Analiza szeregów czasowych
Raport 4

Jan Solarz 243889
Karol Pustelnik

22 czerwca 2021

Spis treści

Wykorzystując dane dochodu narodowego USA (dostępne w pakiecie astsa) *gnp* możemy do nich dopasować odpowiedni model ARIMA za pomocą przeprowadzone pełnej diagnostyki i analizy.

1 Dopasowanie modelu ARIMA do szeregu gnp

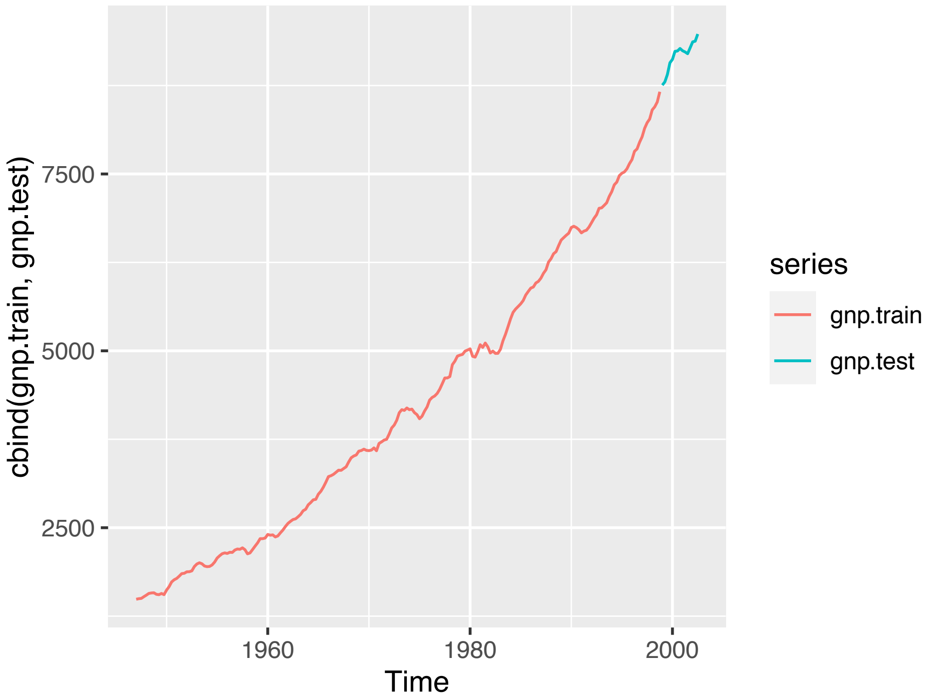
1.1 Podział danych

Zaczynamy od podziału danych na część uczącą i testową. Dane występują z interwałem kwartalnym, część testowa ma horyzont prognozy równy 15 i odnosi się do lat 1999-2002. Prognoza dotyczy tego okresu czasu.

```r
library(astsa)
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
##
## Attaching package: 'forecast'
## The following object is masked from 'package:astsa':
##
##     gas

#install.packages("itsmr")
library(itsmr) # uwaga itsmr::forecast przesłania f-cję forecast z pakietu forecast!!!

##
## Attaching package: 'itsmr'
## The following object is masked from 'package:forecast':
##
##     forecast

library(lmtest)

## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

library(ggfortify)

## Loading required package: ggplot2
## Registered S3 methods overwritten by 'ggfortify':
##   method                 from
##   autoplot.Arima         forecast
##   autoplot.acf           forecast
##   autoplot.ar            forecast
##   autoplot.bats          forecast
##   autoplot.decomposed.ts forecast
##   autoplot.ets           forecast
##   autoplot.forecast      forecast
##   autoplot.stl           forecast
##   autoplot.ts            forecast
##   fitted.ar              forecast
##   fortify.ts             forecast
##   residuals.ar           forecast

library(stats)
library(ggplot2)
library(DescTools)
```

```
##
## Attaching package: 'DescTools'
## The following object is masked from 'package:forecast':
##
##     BoxCox
```

```
### Podział danych na część uczącą i testową
gnp.train <- window(gnp, end=c(1998,4))
gnp.test <- window(gnp, start=c(1999,1))
autoplot(cbind(gnp.train, gnp.test))
```

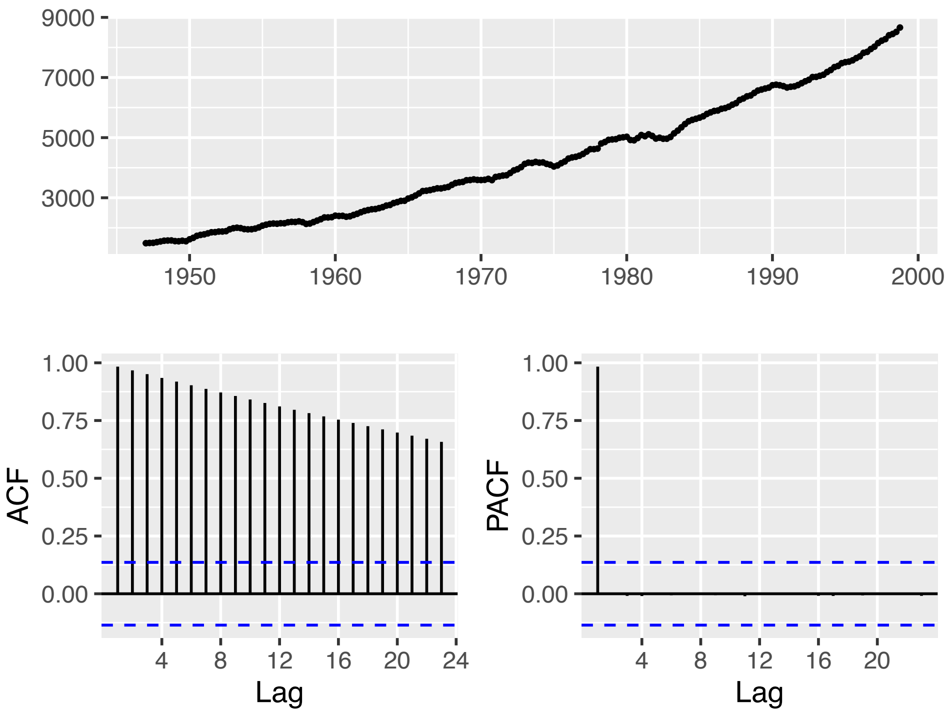



```
h <- length(gnp.test) # horyzont prognozy
h

## [1] 15
```

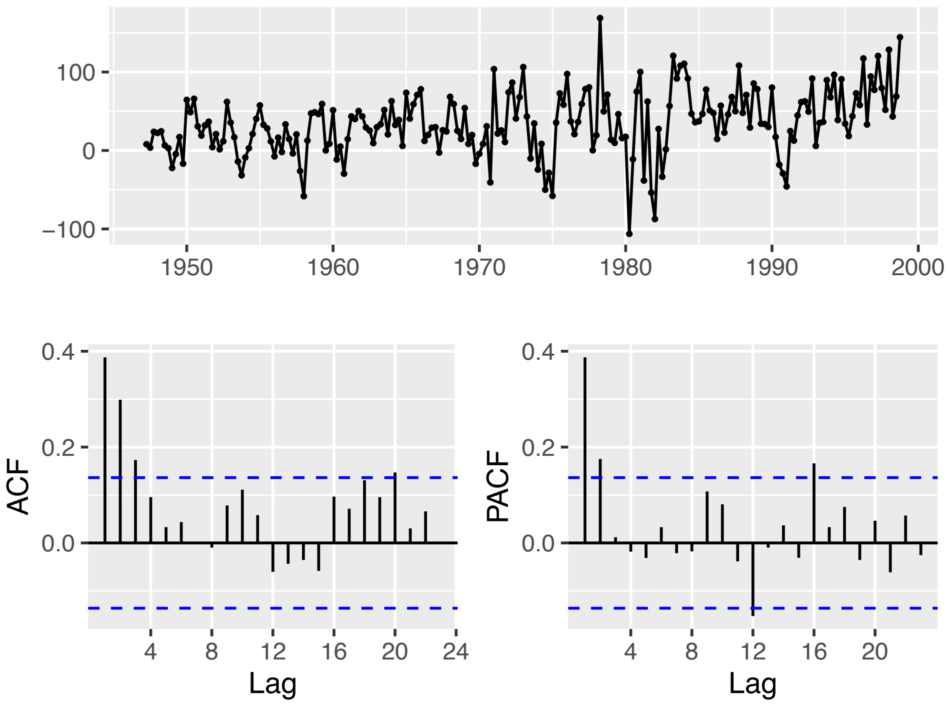
1.2 Identyfikacja transformacji i modeli stacjonarnych

Dokonujemy różnicowania i transformacji Box-Coxa w celu otrzymania postaci stacjonranej. Następnie identyfikujemy rzędy modeli stacjonarnych $MA(q)$, $AR(p)$ i modelu mieszanego $ARMA(p, q)$.

```
ggtsdisplay(gnp.train)
```



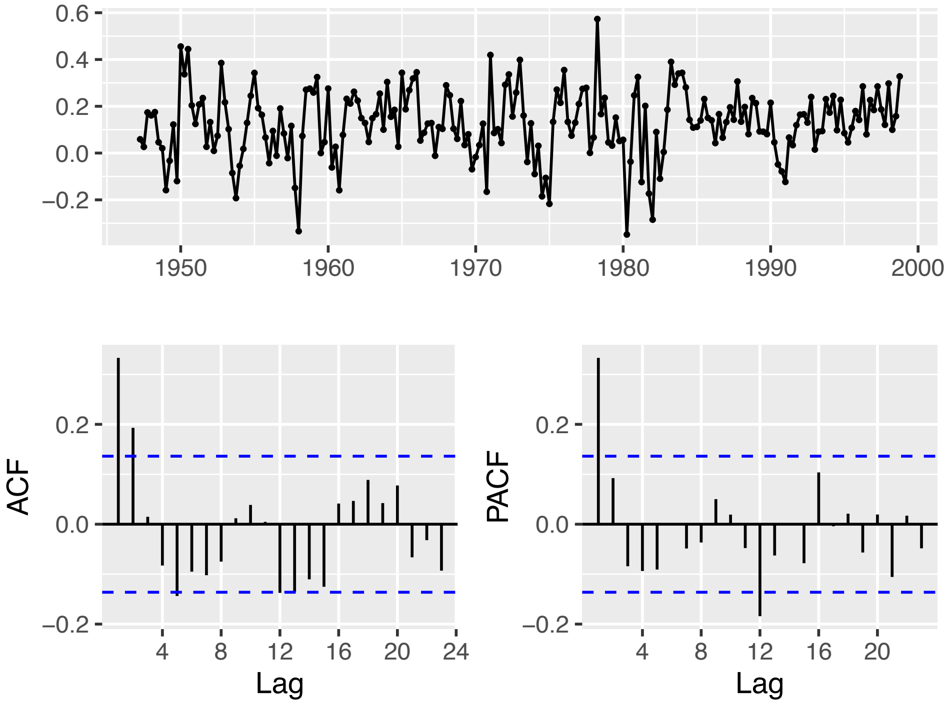

```
cat("Wymagana ilość różnicowań:")

## Wymagana ilość różnicowań:

ndiffs(gnp.train)

## [1] 2

ggtsdisplay(diff(gnp.train))
```




```
lambda <-  BoxCox.lambda(gnp.train)
gnp.train.BC <- BoxCox(gnp.train, lambda=lambda)
cat("Wymagana ilość różnicowań po transfomrmacji Boxa-Coxa:")

## Wymagana ilość różnicowań po transfomrmacji Boxa-Coxa:

ndiffs(gnp.train.BC) # jaka krotność różnicowania jest potrzebna?

## [1] 1

### Wstępna identyfikacja modeli MA(q) i/lub AR(p)
gnp.train.BC.diff <- diff(gnp.train.BC)
Box.test(gnp.train.BC.diff)

##
##  Box-Pierce test
##
## data:  gnp.train.BC.diff
## X-squared = 22.966, df = 1, p-value = 1.649e-06
```

Po zastosowaniu na szeregu notowań transformacji Box-Coxa i jednego różnicowania otrzymujemy szereg stacjonarny.

```
ggtsdisplay(gnp.train.BC.diff)
```



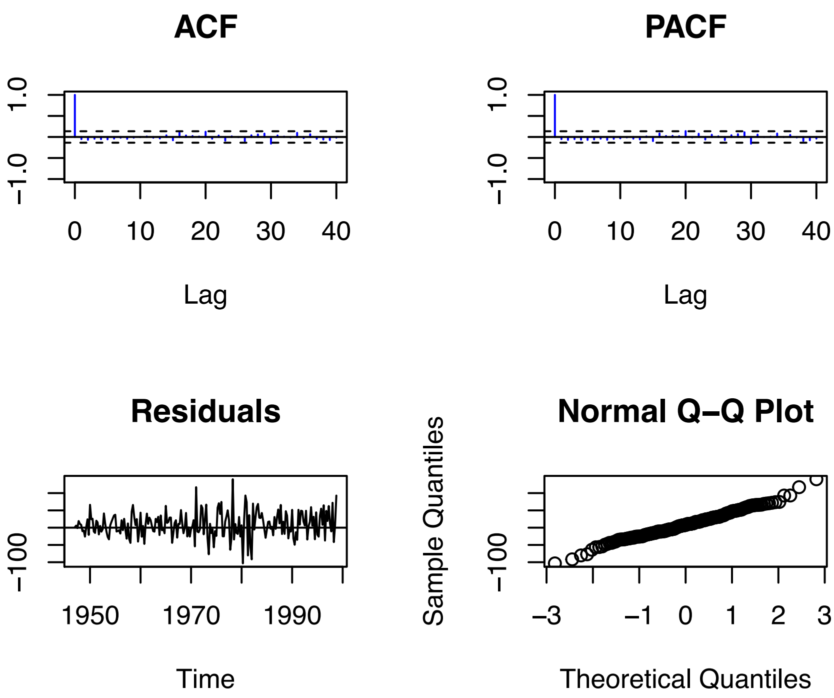

```
ma13<-Arima(gnp.train, order = c(0,1,13))
ar12<-Arima(gnp.train, order = c(12,1,0))
auto.arima<-auto.arima(gnp.train)
auto.arima

## Series: gnp.train
## ARIMA(0,2,3)
##
## Coefficients:
##           ma1      ma2      ma3
##       -0.6915  -0.0624  -0.1857
## s.e.   0.0706   0.0928   0.0646
##
## sigma^2 estimated as 1378:  log likelihood=-1036.3
## AIC=2080.61   AICc=2080.81   BIC=2093.92
```

- Dla modelu $MA(q)$ funkcja autokorelacji ACF dla opóźnienia 13 posiada wartości opóźnień mniejsze od poziomu 13 mieszczące się w przedziale ufności. Stąd wynika że rząd q średniej ruchomej wynosi 13.
- Dla modelu $AR(p)$ funkcja cząstkowej korelacji PACF dla opóźnienia 12 posiada wartości opóźnień mniejsze od poziomu 12 mieszczące się w przedziale ufności. Stąd wynika że rząd p średniej ruchomej wynosi 12.
- Dla modelu $ARIMA(p, q)$ metoda krokowa wskazuje za najlepsze dopasowanie modelu $ARIMA(0, 2, 3)$ o czym świadczą najniższe kryteria informacyjne AIC, AICC, BIC.

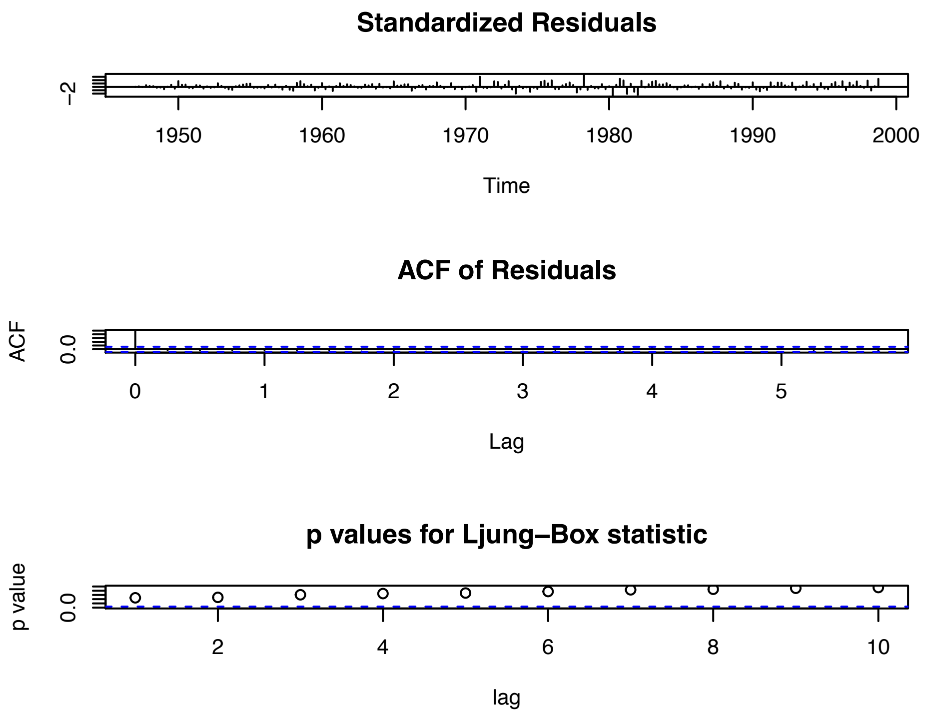
1.3 Zbadanie poprawności reszt -residua

1.3.1 Model średniej ruchomej - MA

```
test(residuals(ma13))

## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     12.38    0.9022
## McLeod-Li Q                Q ~ chisq(20)     58.53         0 *
## Turning points T  (T-137.3)/6.1 ~ N(0,1)       142    0.4408
## Diff signs S       (S-103.5)/4.2 ~ N(0,1)       105    0.7193
## Rank P           (P-10764)/501.8 ~ N(0,1)     11501    0.1419
```

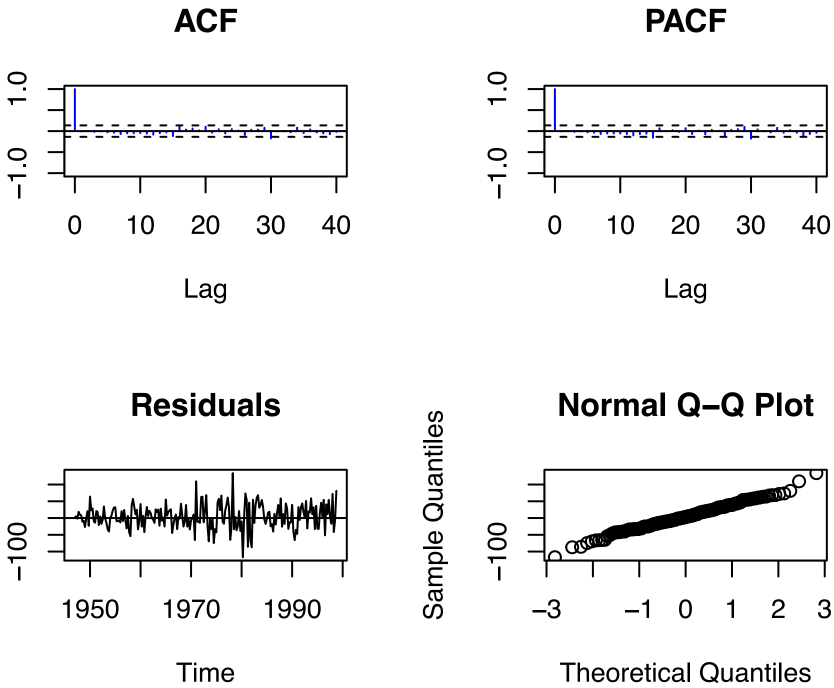



```
tsdiag(ma13)
```

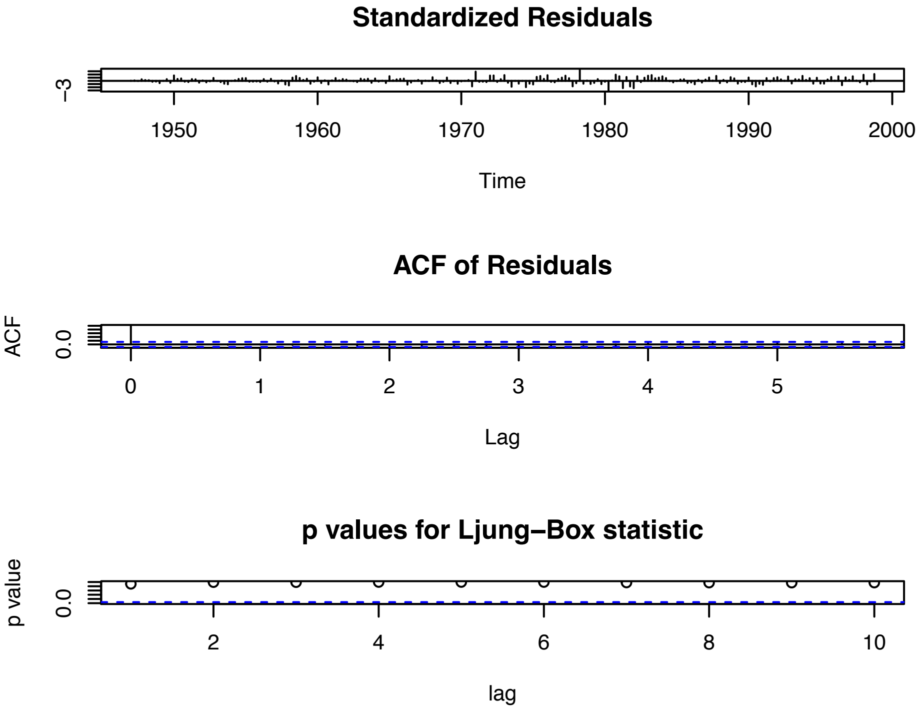


1.3.2 Model autoregresyjny - AR

```
test(residuals(ar12))

## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     16.17    0.7059
## McLeod-Li Q                Q ~ chisq(20)     73.26         0 *
## Turning points T  (T-137.3)/6.1 ~ N(0,1)       146    0.1523
## Diff signs S      (S-103.5)/4.2 ~ N(0,1)       108    0.2809
## Rank P          (P-10764)/501.8 ~ N(0,1)     11157    0.4335
```



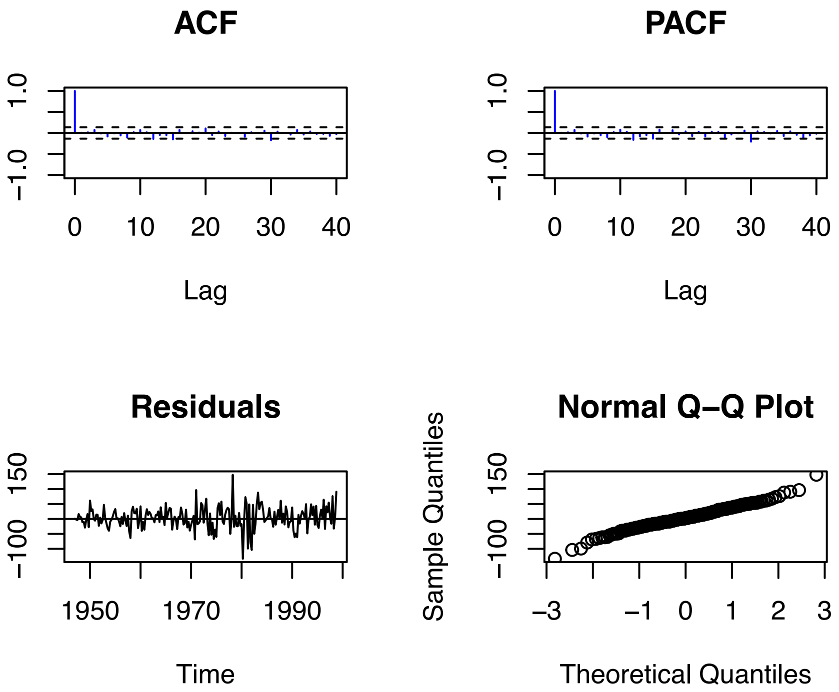

```r
tsdiag(ar12)
```



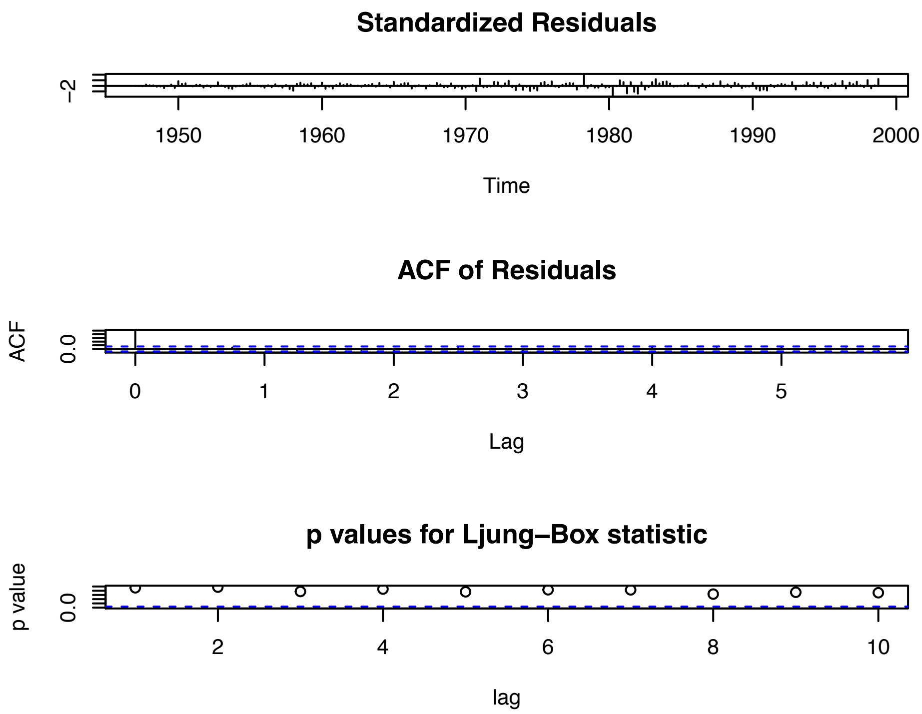
1.3.3 Model ARIMA

```r
test(residuals(auto.arima))
```

```
## Null hypothesis: Residuals are iid noise.
## Test                         Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     22.29    0.3252
## McLeod-Li Q                Q ~ chisq(20)     67.93         0 *
## Turning points T   (T-137.3)/6.1 ~ N(0,1)       141    0.5448
## Diff signs S       (S-103.5)/4.2 ~ N(0,1)       107    0.4017
## Rank P           (P-10764)/501.8 ~ N(0,1)     11254    0.3288
```




```r
tsdiag(auto.arima)
```



Na podstawie zastosowanych testów statystycznych i metod graficznych nie mamy podstaw do odrzecenia hipotez zerowych w każdym z modeli- w każdym teście w każdej z metod p-value jest większe od 0.05. To oznacza że modele są dobrze dopasowane.

1.4 Porównanie modeli

```
cat("Kryteria informacyjne MA: ", ma13$aic,ma13$aicc, ma13$bic)

## Kryteria informacyjne MA:  2117.575 2119.762 2164.233

cat("Kryteria informacyjne AR: ", ar12$aic,ar12$aicc, ar12$bic)

## Kryteria informacyjne AR:  2106.949 2108.835 2150.274

cat("Kryteria informacyjne ARIMA: ", auto.arima$aic,auto.arima$aicc, auto.arima$bic)

## Kryteria informacyjne ARIMA:  2080.609 2080.808 2093.92
```

Kryteria wskazują jednoznacznie za najlepszą wydajność modelu ARIMA.

1.5 Istotność współczynników

1.5.1 Model MA

```
coeftest(ma13)

##
```

```
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1  0.401110   0.069302  5.7878 7.129e-09 ***
## ma2  0.441803   0.077590  5.6941 1.241e-08 ***
## ma3  0.331102   0.081318  4.0717 4.667e-05 ***
## ma4  0.304571   0.083815  3.6338 0.0002792 ***
## ma5  0.241194   0.088992  2.7103 0.0067229 **
## ma6  0.264264   0.092647  2.8524 0.0043395 **
## ma7  0.136921   0.091421  1.4977 0.1342108
## ma8  0.115637   0.088282  1.3099 0.1902456
## ma9  0.187412   0.091269  2.0534 0.0400332 *
## ma10 0.254345   0.077400  3.2861 0.0010158 **
## ma11 0.212519   0.070894  2.9977 0.0027201 **
## ma12 0.065703   0.071839  0.9146 0.3604072
## ma13 0.129422   0.088727  1.4586 0.1446623
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ma.fixed <- numeric(13)
ma.fixed[c(1:6,9:11)] <- NA
ma.zeros <- Arima(gnp.train, order=c(0,1,13), fixed=ma.fixed)
cat("Kryteria informacyjne MA po redukcji: ", ma.zeros$aic,ma.zeros$aicc, ma.zeros$bic)
```

```
## Kryteria informacyjne MA po redukcji:  2114.988 2116.111 2148.315
```

1.5.2 Model AR

```
coeftest(ar12)
```

```
##
## z test of coefficients:
##
##        Estimate Std. Error z value  Pr(>|z|)
## ar1   0.3615396  0.0697038  5.1868 2.139e-07 ***
## ar2   0.2275594  0.0740991  3.0710  0.002133 **
## ar3   0.0557962  0.0753708  0.7403  0.459124
## ar4   0.0193596  0.0754545  0.2566  0.797508
## ar5  -0.0192326  0.0753584 -0.2552  0.798557
## ar6   0.0773832  0.0755866  1.0238  0.305944
## ar7  -0.0045655  0.0753481 -0.0606  0.951684
## ar8  -0.0351148  0.0752805 -0.4665  0.640892
## ar9   0.1335062  0.0751731  1.7760  0.075736 .
## ar10  0.1390855  0.0763465  1.8218  0.068490 .
## ar11  0.0567733  0.0761511  0.7455  0.455948
## ar12 -0.1060658  0.0713578 -1.4864  0.137175
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ar.fixed <- numeric(12)
ar.fixed[c(1:2)] <- NA
ar.zeros <- Arima(gnp.train, order=c(12,1,0), fixed=ar.fixed)
cat("Kryteria informacyjne AR po redukcji: ", ar.zeros$aic,ar.zeros$aicc, ar.zeros$bic)

## Kryteria informacyjne AR po redukcji:  2107.804 2107.922 2117.802
```

1.5.3 Model ARIMA

```
coeftest(auto.arima)

##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1 -0.691464   0.070573 -9.7978 < 2.2e-16 ***
## ma2 -0.062403   0.092779 -0.6726  0.501203
## ma3 -0.185736   0.064647 -2.8731  0.004065 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

arima.fixed <- numeric(3)
arima.fixed[c(1,3)] <- NA
arima.zeros <- Arima(gnp.train, order=c(0,2,3), fixed=arima.fixed)
cat("Kryteria informacyjne ARIMA po redukcji: ", arima.zeros$aic,arima.zeros$aicc, arima

## Kryteria informacyjne ARIMA po redukcji:  2079.042 2079.161 2089.026
```

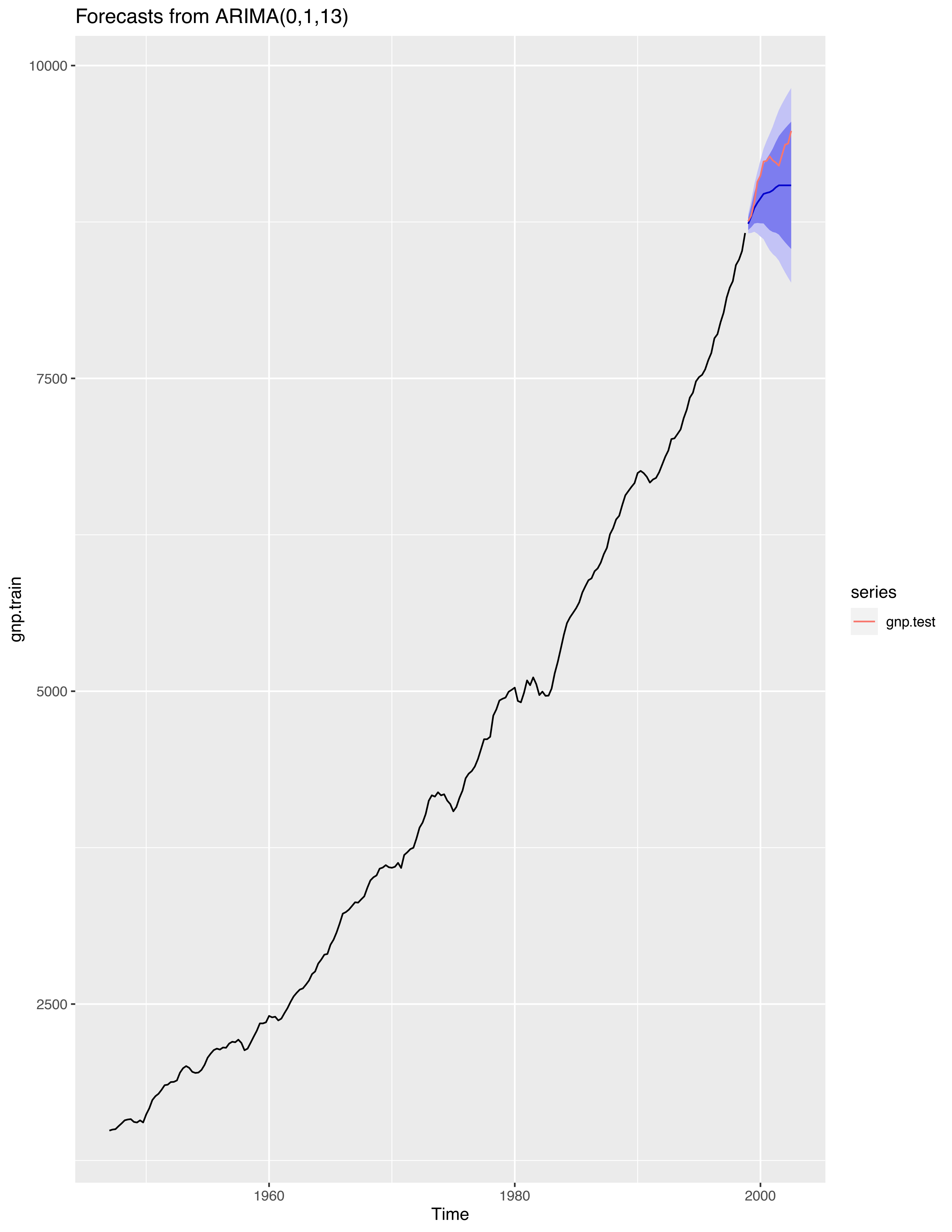
- Usuwamy odpowiednie współczynniki które mają odpowiednio wartości p-value mniejsze od poziomu 0.05.
- Widzimy nieznacznie polepszenie się modeli po redukcji współczynników w kryteraich informacyjnych.

1.6 Wnioski

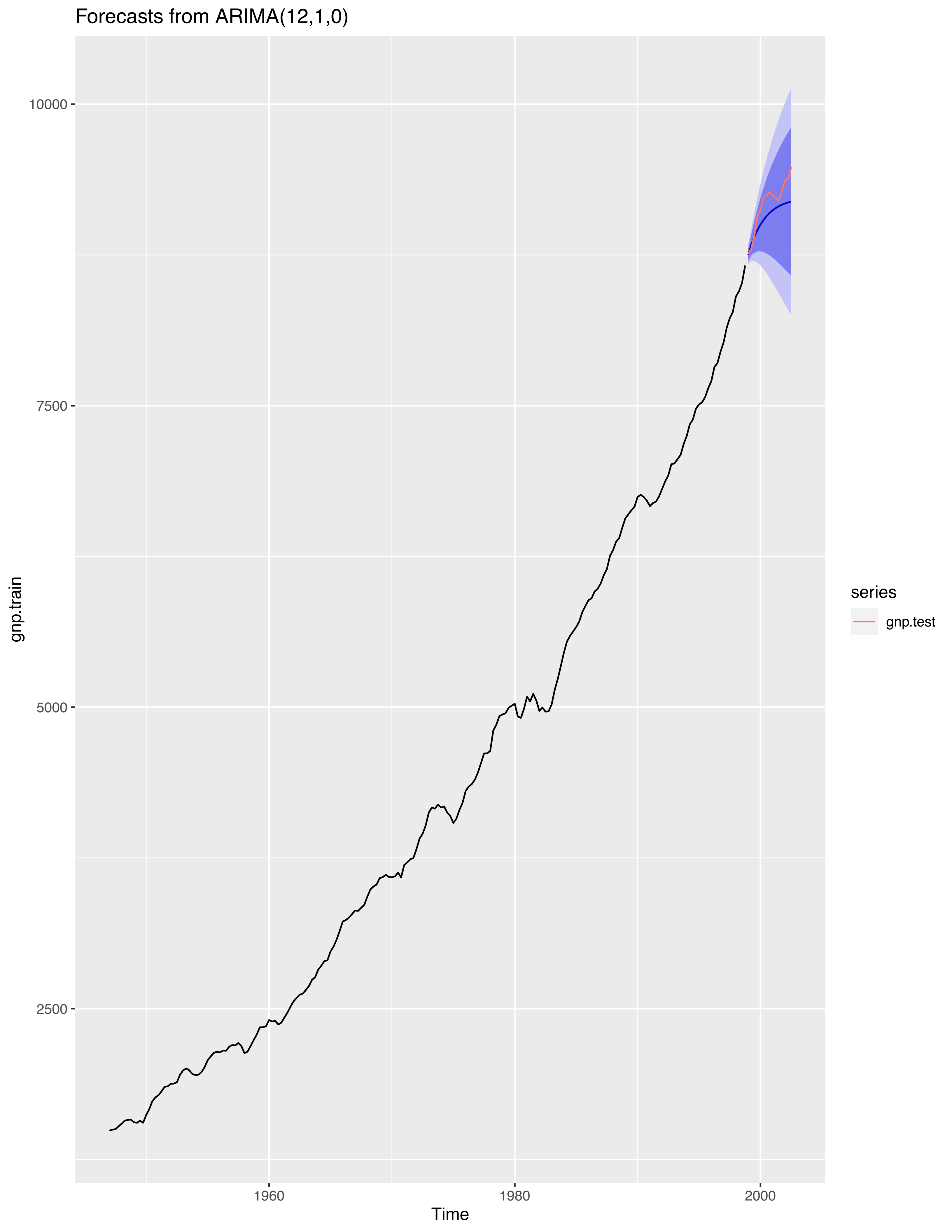
Kryteria informacyjne jednoznacznie wskazują za najlepsze dopasowanie modelu ARIMA. Wartości p-value w testach w analizie reszt jedynie dla modelu ARIMA są zawsze większe od poziomu 0.3 co również zapewnia nas w przekonaniu że jest to najlepszy model. Model $AR(12)$ jest lepiej dopasowany od $MA(12)$- wskazują na to kryteria informacyjne.

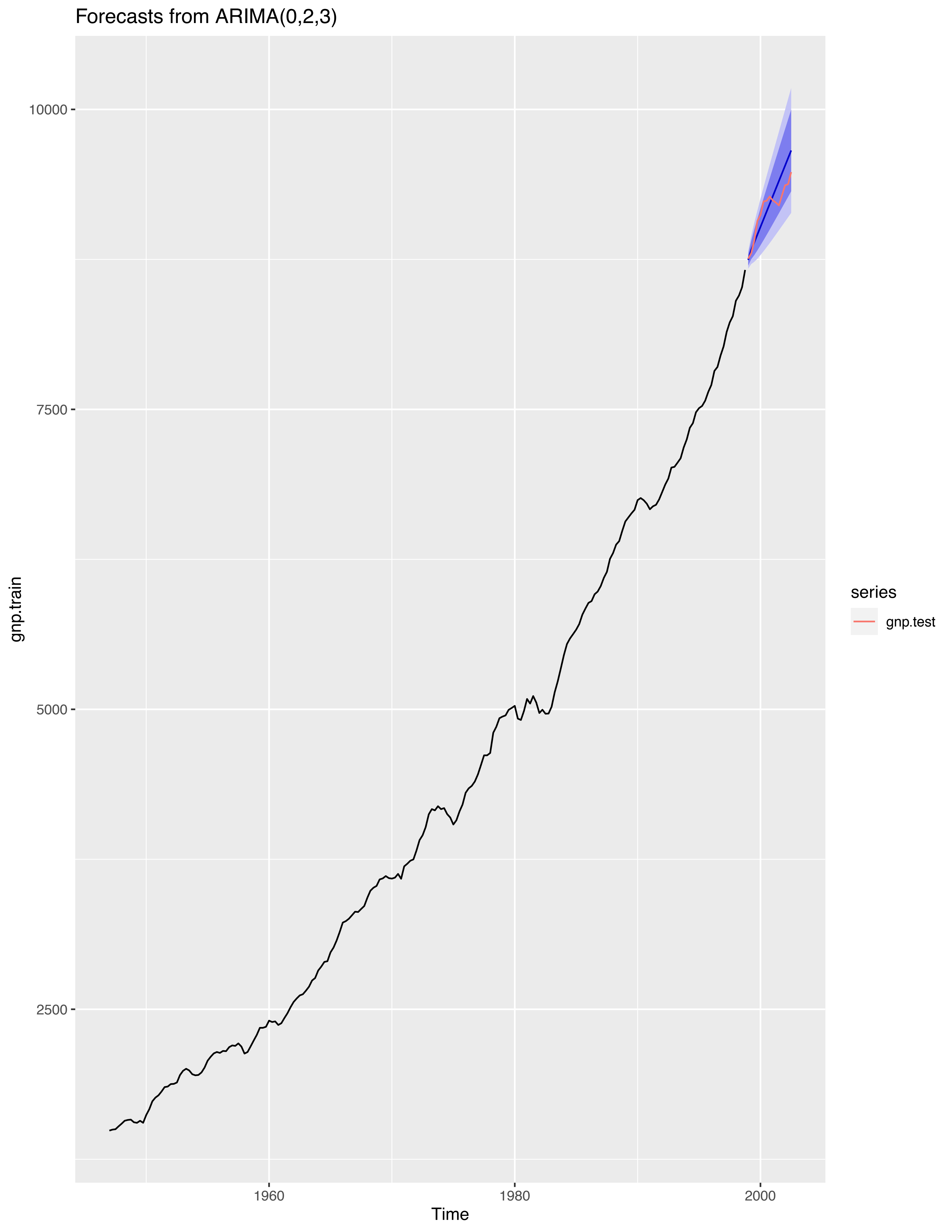
1.7 Konstrukcja prognoz

```r
library(fma)

##
## Attaching package: 'fma'
## The following objects are masked from 'package:itsmr':
##
##     airpass, strikes
## The following objects are masked from 'package:astsa':
##
##     chicken, sales

library(forecast)
h <- length(gnp.test)
ma.forecast <- forecast::forecast(ma.zeros, h = h)
#autoplot(ma.forecast) + autolayer(gnp.test)
ar.forecast <- forecast::forecast(ar.zeros, h = h)
#autoplot(ar.forecast) + autolayer(gnp.test)


arima.forecast <- forecast::forecast(arima.zeros, h = h)
#autoplot(arima.forecast) +autolayer(gnp.test)
par(mfrow=c(2,2))
#autoplot(cbind(gnp.train,gnp.test))
autoplot(ma.forecast)+autolayer(gnp.test)

## Warning: `filter_()` was deprecated in dplyr 0.7.0.
## Please use `filter()` instead.
## See vignette('programming') for more help
## Error: Invalid input: date_trans works with objects of class Date only

autoplot(ar.forecast)+autolayer(gnp.test)

## Error: Invalid input: date_trans works with objects of class Date only

autoplot(arima.forecast)+autolayer(gnp.test)

## Error: Invalid input: date_trans works with objects of class Date only
```



Forecasts from ARIMA(0,1,13)
10000
7500
5000
2500
gnp.train
1960
1980
2000
Time
series
gnp.test


Forecasts from ARIMA(12,1,0)
10000
7500
5000
2500
gnp.train
1960
1980
2000
Time
series
gnp.test



Forecasts from ARIMA(0,2,3)
10000
7500
5000
2500
gnp.train
1960
1980
2000
Time
series
gnp.test

Widzimy, że najbardziej sensowną i zbliżoną do rzeczywistej prognozą jest ta z modelu ARIMA. Przedział ufności tutaj jest najwęższy.

1.8 Ocena dokładności i porównanie

```
accuracy(ma.forecast,gnp.test)

##                     ME      RMSE       MAE       MPE     MAPE     MASE
## Training set  11.23145  37.93244  29.72248 0.2751444 0.789804 0.196597
## Test set     209.97916 239.77863 209.97916 2.2634975 2.263498 1.388891
##                     ACF1 Theil's U
## Training set -0.06964959        NA
## Test set      0.65105206  3.147886

accuracy(ar.forecast,gnp.test)

##                      ME      RMSE       MAE       MPE      MAPE      MASE
## Training set   8.669031  38.61327  29.43109 0.2128926 0.7868895 0.1946697
## Test set     118.892940 145.02473 123.18656 1.2791110 1.3278921 0.8148078
##                    ACF1 Theil's U
## Training set -0.0811651        NA
## Test set      0.5813512  1.905563

accuracy(arima.forecast,gnp.test)

##                      ME      RMSE       MAE         MPE      MAPE      MASE
## Training set   3.885223  36.71698  27.59238  0.08738656 0.7352485 0.1825076
## Test set     -39.027410 130.06256 110.77085 -0.41319888 1.1959731 0.7326851
```

```
##                      ACF1 Theil's U
## Training set 0.03419921        NA
## Test set     0.84575759  1.711471
```

Miary dokładności błędów jednoznacznie wskazują na model automatyczny typu ARIMA. Najmniej optymalnym modelem jest model MA.

2 Porównanie dokładności prognoz dla wybranych danych

2.1 Podział danych

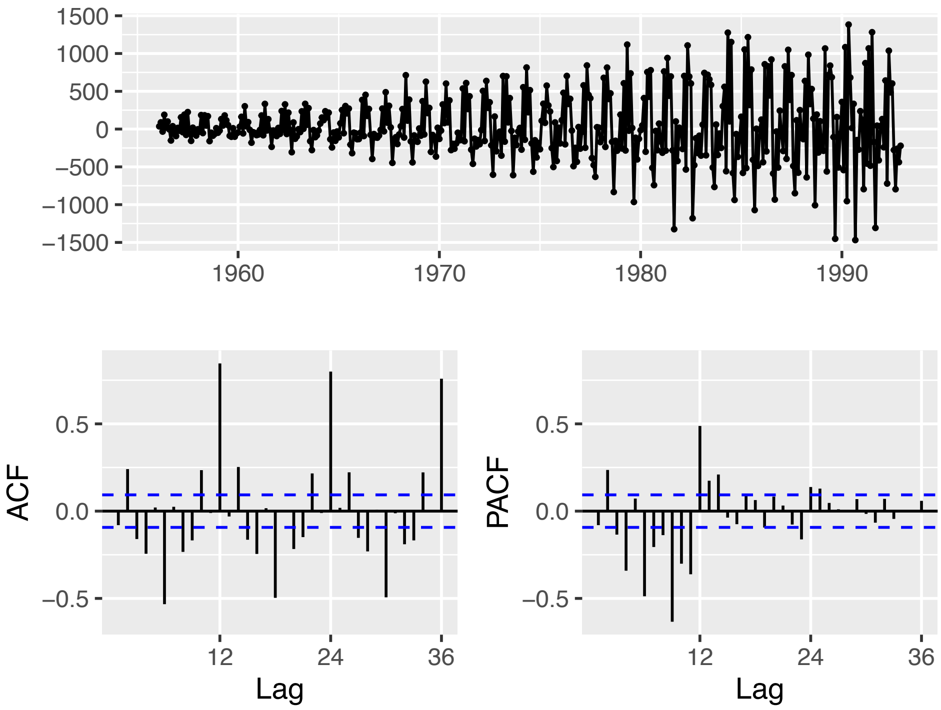
Wybieramy jako zbiór danych miesięczną produkcję energii elektrycznej w Australii w latach 1965-1995. Jako zbiór testowy bierzemy lata 1993-1995.

```
### Podział danych na część uczącą i testową
elec.train <- window(elec, end=c(1992,12))
elec.test <- window(elec, start=c(1993,1))
autoplot(cbind(elec.train, elec.test))
```


```
h <- length(elec.test) # horyzont prognozy
h

## [1] 32
```

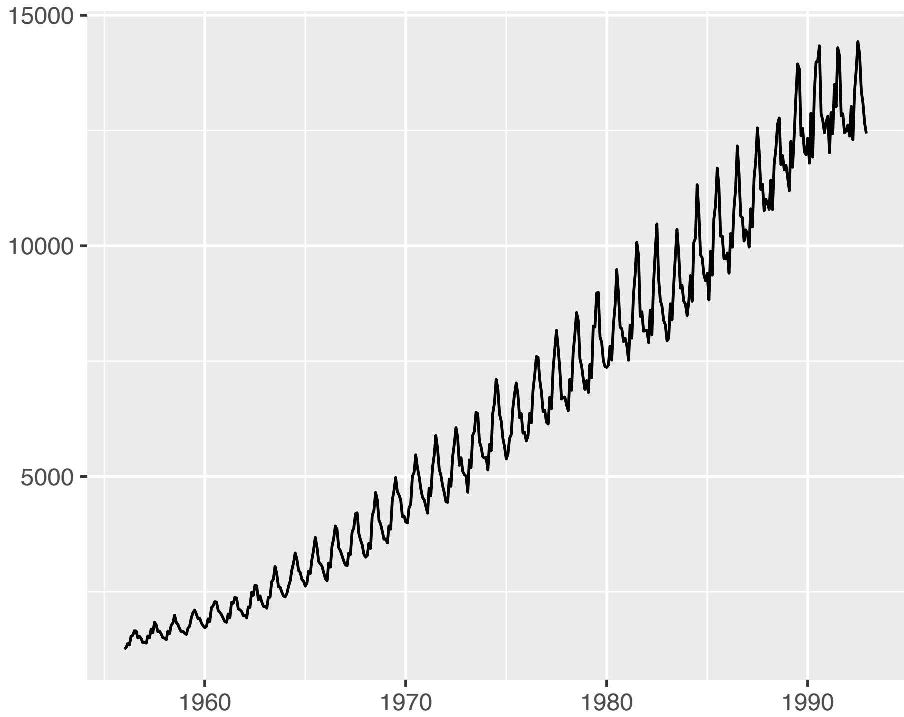
2.2 Dopasowanie odpowiednich modeli

```
 lambda <-  BoxCox.lambda(elec.train)
elec.train.box <- BoxCox(elec.train, lambda = lambda)
ndiffs(elec.train)

## [1] 1

elec.train.diff.box<-diff(elec.train,1)
Box.test(elec.train.diff.box)

##
##  Box-Pierce test
##
## data:  elec.train.diff.box
## X-squared = 2.86, df = 1, p-value = 0.09081

ggtsdisplay(elec.train.diff.box)
```

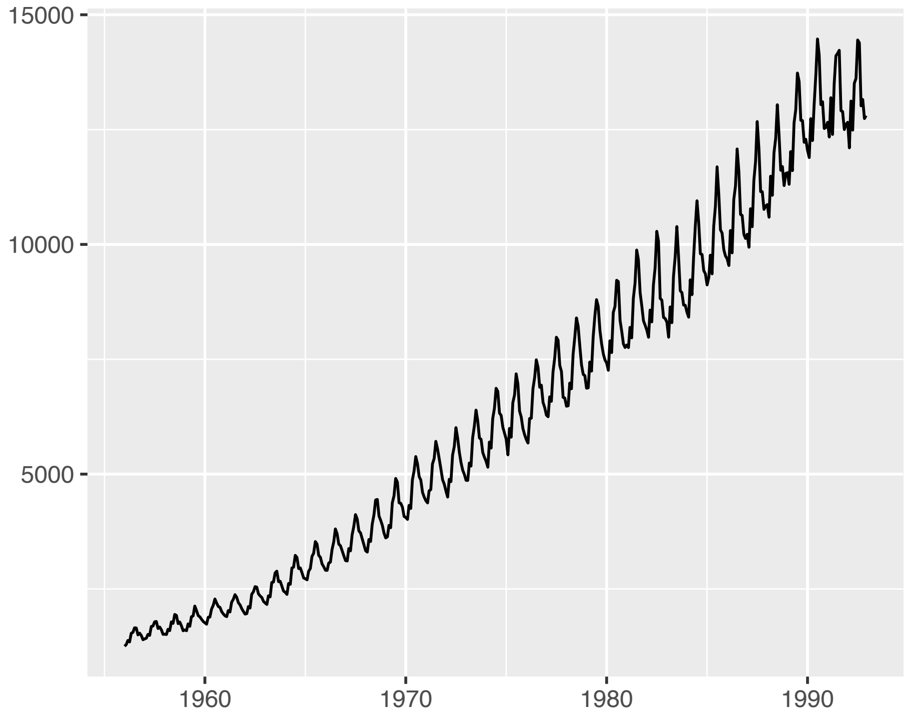


Po zastosowaniu tranformacji Boxa-Coxa i różnicowaniu p-value jest większe od 0.05 więc nie mamy podstaw do odrzucenia hipotezy zerowej. Raczej mamy do czynienia z białym szumem.

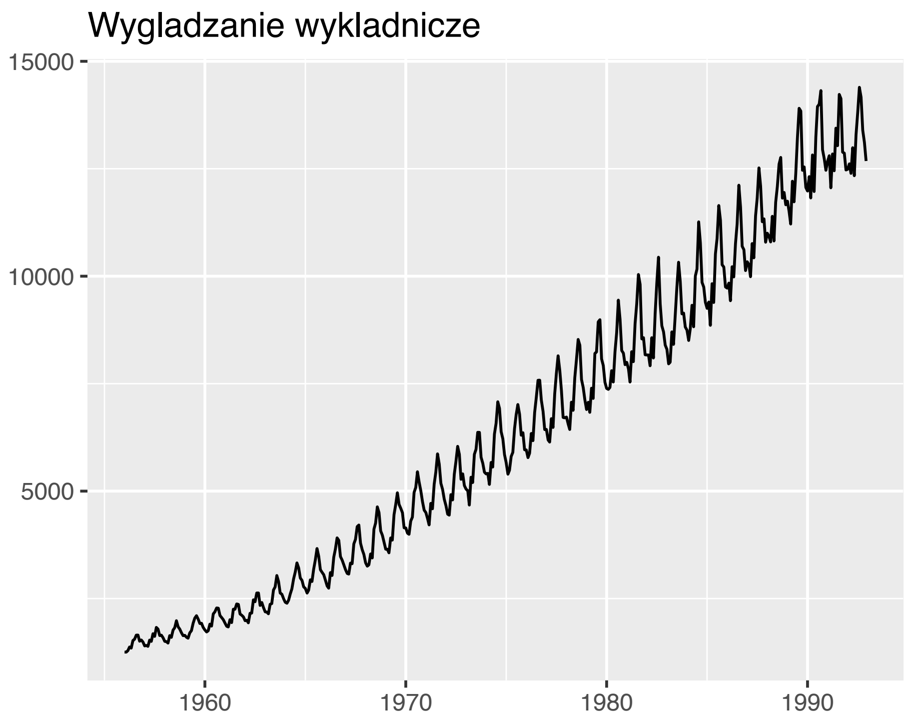
```r
autoplot(elec.train)
```




```r
model_arima<-auto.arima(elec.train)
model_arima

## Series: elec.train
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.6695  -0.5914
## s.e.   0.0383   0.0346
##
## sigma^2 estimated as 24997:  log likelihood=-2795.71
## AIC=5597.42   AICc=5597.47   BIC=5609.62

autoplot(model_arima$fitted)
```



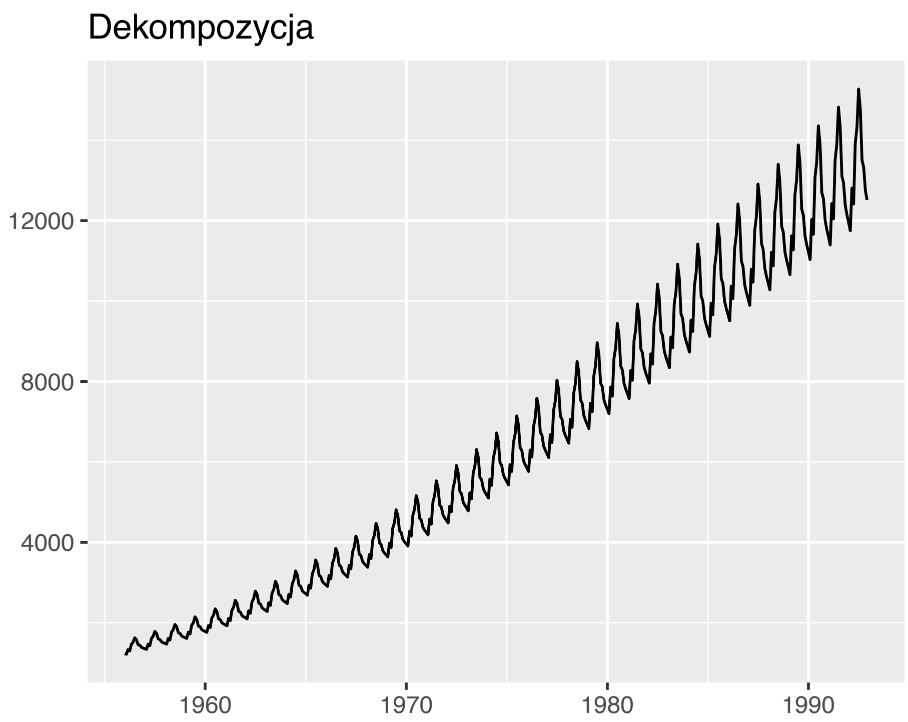

```r
library(expsmooth)

ses.alfa.optim <- ses(elec.train)$fitted
autoplot(ses.alfa.optim, main="Wygladzanie wykladnicze", lwd=.5)
```



```r
# dekompozycja na podstawie modelu regresji: trend kwadratowy + sezonowość
tslm.1 <- tslm(elec.train ~ trend + season+ I(trend ^ 2),lambda = 0)
summary(tslm.1)
```

```
##
## Call:
## tslm(formula = elec.train ~ trend + season + I(trend^2), lambda = 0)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.084808 -0.021988  0.000649  0.022311  0.099516
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.107e+00  7.111e-03 999.366  < 2e-16 ***
## trend        8.052e-03  4.982e-05 161.622  < 2e-16 ***
## season2     -2.021e-02  7.804e-03  -2.589 0.009939 **
## season3      6.381e-02  7.804e-03   8.177 3.30e-15 ***
## season4      2.986e-02  7.804e-03   3.826 0.000149 ***
## season5      1.416e-01  7.804e-03  18.150  < 2e-16 ***
## season6      1.683e-01  7.804e-03  21.562  < 2e-16 ***
## season7      2.299e-01  7.804e-03  29.462  < 2e-16 ***
## season8      1.937e-01  7.804e-03  24.820  < 2e-16 ***
## season9      1.019e-01  7.804e-03  13.051  < 2e-16 ***
## season10     8.667e-02  7.805e-03  11.104  < 2e-16 ***
## season11     3.993e-02  7.805e-03   5.116 4.71e-07 ***
## season12     1.864e-02  7.805e-03   2.388 0.017364 *
## I(trend^2)  -6.422e-06  1.084e-07 -59.242  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03357 on 430 degrees of freedom
## Multiple R-squared:  0.9976,Adjusted R-squared:  0.9976
## F-statistic: 1.39e+04 on 13 and 430 DF,  p-value: < 2.2e-16
```

```r
autoplot(tslm.1$fitted, main = "Dekompozycja")
```




```
Box.test(model_arima$fitted)

##
##  Box-Pierce test
##
## data:  model_arima$fitted
## X-squared = 434.48, df = 1, p-value < 2.2e-16

Box.test(ses.alfa.optim)

##
##  Box-Pierce test
##
## data:  ses.alfa.optim
## X-squared = 433.86, df = 1, p-value < 2.2e-16

Box.test(tslm.1$fitted.values)

##
##  Box-Pierce test
##
## data:  tslm.1$fitted.values
## X-squared = 433.74, df = 1, p-value < 2.2e-16
```

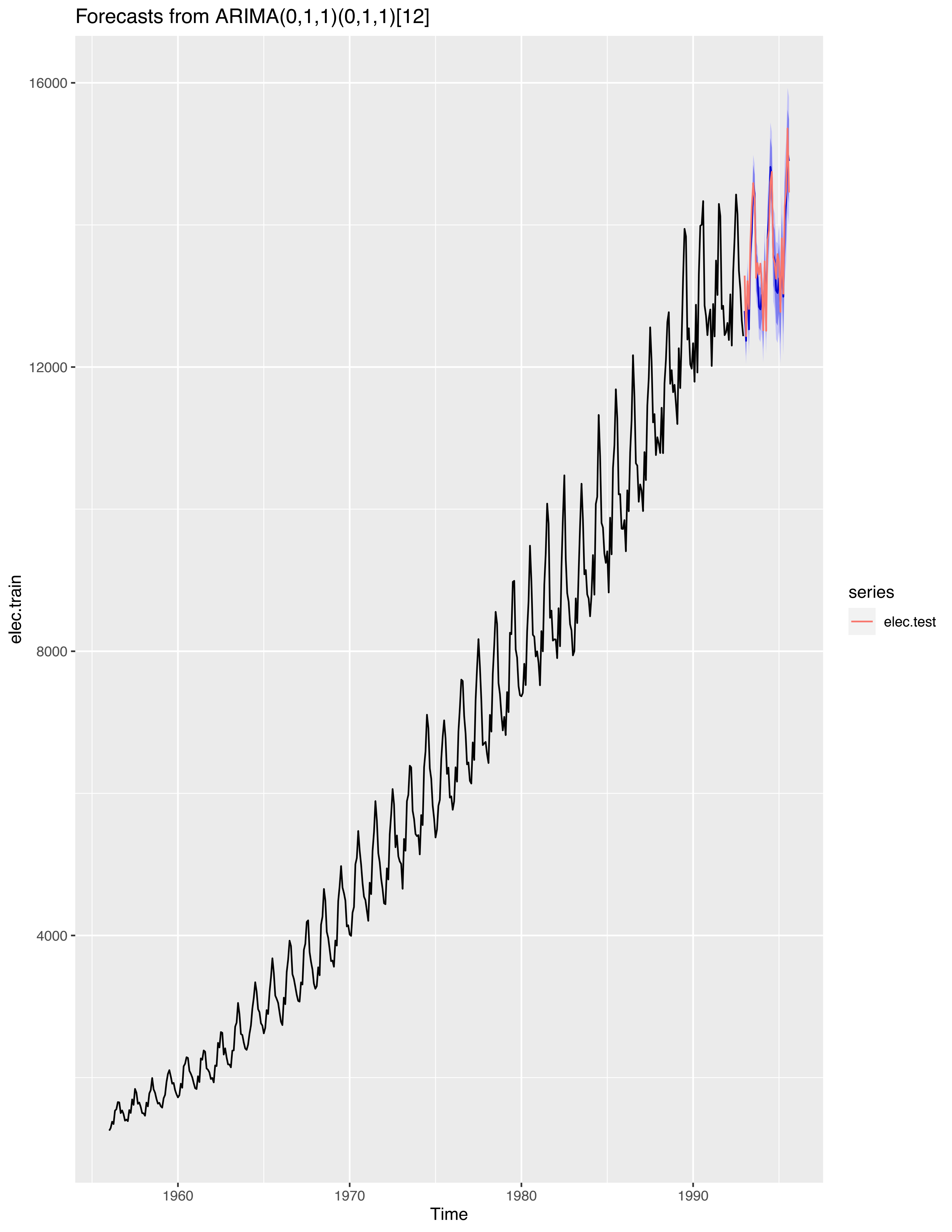
- Tworzymy model typu arima rzędu (0,1,1)
- Następnie korzystamy z algorytmu wygładzania wykładniczego przy optymalnym doborze parametru alfa

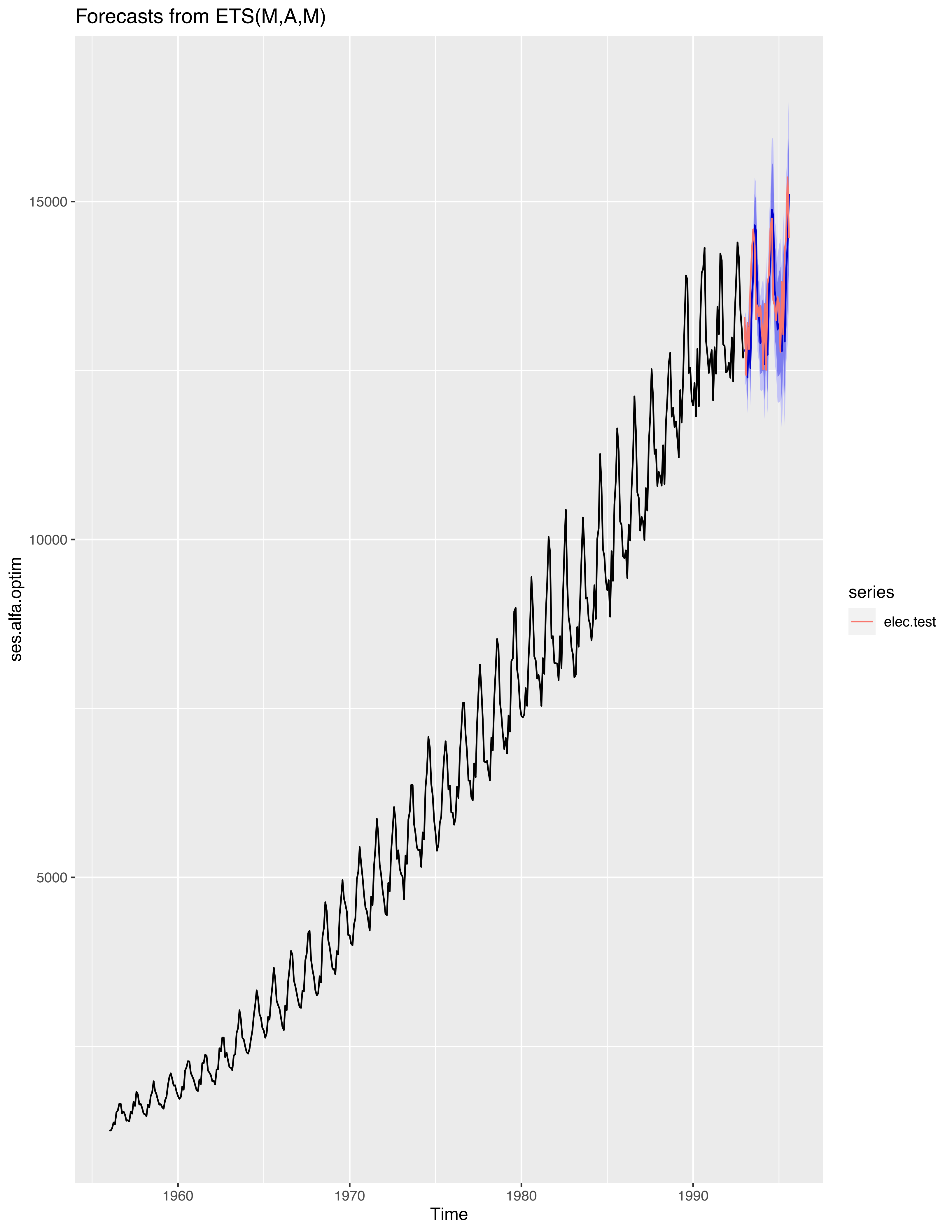
- Ostatni model bazuje na dekompozycji na podstawie modelu regresji uwzględniającym trend kwadratowy i sezonowość.
- Wartości testów statystycznych poświadczają sens modelów

2.3 Wyznaczenie prognoz dla modeli i przedstawienie ich

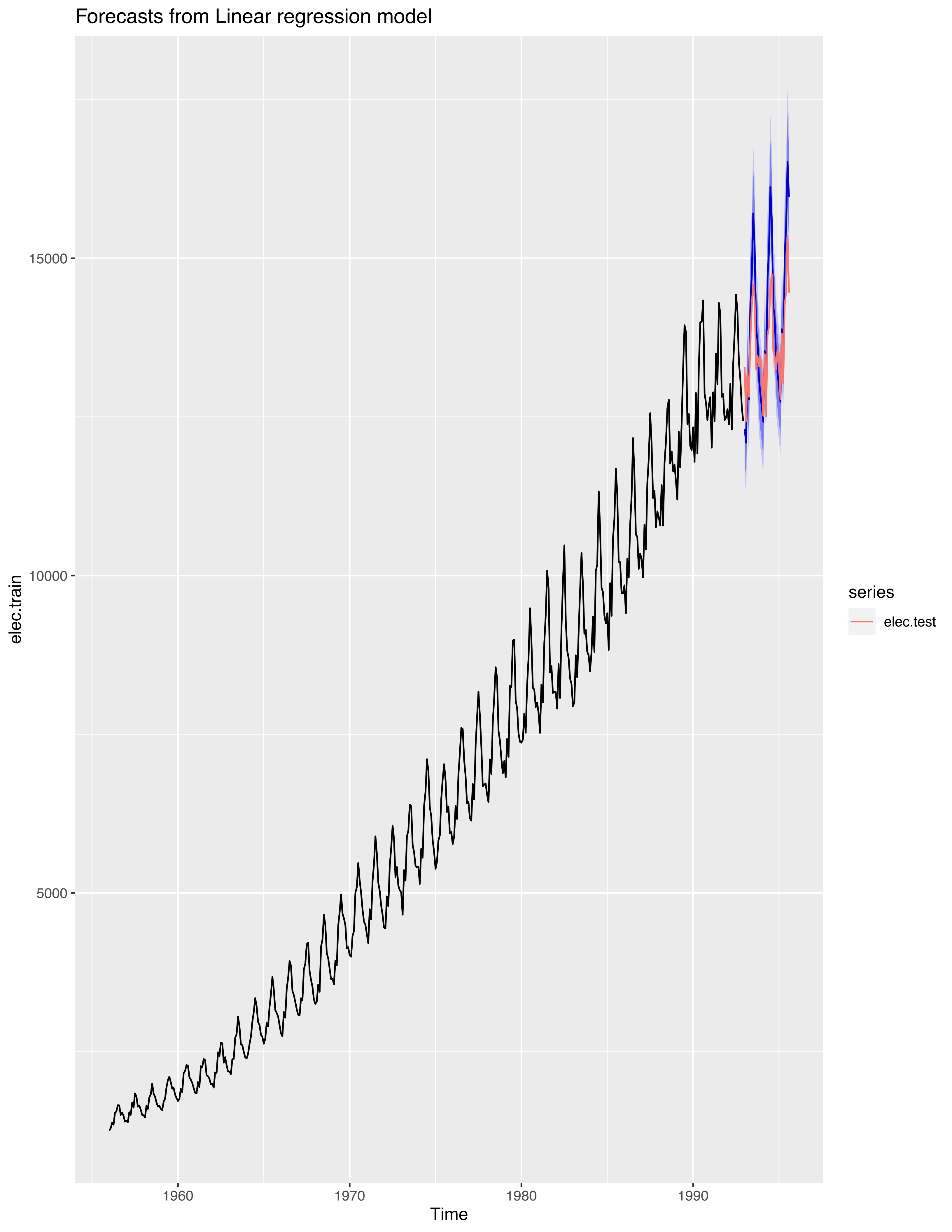
Na podstawie skonstruowanych modeli dokonujemy prognozy na zbiorze testowym. Budujemy również oparty na naiwnej metodzie prognozy (błądzenie losowe) w celu rzetelnego porównania.

```
h <- length(elec.test)
#model.naive<-snaive(elec.train)
arima.forecast <- forecast::forecast(model_arima, h = h)
ses.forecast <- forecast::forecast(ses.alfa.optim, h = h)
tslm.forecast <- forecast::forecast(tslm.1, h = h)
naive.forecast<-snaive(elec.train)
#autoplot(arima.forecast) +autolayer(gnp.test)
par(mfrow=c(2,2))
#autoplot(cbind(gnp.train,gnp.test))
autoplot(arima.forecast)+autolayer(elec.test)

## Error: Invalid input: date_trans works with objects of class Date only

autoplot(ses.forecast)+autolayer(elec.test)

## Error: Invalid input: date_trans works with objects of class Date only

autoplot(tslm.forecast)+autolayer(elec.test)

## Error: Invalid input: date_trans works with objects of class Date only

autoplot(naive.forecast)+autolayer(elec.test)

## Error: Invalid input: date_trans works with objects of class Date only
```


Forecasts from ARIMA(0,1,1)(0,1,1)[12]
16000
12000
8000
4000
elec.train
1960
1970
1980
1990
Time
series
elec.test

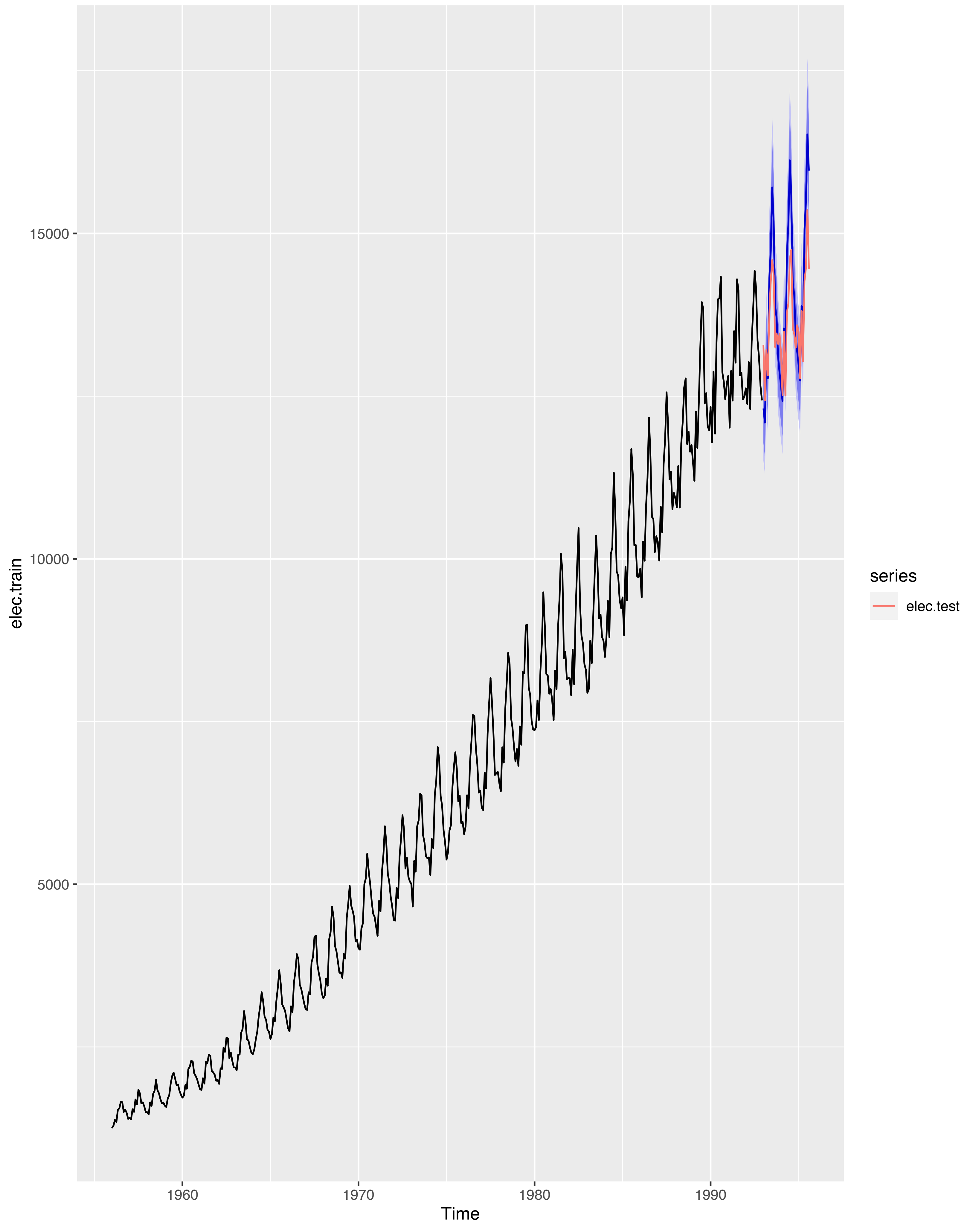


Forecasts from ETS(M,A,M)
15000
10000
5000
ses.alfa.optim
1960
1970
1980
1990
Time
series
elec.test



Forecasts from Linear regression model
15000
10000
5000
elec.train
1960
1970
1980
1990
Time
series
elec.test

Forecasts from Linear regression model



2.4 Porównanie dokładności

```
accuracy(model_arima$fitted,elec.train)

##                ME     RMSE      MAE        MPE     MAPE        ACF1 Theil's U
## Test set 1.637699 155.4112 108.4411 0.01632185 1.780561 0.007556606 0.3618897

accuracy(ses.alfa.optim,elec.train)

##                ME   RMSE      MAE       MPE     MAPE        ACF1 Theil's U
## Test set 26.61184 434.91 314.1734 0.3452704 4.943721 -0.01574756  1.001061

accuracy(tslm.1$fitted.values,elec.train)

##                  ME     RMSE      MAE         MPE     MAPE      ACF1 Theil's U
## Test set 0.07994717 252.8728 171.3304 -0.05458318 2.650288 0.7352673 0.5224678

accuracy(snaive(elec)$fitted,elec.train)

##                ME    RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set 324.1343 400.173 340.2546 5.863179 6.020125 0.5331916  1.052554

accuracy(arima.forecast,elec.test)

##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set   1.637699 155.4112 108.4411 0.01632185 1.780561 0.3187057
## Test set     103.670086 259.1988 199.3885 0.76316590 1.459976 0.5859979
##                     ACF1 Theil's U
## Training set 0.007556606        NA
## Test set     0.227134774 0.3442258
```

```r
accuracy(ses.forecast,elec.test)

##                      ME     RMSE       MAE        MPE     MAPE      MASE
## Training set  -1.267708 150.0385  97.41191 0.05527837 1.537711 0.2871615
## Test set     143.586378 708.4009 601.32984 0.93440461 4.401336 1.7726660
##                    ACF1 Theil's U
## Training set  0.2160129        NA
## Test set     -0.2096247  1.002657

accuracy(tslm.forecast,elec.test)

##                     ME    RMSE      MAE       MPE     MAPE MASE      ACF1
## Training set       Inf     Inf      Inf       NaN      NaN  NaN        NA
## Test set     -367.2003 733.272 601.5153 -2.544548 4.314066  NaN 0.7066701
##              Theil's U
## Training set        NA
## Test set     0.9752321

accuracy(naive.forecast,elec.test)

##                    ME     RMSE      MAE      MPE     MAPE    MASE      ACF1
## Training set 324.1343 400.1730 340.2546 5.863179 6.020125 1.00000 0.5331916
## Test set     397.0000 491.4618 405.4167 2.932716 2.996219 1.19151 0.1873200
##              Theil's U
## Training set        NA
## Test set     0.7219367
```

Opierając się głównie na najbardziej odppowiedniej miarze MASE widać że zdecydowanie najlepsze dopasowanie ma model ARIMA. Reszta typów miar podobnie wskazuje.

2.5 Wnioski dotyczące optymalnego modelu

Wszystkie testy i miary dokładności błędów jednomyślnie wskazują że najbardziej optymalnym modelem jest model typu ARIMA. Najgorszy wydaje się być model demompozycji, istnieje tutaj problem z miarami. Model oparty na algorytmie wygładzania wykładniczego również posiada dokładność na przyzwoitym poziomie- miary są niższe niż przy modelu naiwnym.