hw(euretail.learn, seasonal = "multiplicative")

model.ets <- ets(euretail.learn)
summary(model.ets)
```

```
## ETS(A,A,A)
##
```

```

## Call:
## ets(y = euretail.learn)
##
## Smoothing parameters:
##   alpha = 0.8487
##   beta  = 1e-04
##   gamma = 0.0012
##
## Initial states:
##   l = 89.3025
##   b = 0.2473
##   s = 0.1822 0.2615 -0.1366 -0.3071
##
## sigma: 0.3745
##
##      AIC      AICc      BIC
## 100.7838 105.5207 117.6246
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.01823076 0.341889 0.2372545 0.01847822 0.2499551 0.2118774
##              ACF1
## Training set 0.1752311

```

Na podstawie przeprowadzonych estymacji dokonamy oceny w oparciu o kryteria informacyjne oraz miary błędów dopasowania na zbiorze uczącym.

Spośród ręcznie identyfikowanych modeli ARIMA lepsze dopasowanie uzyskał model ARIMA(0,2,9)(0,1,0)₄, jednak ze względu na dużą liczbę parametrów i wysoką wartość kryterium BIC uznamy go za nadmiernie złożony. Najlepsze wyniki w klasie modeli ARIMA osiągnął model automatyczny ARIMA(1,0,0)(1,1,0)₄ z dryfem, który charakteryzuje się najniższymi wartościami kryteriów informacyjnych oraz prostą strukturą, co czyni go dobrym kandydatem do prognozowania.

Model TSLM z sezonowością oraz nieliniowym trendem bardzo dobrze opisuje dane historyczne, co potwierdza wysoka wartość skorygowanego współczynnika R^2 oraz istotność większości parametrów. Ze względu na deterministyczny charakter trendu model ten będzie traktowany jako punkt odniesienia w dalszej analizie.

Model ETS(A,A,A) cechuje się niskimi wartościami błędów prognoz na zbiorze uczącym, jednak wyższe wartości kryteriów informacyjnych wskazują na większą złożoność modelu.

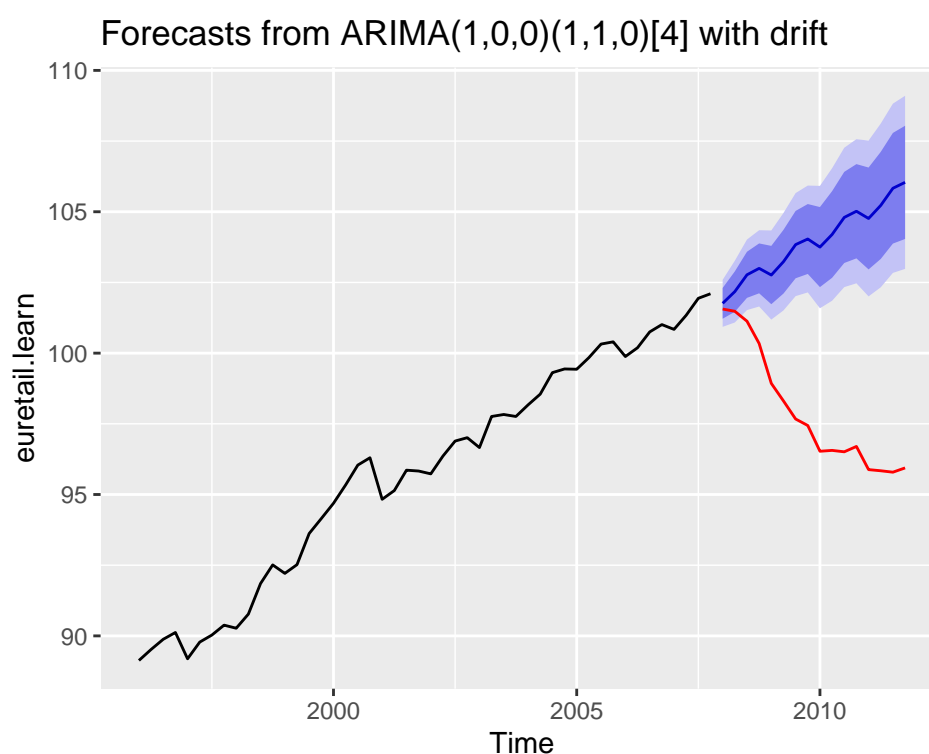
Do dalszej analizy i porównania prognoz wybierzemy modele

- ARIMA(1,0,0)(1,1,0)₄ z dryfem,
- model TSLM z trendem kwadratowym,
- model ETS(A,A,A).

2.4 Wyznaczenie prognoz dla zbioru testowego

W pierwszym kroku skonstruujemy prognozy na zbiór testowy z wykorzystaniem wybranego modelu ARIMA, który został wskazany w poprzednim etapie analizy jako jeden z najlepszych modeli pod względem kryteriów informacyjnych. Prognozy zostały wyznaczone na h okresów do przodu, gdzie h odpowiada liczbie obserwacji w zbiorze testowym. Na wykresie 18 przedstawiamy zarówno prognozy wraz z przedziałami predykcyjnymi, jak i rzeczywiste obserwacje testowe.

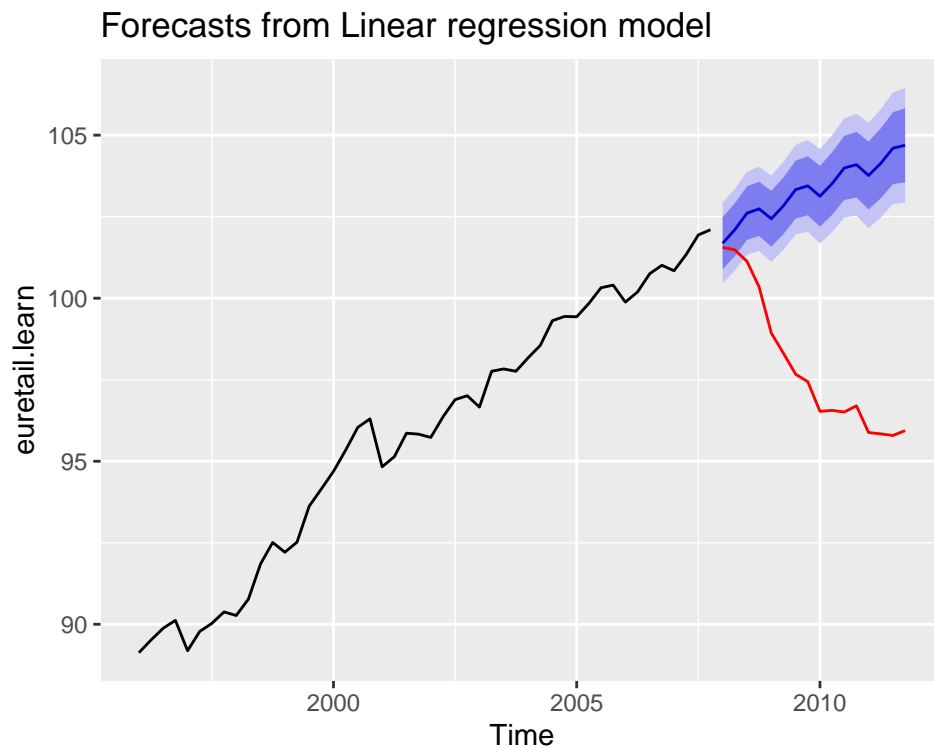
```
forecast.arima <- forecast(  
  model.auto.aicc,  
  h = h  
)
```



Rysunek 18: Prognoza ARIMA na zbiorze testowym

Następnie wykonamy prognozowanie na zbiór testowy przy użyciu modelu dekompozycyjnego TSLM, uwzględniającego sezonowość oraz nieliniowy trend kwadratowy. Model ten pozwala w sposób jawny opisać długookresową tendencję rozwojową szeregu oraz wahania sezonowe. Uzyskane prognozy prezentujemy na obrazku 19.

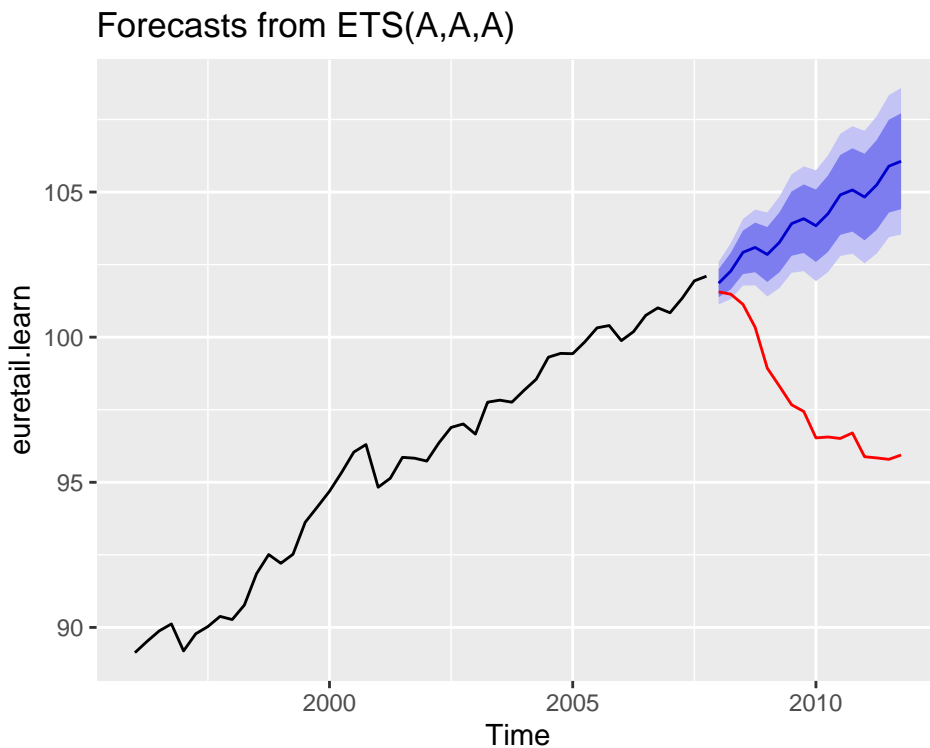
```
forecast.tslm <- forecast(  
  model.tslm,  
  h = h  
)
```



Rysunek 19: Prognoza TSLM na zbiorze testowym

Kolejnym rozważanym podejściem będzie automatycznie dobrany model ETS, oparty na algorytmach wygładzania wykładniczego. Metoda ta pozwala na elastyczne modelowanie poziomu, trendu oraz sezonowości szeregu czasowego. Prognozy wygenerowane przez model ETS obrazujemy na rysunku 20.

```
forecast.ets <- forecast(
  model.ets,
  h = h
)
```



Rysunek 20: Prognoza ETS na zbiorze testowym

W celu obiektywnego porównania rozważanych modeli wykorzystamy miary błędów prognoz wyznaczone na zbiorze testowym, takie jak RMSE, MAE oraz MAPE. Porównanie to pozwala wskazać model charakteryzujący się najlepszą jakością predykcji poza próbą uczącą, a tym samym rekomendowany do dalszych analiz i zastosowań praktycznych.

```
cat("ARIMA\n")
```

```
## ARIMA
```

```
accuracy(forecast.arima, euretail.test)
```

```
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set  0.0229874 0.3919481 0.2850336 0.02506754 0.2992295 0.254546
## Test set     -6.0368630 6.8417163 6.0368630 -6.23554999 6.2355500 5.391150
##              ACF1 Theil's U
## Training set  0.1433959      NA
## Test set      0.8220115 12.35776
```

```
cat("\ntsml\n")
```

```
##
```

```
## tslm
```

```
accuracy(forecast.tslm, euretail.test)
```

```
##                ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -2.960657e-16 0.5089251 0.3794808 -0.002951526 0.4036361 0.3388909
## Test set     -5.405615e+00 6.1078827 5.4056145 -5.582924909 5.5829249 4.8274211
##                ACF1 Theil's U
## Training set 0.7518894      NA
## Test set     0.8222905 11.02894
```

```
cat("\nets\n")
```

```
##
## ets
```

```
accuracy(forecast.ets, euretail.test)
```

```
##                ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.01823076 0.341889 0.2372545 0.01847822 0.2499551 0.2118774
## Test set     -6.10925411 6.895260 6.1092541 -6.30904041 6.3090404 5.4557983
##                ACF1 Theil's U
## Training set 0.1752311      NA
## Test set     0.8215713 12.45259
```

Niska jakość prognoz w zbiorze testowym wynika głównie z istotnej zmiany zachowania szeregu w tym okresie. W danych testowych obserwujemy wyraźny spadek poziomu szeregu, który nie był obecny w zbiorze uczącym. W konsekwencji modele, które poprawnie ekstrapolują dotychczasowy trend i sezonowość, generują prognozy wizualnie spójne z historią, jednak obarczone dużymi błędami punktowymi.

2.5 Wnioski

Na podstawie przeprowadzonej analizy porównawczej rozważanych podejść do modelowania szeregu czasowego można stwierdzić, że modele ARIMA oraz ETS charakteryzowały się bardzo zbliżoną jakością dopasowania do zbioru uczącego. W obu przypadkach uzyskaliśmy relatywnie niskie wartości miar błędu, co świadczy o dobrej zdolności tych modeli do opisu dynamiki szeregu, obejmującej zarówno trend, jak i sezonowość. Model dekompozycyjny typu `tslm`, wypadł najslabiej pod względem miar błędu, co sugeruje, że dobrze opisuje długookresową strukturę trendu, lecz gorzej radzi sobie z krótkookresowymi wahaniami szeregu.

W trakcie analizy wykorzystaliśmy zarówno klasyczne metody identyfikacji modeli (ACF, PACF, różnicowanie, eliminację sezonowości), jak i podejścia automatyczne (`auto.arima()`, `ets()`). Dodatkowo sprawdziliśmy istotność współczynników, własności reszt oraz poprawność założeń modelowych, co pozwoliło na świadomy wybór modeli rekomendowanych do dalszej analizy i prognozowania.

Należy jednak podkreślić, że ocena jakości prognoz na zbiorze testowym okazała się proble-

matyczna dla wszystkich rozważanych modeli. W okresie testowym obserwujemy wyraźną zmianę zachowania szeregu czasowego, polegającą na gwałtownym spadku poziomu, który nie był widoczny w zbiorze uczącym. W konsekwencji nawet modele dobrze dopasowane do danych historycznych generowały prognozy obarczone dużymi błędami punktowymi, co znalazło odzwierciedlenie w wysokich wartościach RMSE, MAE oraz współczynnika Theila.

W związku z powyższym niską jakość prognoz w zbiorze testowym należy interpretować nie jako błąd konstrukcji modeli, lecz jako efekt zmiany struktury procesu generującego dane. Ostatecznie, biorąc pod uwagę zarówno dopasowanie do danych uczących, jak i własności modeli, za najbardziej adekwatne podejścia dla analizowanego szeregu uznajemy modele ARIMA oraz ETS, przy czym ich skuteczność prognostyczna jest istotnie ograniczona przez niestabilność szeregu w końcowym okresie obserwacji.