Analiza szeregów czasowych - raport 4.

Miłosz Kubera(249823)
Patryk Krukowski(249824)

9 czerwca 2021

Spis treści

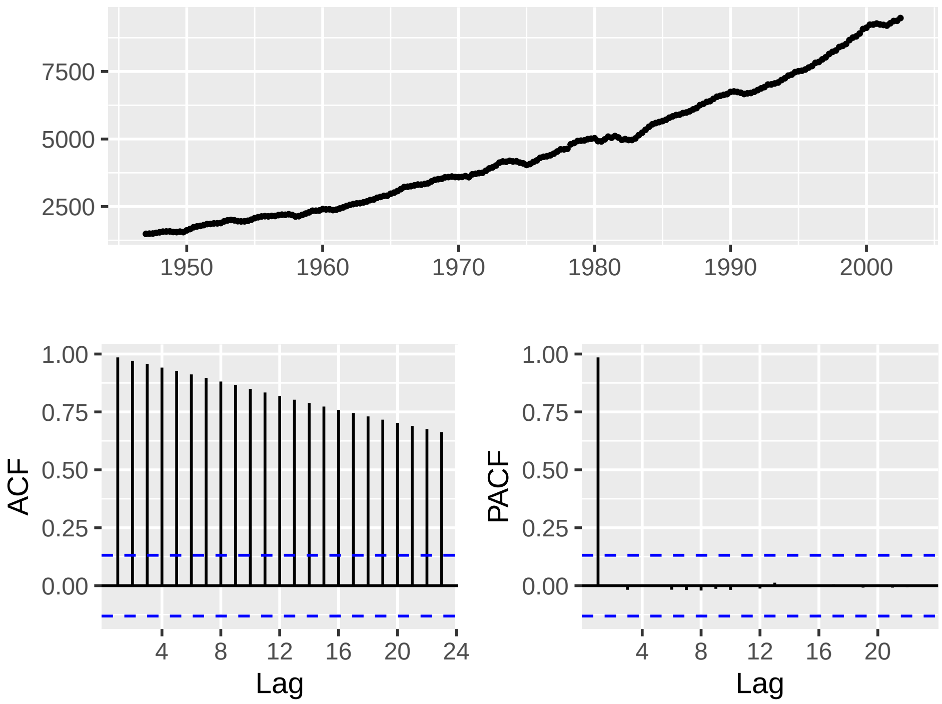
1 Zadanie 2.

1.1 Wstęp i opis eksperymentów

W tej części sprawozdania zajmiemy się dopasowaniem odpowiedniego modelu ARIMA do danych *gnp* (dane dochodu narodowego USA) dostępnych w pakiecie *astsa*. Analizowany szereg czasowy ma długość 223 oraz nie występują w nim obserwacje zakodowane jako *NA*. Wczytujemy odpowiednie dane, ustawiając jednocześnie parametr $frequency = 4$, jako że badane dane są kwartalne.

```
szereg <- ts(gnp, start=start(gnp), end=end(gnp), frequency=4)
length(szereg)

## [1] 223

anyNA(szereg)

## [1] FALSE
```

Zwizualizujmy rozważany szereg czasowy wraz z jego estymatorami funkcji ACF i PACF. Z rysunku 1 wynika, że w danych występuje trend długoterminowy. Ponadto nie obserwujemy sezonowości. Wykonane operacje/eksperymenty:



Rysunek 1: Szereg czasowy *gnp* wraz z estymatorami funkcji ACF i PACF

- Podział danych na część uczącą oraz testową - na części uczącej dopasujemy model, a na części testowej sprawdzimy jego zdolności predykcyjne.
- Identyfikacja transformacji niezbędnych do przekształcenia danych do postaci stacjonarnej, w szczególności użyjemy tranformacji potęgowej Boxa-Coxa oraz różnicowania.
- Identyfikacja modeli stacjonarnych, takich ja AR(p), MA(q), ARMA(p,q).
- Weryfikacja poprawności dopasowania modelu przeprowadzona na gruncie analizy reszt - użyjemy tutaj m.in. graficznego testu białoszumowości oraz testów losowości i normalności.
- Ocena istotności współczynników.
- Zastosowanie modelu ARIMA do konstrukcji prognoz dla zbioru testowego.
- Ocena dokładności wykonanych prognoz - użyjemy tutaj kryteriów, takich jak RMSE, MAE, MAPE i MASE oraz odniesiemy się do metod referencyjnych (prosta średnia ruchoma, metoda naiwna, metoda naiwna z dryfem).

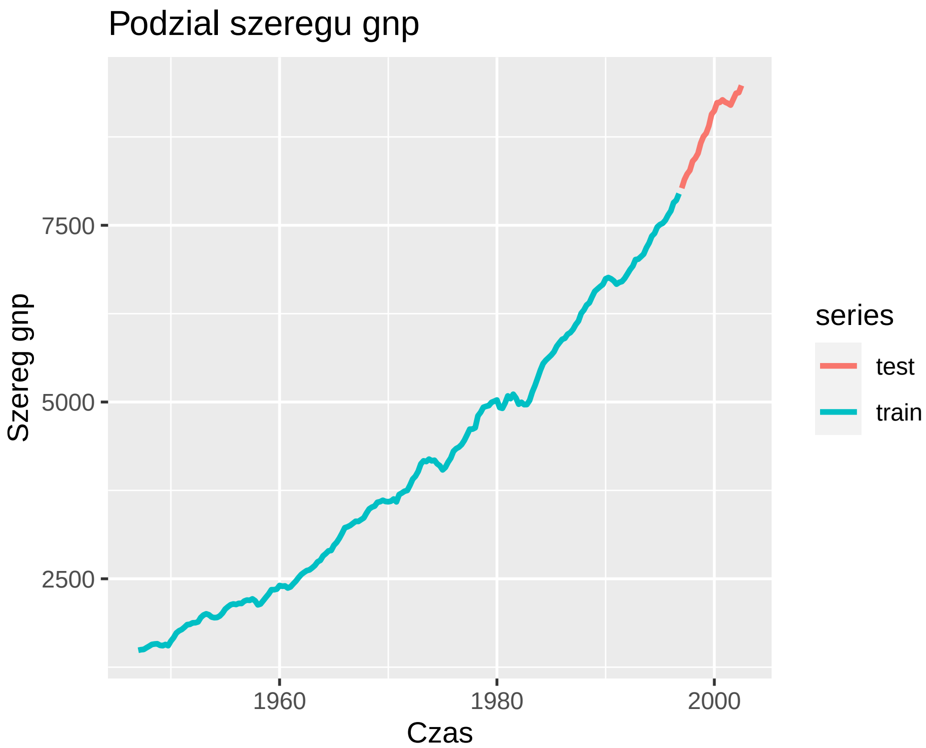
Po wstępnym przyjrzeniu się szeregowi czasowemu przystępujemy do właściwej analizy.

1.2 (a)

Dzielimy dane na część treningową oraz testową przy użyciu funkcji *window*.

```
train <- window(szereg, end=c(1996, 4))
test <- window(szereg, start=c(1997, 1))
length(test)/(length(test)+length(train))

## [1] 0.103139
```

Liczba obserwacji ze zbioru testowego stanowi około 10% całości wyjściowych danych.

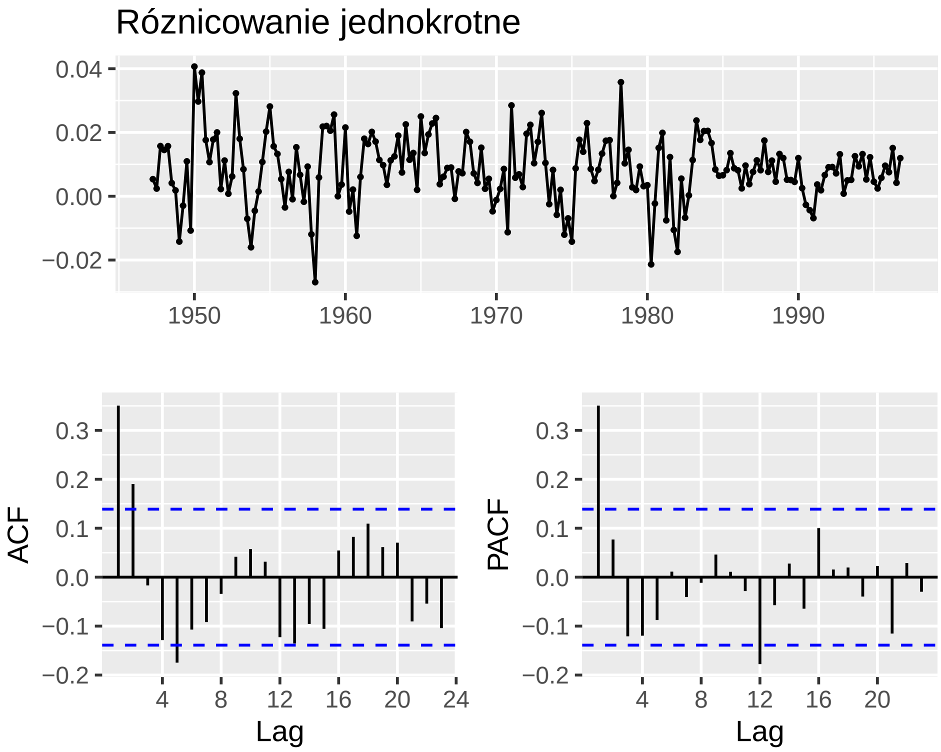


Rysunek 2: Podział danych na część uczącą i testową

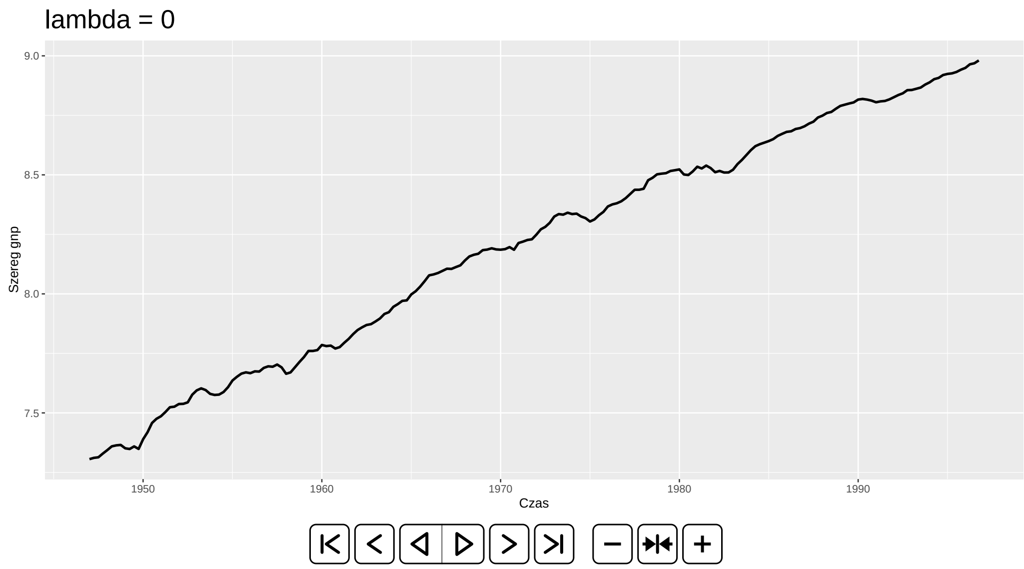
1.3 (b)

Zidentyfikujemy teraz transformacje umożliwiające przekształcenie danych do postaci stacjonarnej.

Analizując rysunek nr 1 dochodzimy do wniosku, że aby przekształcić dane do postaci stacjonarnej, warto użyć tranformacji potęgowej Boxa-Coxa. Prześledźmy w tym celu poniższą animację.



Rysunek 4: Efekt jednokrotnego różnicowania zlogarytmowanego szeregu *gnp*



Rysunek 3: Transformacje potęgowe Boxa-Coxa dla szeregu czasowego gnp

Transformacja logarytmiczna wyjściowego szeregu ($\lambda = 0$) jest już dla nas satysfakcjonująca, tzn. sprawia, że wariancja jest bardziej jednorodna, a trend, który początkowo przypominał trend kwadratowy, przypomina teraz trend liniowy - sugeruje to użycie różnicowania jednokrotnego. Rzeczywiście, jednokrotne różnicowanie zlogarytmowanego szeregu czsasowego *gnp* jest wystarczające. Przeprowadzone przez nas operacje doprowadziły nas do otrzymania szeregu, który możemy potraktować jako stacjonarny.

Z ciekawości sprawdzimy jeszcze, jaką krotność różnicowania oraz jaki parametr λ wybiorą dla nas wbudowane funkcje *BoxCox.lambda* oraz *ndiffs*.

```
train_log <- BoxCox(train, lambda=0)
BoxCox.lambda(train_log)

## [1] 1.999924

ndiffs(train_log)

## [1] 1
```

Funkcja *ndiffs* zwraca nam liczbę różnicowań taką, jaką sami wybraliśmy, natomiast funkcja *BoxCox.lambda* zwraca nam wartość $\lambda \sim 2$.

Wniosek 1 *Szereg otrzymany w wyniku zróżnicowania zlogarytmowanego szeregu czasowego gnp możemy uznać za stacjonarny.*

1.4 (c)

Przejdźmy do identyfikacji modeli stacjonarnych, korzystając z:

- ACF - wracając do rysunku nr 1.4 widzimy, że odpowiednim modelem wydaje się być model MA(2) lub MA(5) (rząd równy 5 jest bliski ”granicy nieistotności”).
- PACF - wracając do rysunku nr 1.4 widzimy, że odpowiednim modelem wydaje się być model AR(1) lub AR(12) z dokładnością do tego, że rząd równy 12 jest bliski granicy nieistotności. Ponadto byłby to także model o wiele bardziej złożony.

Funkcja ACF z rysunku nr zachowuje się jak tłumiona kombinacja sinusoid, co budzi nadzieje na dopasowanie modelu ARMA(p,q). W tym celu skorzystamy z funkcji *auto.arima* z pakietu *forecast*, wykorzystując minimalizację odpowiednich kryteriów:

- AIC

```
fit_arma_aic <- auto.arima(
    y=train,
    trace=FALSE,
    stepwise=FALSE,
    lambda=0,
    ic='aic',
    method='ML'
    )
fit_arma_aic

## Series: train
## ARIMA(2,1,2) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1      ar2      ma1     ma2   drift
##       1.3450  -0.7374  -1.0574  0.5523  0.0084
## s.e.  0.1438   0.1606   0.1970  0.2070  0.0009
```

```
##
## sigma^2 estimated as 9.528e-05:  log likelihood=641.53
## AIC=-1271.06   AICc=-1270.62   BIC=-1251.3
```

Uwaga 1 *W funkcji auto.arima ustawiamy parametr* $\lambda = 0$.

Dopasowanym modelem jest model ARIMA(2,1,2) z dryfem. Z definicji oznacza to, że do 1-krotnie zróżnicowanego zlogarytmowanego szeregu czasowego *gnp* (chodzi oczywiście o jego część treningową) dopasowalibyśmy model ARMA(2,2) z niezerową średnią.

- AICc

```
fit_arma_aicc <- auto.arima(
  y=train,
  trace=FALSE,
  stepwise=FALSE,
  lambda=0,
  ic='aicc',
  method='ML'
)
fit_arma_aicc

## Series: train
## ARIMA(2,1,2) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1      ar2      ma1     ma2   drift
##       1.3450  -0.7374  -1.0574  0.5523  0.0084
## s.e.  0.1438   0.1606   0.1970  0.2070  0.0009
##
## sigma^2 estimated as 9.528e-05:  log likelihood=641.53
## AIC=-1271.06   AICc=-1270.62   BIC=-1251.3
```

Dopasowanym modelem jest model ARIMA(2,1,2) z dryfem, czyli z definicji oznacza to, że do 1-krotnie zróżnicowanego zlogarytmowanego szeregu czasowego *gnp* dopasowalibyśmy model ARMA(2,2) z niezerową średnią.

- BIC

```
#po 1-krotnym zró■nicowaniu zlogarytmowanego szeregu
fit_arma_bic <- auto.arima(
  y=train,
  trace=FALSE,
  stepwise=FALSE,
  lambda=0,
  ic='bic',
```

```
  method='ML'
)
fit_arma_bic

## Series: train
## ARIMA(1,1,0) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1   drift
##       0.3491  0.0084
## s.e.  0.0662  0.0011
##
## sigma^2 estimated as 9.806e-05:  log likelihood=637.22
## AIC=-1268.44   AICc=-1268.32   BIC=-1258.56
```

Dopasowanym modelem jest model ARIMA(1,1,0) z dryfem, czyli z definicji oznacza to, że do 1-krotnie zróżnicowanego zlogarytmowanego szeregu czasowego *gnp* (dla ułatwienia oznaczmy ten szereg jako *gnp_log_diff*) dopasowalibyśmy model AR(1) z niezerową średnią.

Wniosek 2 *Do dalszej analizy z modeli ARMA wybieramy model ARMA(2,2), by modelować gnp_log_diff.*

Użyjemy teraz funkcji *auto.arima* do dopasowania modelu autoregresji oraz ruchomej średniej do *gnp_log_diff* przy użyciu minimalizacji kryteriów informacyjnych AIC, AICc oraz BIC. Najpierw model autoregresji:

- AIC

```
fit_ar_aic <- auto.arima(train,
                         trace=F,
                         lambda=0,
                         stepwise=F,
                         max.q=0,
                         ic='aic',
                         method='ML')
fit_ar_aic

## Series: train
## ARIMA(3,1,0) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1     ar2      ar3   drift
##       0.3314  0.1163  -0.1208  0.0084
## s.e.  0.0701  0.0736   0.0702  0.0010
##
## sigma^2 estimated as 9.702e-05:  log likelihood=639.28
## AIC=-1268.55   AICc=-1268.24   BIC=-1252.09
```

Wybranym modelem jest model ARIMA(3,1,0) z dryfem, czyli do szeregu *gnp_log_diff* dopasowalibyśmy model AR(3) z niezerową średnią.

- AICc

```
fit_ar_aicc <- auto.arima(train,
                          trace=F,
                          lambda=0,
                          stepwise=F,
                          max.q=0,
                          ic='aicc',
                          method='ML')
fit_ar_aicc
```

```
## Series: train
## ARIMA(1,1,0) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1   drift
##       0.3491  0.0084
## s.e.  0.0662  0.0011
##
## sigma^2 estimated as 9.806e-05:  log likelihood=637.22
## AIC=-1268.44   AICc=-1268.32   BIC=-1258.56
```

Wybranym modelem jest model ARIMA(1,1,0) z dryfem, czyli do szeregu *gnp_log_diff* dopasowalibyśmy model AR(1) z niezerową średnią.

- BIC

```
fit_ar_bic <- auto.arima(train,
                         trace=F,
                         lambda=0,
                         stepwise=F,
                         max.q=0,
                         ic='bic',
                         method='ML')
fit_ar_bic
```

```
## Series: train
## ARIMA(1,1,0) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1   drift
##       0.3491  0.0084
## s.e.  0.0662  0.0011
```

```
##
## sigma^2 estimated as 9.806e-05:  log likelihood=637.22
## AIC=-1268.44   AICc=-1268.32   BIC=-1258.56
```

Wybranym modelem jest model ARIMA(1,1,0) z dryfem, czyli do szeregu *gnp_log_diff* dopasowalibyśmy model AR(1) z niezerową średnią.

Wniosek 3 *Do dalszej analizy z modeli autoregresji wybieramy model AR(1) z niezerową średnią jako model najczęściej wybierany spośród rozważanych kryteriów informacyjnych, oraz jako model wybrany w wyniku analizy funkcji PACF.*

Podobnie, jak szybciej, używamy funkcji *auto.arima* do wyboru rzędu modelu ruchomej średniej MA(q).

- AIC

```
fit_ma_aic <- auto.arima(train,
                         trace=F,
                         lambda=0,
                         stepwise=F,
                         max.p=0,
                         ic='aic',
                         method='ML')
fit_ma_aic

## Series: train
## ARIMA(0,1,2)(0,0,1)[4] with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ma1     ma2     sma1   drift
##       0.3050  0.1994  -0.1065  0.0084
## s.e.  0.0699  0.0641   0.0755  0.0009
##
## sigma^2 estimated as 9.676e-05:  log likelihood=639.54
## AIC=-1269.08   AICc=-1268.77   BIC=-1252.61
```

Wybranym modelem jest model ARIMA(0,1,2)(0,0,1)[4] z dryfem, czyli do szeregu *gnp_log_diff* dopasowalibyśmy model ARMA(0,2)(0,1)[4] (z wariantem sezonowym) z niezerową średnią.

- AICc

```
fit_ma_aicc <- auto.arima(train,
                          trace=F,
                          lambda=0,
                          stepwise=F,
                          max.p=0,
```

```
                          ic='aicc',
                          method='ML')
fit_ma_aicc

## Series: train
## ARIMA(0,1,2) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ma1     ma2   drift
##       0.3073  0.2083  0.0084
## s.e.  0.0688  0.0685  0.0010
##
## sigma^2 estimated as 9.726e-05:  log likelihood=638.54
## AIC=-1269.08   AICc=-1268.87   BIC=-1255.91
```

Wybranym modelem jest model ARIMA(0,1,2) z dryfem, czyli do szeregu *gnp_log_diff* dopasowalibyśmy model MA(2) z niezerową średnią.

- BIC

```
fit_ma_bic <- auto.arima(train,
                         trace=F,
                         lambda=0,
                         stepwise=F,
                         max.p=0,
                         ic='bic',
                         method='ML')
fit_ma_bic

## Series: train
## ARIMA(0,1,2) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ma1     ma2   drift
##       0.3073  0.2083  0.0084
## s.e.  0.0688  0.0685  0.0010
##
## sigma^2 estimated as 9.726e-05:  log likelihood=638.54
## AIC=-1269.08   AICc=-1268.87   BIC=-1255.91
```

Wybranym modelem jest model ARIMA(0,1,2) z dryfem, czyli do szeregu *gnp_log_diff* dopasowalibyśmy model MA(2) z niezerową średnią.

Wniosek 4 *Do dalszej analizy z modeli ruchomej średniej wybieramy model MA(2) z niezerową średnią jako model najczęściej wybierany spośród rozważanych kryteriów informacyjnych, oraz jako model wybrany w wyniku analizy funkcji ACF.*

	ACF	PACF	AIC	AICc	BIC
ARMA(2,2)			+	+	-
AR(1)		+	-	+	+
MA(2)	+		-	+	+

Tabela 1: Przyjęte modele poprzez odpowiednie kryteria; znak "+" mówi nam o tym, że model został przyjęty, znak "-" mówi nam o tym, że dany model nie został przyjęty

Zatem tak, jak to wynika z tabeli 1, do modelowania *gnp_log_diff* używamy:

- ARMA(2,2)
- AR(1)
- MA(2)

Oczywiście, wszystkie powższe modele mają niezerową średnią, jako że parametr dryfu w odpowiadającym im modelach typu ARIMA jest niezerowy.

1.5 (d)

Zbadamy teraz dopasowanie modeli za pomocą analizy otrzymanych reszt.

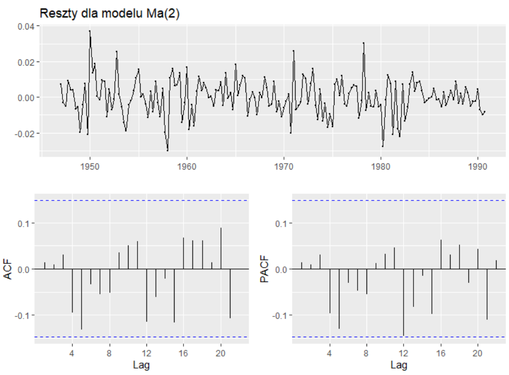
```
fit_arma <- Arima(train,
                  order=c(2,1,2),
                  seasonal=c(0,0,0),
                  lambda=0,
                  include.drift=TRUE
                  )

fit_ar <- Arima(train,
                order=c(1,1,0),
                seasonal=c(0,0,0),
                lambda=0,
                include.drift = TRUE
                )
fit_ma <- Arima(train,
                order=c(0,1,2),
                seasonal=c(0,0,0),
                lambda=0,
                include.drift = TRUE
)

#wyznaczamy reszty
res_ma <- fit_ma$residuals
res_ar <- fit_ar$residuals
res_arma <- fit_arma$residuals
```

(a) Metody graficzne:

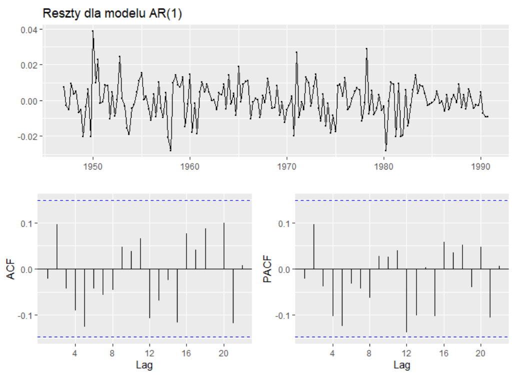
- Graficzny test białoszumowości

```
#sprawdzamy wyniki testu graficznego
graph_test(res_ar)[1]

## [[1]]
## [1] 0.96

graph_test(res_ma)[1]

## [[1]]
## [1] 0.96

graph_test(res_arma)[1]

## [[1]]
## [1] 0.96
```

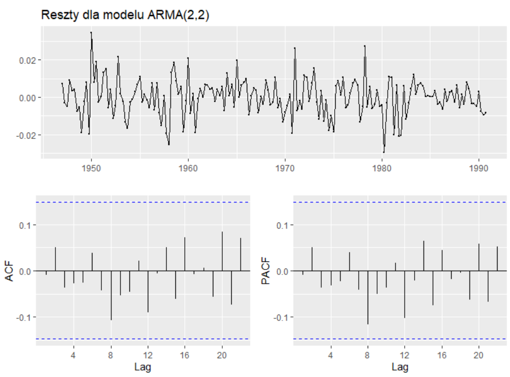
Jak widzimy, na poziomie ufności $\alpha = 0.05$ nie mamy podstaw do odrzucenia hipotezy zerowej, mówiącej o tym, że otrzymane reszty tworzą biały szum. Spójrzmy jeszcze na wykresy reszt narysowanymi wraz z ich estymatorami funkcji ACF i PACF. Rysunki



Rysunek 5: Wykresy dla reszt modelu MA(2) zwrócone przez funkcję *ggtsdisplay*



Rysunek 6: Wykresy dla reszt modelu AR(1) zwrócone przez funkcję *ggtsdisplay*

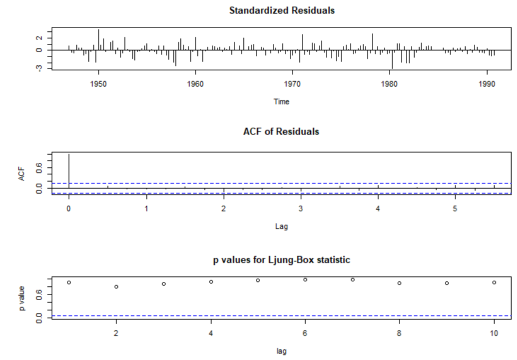


Rysunek 7: Wykresy dla reszt modelu ARMA(2,2) zwrócone przez funkcję *ggtsdisplay*

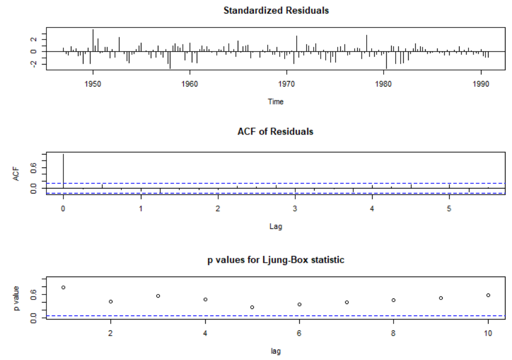
nr 5, 6 oraz 7 pokazują nam, że otrzymane reszty zachowują się, jak biały szum.

Wniosek 5 *Reszty modelów dopasowanych do gnp_log_diff zachowują się, jak biały szum.*

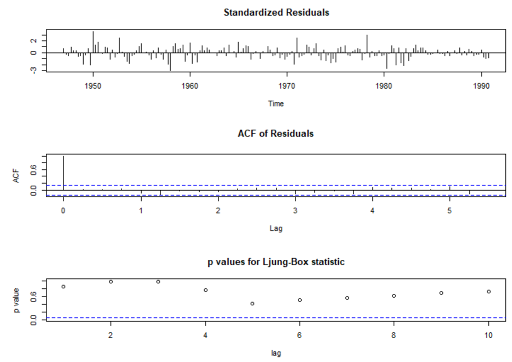
- P-wartości dla statystyki Ljungi-Boxa dla różnych opóźnień



Rysunek 8: Wykresy dla reszt modelu ARMA(2,2) zwrócone przez funkcję *tsdiag*



Rysunek 9: Wykresy dla reszt modelu AR(1) zwrócone przez funkcję *tsdiag*



Rysunek 10: Wykresy dla reszt modelu MA(2) zwrócone przez funkcję *tsdiag*

Najbardziej obiecująco wyglądają wykresy dla reszt modelu ARMA(2,2) - otrzymane p-wartości dla dowolnego obserwowanego przez nas opóźnienia (od 1 do 10) są bardzo bliskie 1. Jednak z dokładnością do tej uwagi, to tak naprawdę p-wartości wszystkich trzech modeli są sporo większe od przyjętego progu $\alpha = 0.05$.

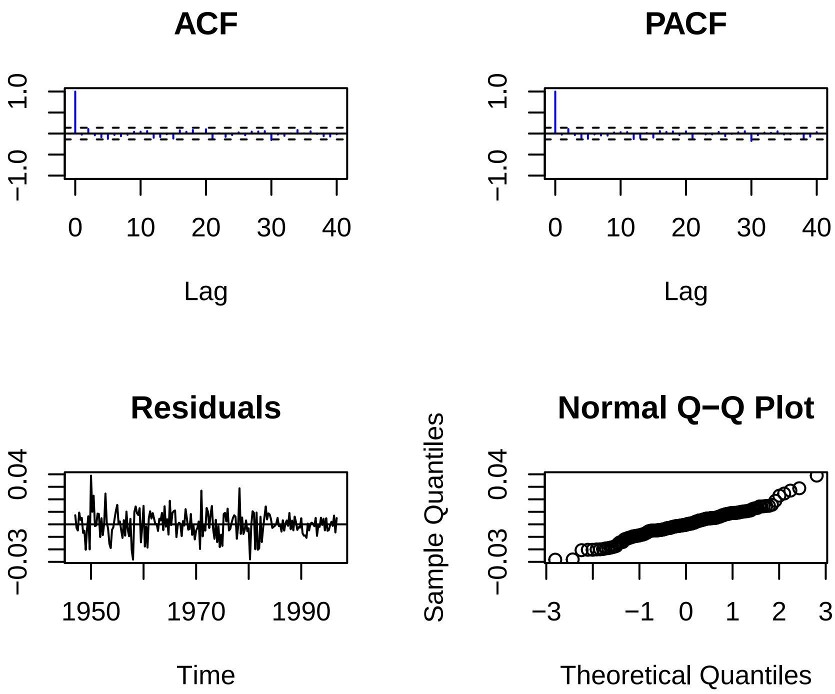
Uwaga 2 *Zdecydowaliśmy się użyć funkcji* tsdiag *zamiast* ggtsdiag *z pakietu* ggfortify *z powodu nadpisywania funkcji* autoplot *występującej zarówno w paczce* ggfortify, *jak i* forecast.

Wniosek 6 *Oczywiście wniosek jest taki sam, jak ten z powyższej "kropki". Zauważmy, że ta "kropka" nie wniosła nic nowego do diagnostyki modelu względem tej wcześniejszej - jest to swego rodzaju uzupełnienie tego, co zrobiliśmy szybciej.*

(b) Testy losowości

- dla reszt z modelu AR(1) (ARIMA(1,1,0) z dryfem)

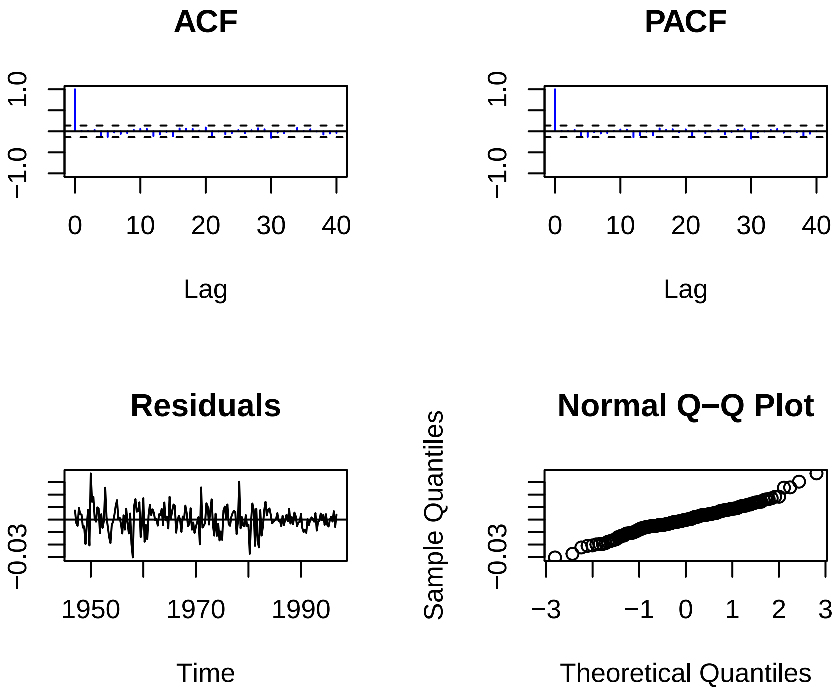
```
## Null hypothesis: Residuals are iid noise.
## Test                          Distribution Statistic   p-value
## Ljung-Box Q                  Q ~ chisq(20)     23.21    0.2787
## McLeod-Li Q                  Q ~ chisq(20)     21.39    0.3746
## Turning points T    (T-132)/5.9 ~ N(0,1)       143    0.0639
## Diff signs S        (S-99.5)/4.1 ~ N(0,1)      106    0.1122
## Rank P            (P-9950)/473.2 ~ N(0,1)     9435    0.2764
```



Żaden z powyższych testów (tzn. test Ljungi-Boxa, McLeod-Li, punktów zwrotnych, znakowy, rangowy) na poziomie istotności $\alpha = 0.05$ nie odrzuca hipotezy H_0: reszty tworzą biały szum.

- dla reszt z modelu MA(2) (ARIMA(0,1,2) z dryfem)

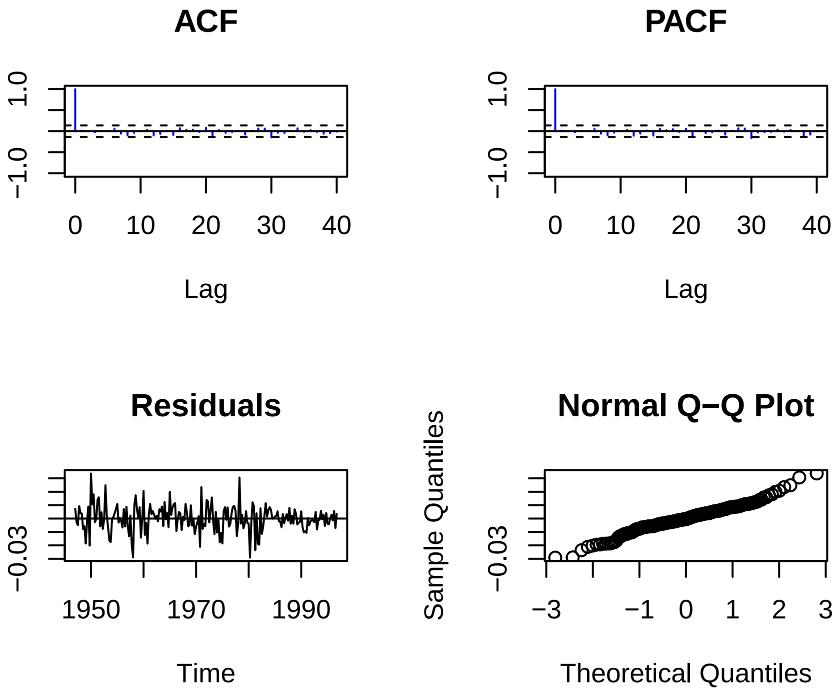
```
## Null hypothesis: Residuals are iid noise.
## Test                          Distribution Statistic   p-value
## Ljung-Box Q                  Q ~ chisq(20)     20.05    0.4549
## McLeod-Li Q                  Q ~ chisq(20)     22.34    0.3221
## Turning points T    (T-132)/5.9 ~ N(0,1)       135    0.6133
## Diff signs S        (S-99.5)/4.1 ~ N(0,1)      104    0.2715
## Rank P            (P-9950)/473.2 ~ N(0,1)     9439    0.2801
```



Podobnie, jak wyżej, żaden z powyższych testów (tzn. test Ljungi-Boxa, McLeod-Li, punktów zwrotnych, znakowy, rangowy) na poziomie istotności $\alpha = 0.05$ nie odrzuca hipotezy H_0: reszty tworzą biały szum.

- dla reszt z modelu ARMA(2,2) (ARIMA(2,1,2) z dryfem)

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     12.26     0.907
## McLeod-Li Q                Q ~ chisq(20)      21.3    0.3796
## Turning points T    (T-132)/5.9 ~ N(0,1)       133    0.8662
## Diff signs S       (S-99.5)/4.1 ~ N(0,1)       105     0.179
## Rank P           (P-9950)/473.2 ~ N(0,1)      9304    0.1722
```



I tak, jak szybciej - żaden z powyższych testów (tzn. test Ljungi-Boxa, McLeod-Li, punktów zwrotnych, znakowy, rangowy) na poziomie istotności $\alpha = 0.05$ nie odrzuca hipotezy H_0: reszty tworzą biały szum.

Wniosek 7 *Testy losowości potwierdzają jednogłośnie, że wszystkie otrzymane powyżej reszty zachowują się, jak biały szum.*

(c) Testy normalności:

- Test Jarque-Bera

```
##
##  Jarque Bera Test
##
## data:  res_ar
## X-squared = 17.055, df = 2, p-value = 0.000198
##
##
##  Skewness
##
## data:  res_ar
## statistic = 0.11289, p-value = 0.5146
##
##
##  Kurtosis
##
## data:  res_ar
## statistic = 4.4127, p-value = 4.543e-05
```

```
##
##  Jarque Bera Test
##
## data:  res_ma
## X-squared = 15.802, df = 2, p-value = 0.0003704
##
##
##  Skewness
##
## data:  res_ma
## statistic = 0.10349, p-value = 0.5502
##
##
##  Kurtosis
##
## data:  res_ma
## statistic = 4.3614, p-value = 8.495e-05
##
##  Jarque Bera Test
##
## data:  res_arma
## X-squared = 13.798, df = 2, p-value = 0.001009
##
##
##  Skewness
##
## data:  res_arma
## statistic = 0.0021007, p-value = 0.9903
##
##
##  Kurtosis
##
## data:  res_arma
## statistic = 4.2868, p-value = 0.0002036
```

Jak widzimy, na poziomie ufności $\alpha = 0.05$ test Jarque-Bera odrzuca H_0: rozkład reziduów jest normalny.

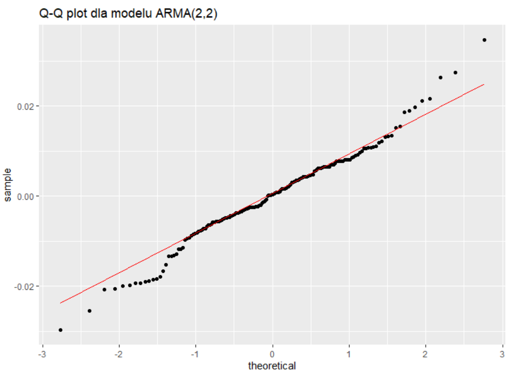
- Test Shapiro-Wilka

```
##
##  Shapiro-Wilk normality test
##
## data:  res_ar
## W = 0.97646, p-value = 0.001926
##
##  Shapiro-Wilk normality test
##
## data:  res_ma
```

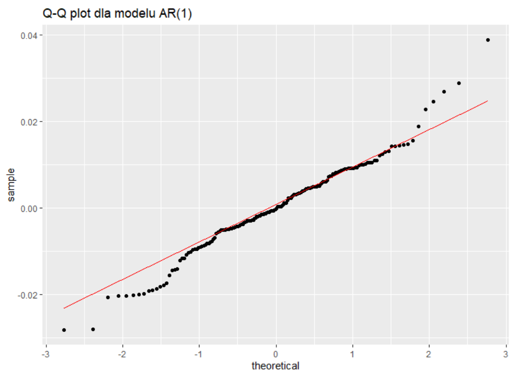
```
## W = 0.98118, p-value = 0.008763
##
##  Shapiro-Wilk normality test
##
## data:  res_arma
## W = 0.97861, p-value = 0.003799
```

Tak, jak wyżej, odrzucamy H_0: rozkład reziduów jest normalny na poziomie ufności $\alpha = 0.05$.

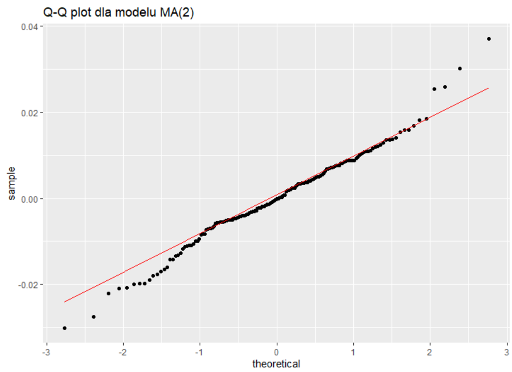
(d) Wykresy typu Q-Q ;rysunki nr 11, 12, 13 są do siebie bardzo podobne. Obserwujemy w



Rysunek 11: Wykresy typu Q-Q dla reszt modelu ARMA(2,2)



Rysunek 12: Wykresy typu Q-Q dla reszt modelu AR(1)



Rysunek 13: Wykresy typu Q-Q dla reszt modelu MA(2)

nich spore odstępstwo od rozkładu normalnego ”na początku i końcu wykresu”. Prawdopodobnie właśnie to zjawisko powoduje odrzucenie hipotezy o normalności przez testy statystyczne.

Wniosek 8 *Z analizy tej podsekcji wynika, że reszty rozważanych modeli tworzą biały szum, natomiast nie jest spełnione założenie o normalności.*

1.6 (e)

Skonstruujemy teraz tabelę ilustrującą porównanie jakości dopasowania modelów do danych *gnp_log_diff* w oparciu o kryteria informacyjne AIC, AICc oraz BIC.

	AIC	AICc	BIC
ARMA(2,2)	-1271.055	-1270.618	-1251.296
AR(1)	-1268.443	-1268.32	-1258.563
MA(2)	-1269.08	-1268.874	-1255.907

Tabela 2: Porównanie jakości dopasowania modelów do danych przy użyciu kryteriów informacyjnych AIC, AICc oraz BIC

Z tabeli nr 2 wynika, że nie ma jednoznacznej odpowiedzi na pytanie: "Który model jest najlepszy?" Wg AIC chronologicznie najlepsze są odpowiednio modele ARMA(2,2), MA(2), AR(1). AICc wybiera modele tak samo, jak AIC, natomiast wg BIC najlepiej dopasowanym modelem jest model AR(1), a potem odpowiednio MA(2) i ARMA(2,2). Jeżeli jakości dopasowania wyszły nam bardzo zbliżone do siebie, to być może warto by wybrać ten mniej złożony model - byłby on zapewne łatwiej interpretowalny.

1.7 (f)

Sprawdźmy, czy w dopasowanych modelach znajdują się współczynniki nieistotne.

- Model AR(1)

```
lmtest::coeftest(fit_ar)

##
## z test of coefficients:
##
##         Estimate Std. Error z value  Pr(>|z|)
## ar1   0.3491255  0.0661843   5.275 1.327e-07 ***
## drift 0.0084176  0.0010711   7.859 3.873e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

  Wszystkie współczynniki są istotne na poziomie ufności $\alpha = 0.05$.

- Model MA(2)

```
lmtest::coeftest(fit_ma)

##
## z test of coefficients:
##
##         Estimate Std. Error z value  Pr(>|z|)
## ma1   0.3072107  0.0688038  4.4650 8.006e-06 ***
## ma2   0.2082728  0.0684557  3.0424  0.002347 **
## drift 0.0084048  0.0010499  8.0055 1.190e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

  Wszystkie współczynniki są istotne na poziomie ufności $\alpha = 0.05$.

- ARMA(2,2)

```
lmtest::coeftest(fit_arma_aic)

##
## z test of coefficients:
##
##          Estimate   Std. Error z value  Pr(>|z|)
## ar1    1.34495315  0.14378022  9.3542 < 2.2e-16 ***
## ar2   -0.73739927  0.16055728 -4.5927 4.374e-06 ***
## ma1   -1.05742775  0.19700088 -5.3676 7.978e-08 ***
## ma2    0.55230879  0.20700043  2.6682  0.007627 **
## drift  0.00837865  0.00086447  9.6923 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Wszystkie współczynniki są istotne na poziomie ufności $\alpha = 0.05$.

Wniosek 9 *Na poziomie ufności równym $\alpha = 0.05$ rozważane przez nas modele zawierają tylko współczynniki statystycznie istotne.*

1.8 (g)

Odpowiedź jest taka sama, jak w podpunkcie (e) - nie można określić najlepszego modelu.

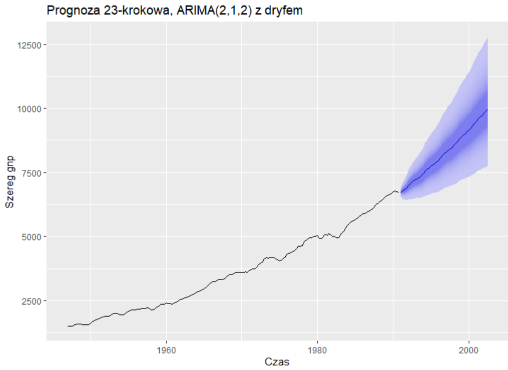
1.9 (h)

Na podstawie dopasowanych modeli ARIMA skonstruujemy teraz prognozy dla zbioru testowego z uwzględnieniem prognoz punktowych oraz przedziałowych. Przyjęliśmy horyzont prognozy równy 23. Wyznaczamy odpowiednie prognozy - z użyciem wykresów wachlarzowych, są to tak naprawdę przedziały predykcyjne narysowane dla różnych poziomów ufności. Wykresy wachlarzowe mają tę zaletę, że pozwalają przeanalizować różne możliwe scenariusze przyszłej prognozy.

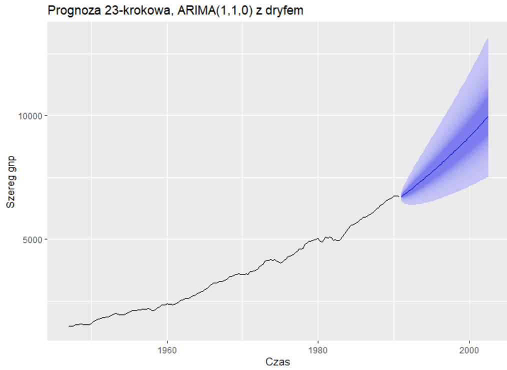
Wykresy prognoz z rysunków nr 14, 15, 16 są do siebie bardzo zbliżone, tzn. dla ustalonych poziomów ufności dopasowane modele wyznaczają prognozy, które na pierwszy rzut oka wydają się być bardzo podobne. Ta kwestia powinna się wyklarować w ostatniej podsekcji, w której wyznaczymy pewne miary dokładności wyznaczonych prognoz. Innymi słowy, skorzystamy z wiedzy, która dysponujemy w postaci zbioru testowego.

```
h <- length(test)

prediction_arma <- forecast::forecast(fit_arma, h=h)
prediction_ar <- forecast::forecast(fit_ar, h=h)
prediction_ma <- forecast::forecast(fit_ma, h=h)
```

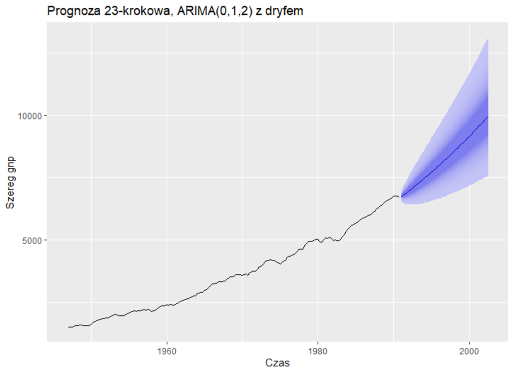
Wizualizujemy.



Rysunek 14: Prognoza 23-krokowa dla modelu ARIMA(2,1,2) z dryfem dopasowanego do treningowej części szeregu czasowego *gnp*



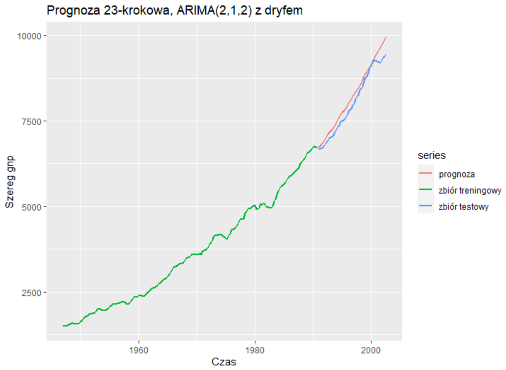
Rysunek 15: Prognoza 23-krokowa dla modelu ARIMA(1,1,0) z dryfem dopasowanego do treningowej części szeregu czasowego *gnp*



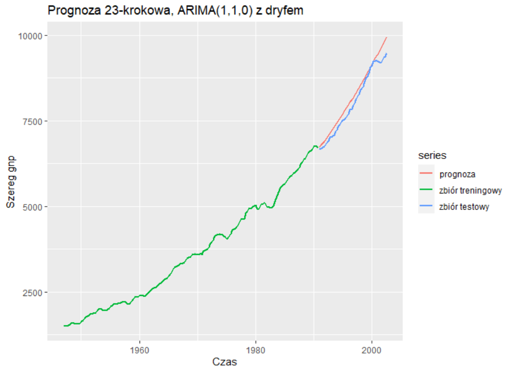
Rysunek 16: Prognoza 23-krokowa dla modelu ARIMA(0,1,2) z dryfem dopasowanego do treningowej części szeregu czasowego *gnp*

Warto także porównać bezpośrednio prognozowane wartości z tymi obserwowanymi (rysunki nr 17, 18, 19).

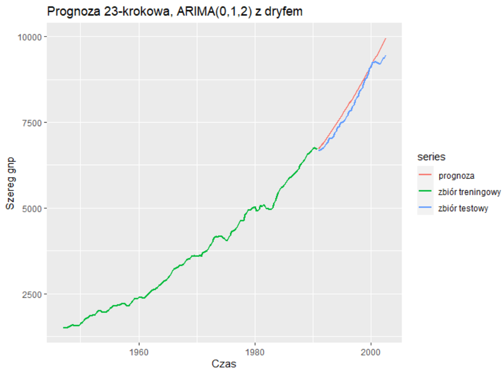
Generalnie uzyskane prognozy są dobre - zauważamy we wszystkich przypadkach jednak mały problem z prognozą ostatnich wartości, nie jest to jednak nic dziwnego i poniekąd zgodne z intuicją. Obserwację tę potwierdzają także coraz szersze przedziały predykcyjne (w miarę upływu czasu).



Rysunek 17: Prognoza 23-krokowa dla modelu ARIMA(2,1,2) z dryfem dopasowanego do treningowej części szeregu czasowego *gnp* - bezpośrednie porównanie



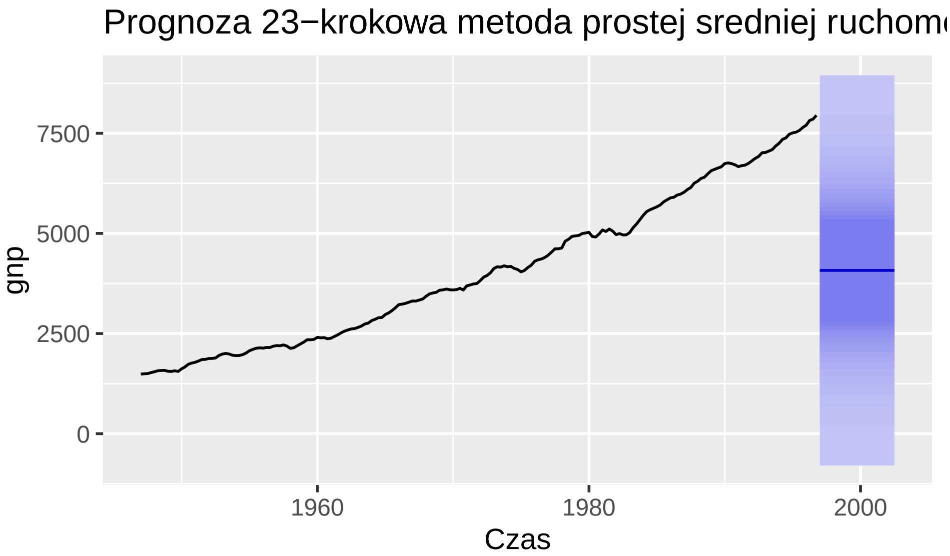
Rysunek 18: Prognoza 23-krokowa dla modelu ARIMA(1,1,0) z dryfem dopasowanego do treningowej części szeregu czasowego *gnp* - bezpośrednie porównanie



Rysunek 19: Prognoza 23-krokowa dla modelu ARIMA(0,1,2) z dryfem dopasowanego do treningowej części szeregu czasowego *gnp* - bezpośrednie porównanie

Użyjemy teraz prostych metod prognozowania jako metod referencyjnych. Służą one głównie do tego, by nie "przekombinować" analizy. Warto się upewnić, że użyte narzędzia rzeczywiście prowadzą do uzyskania trafniejszych wyników.

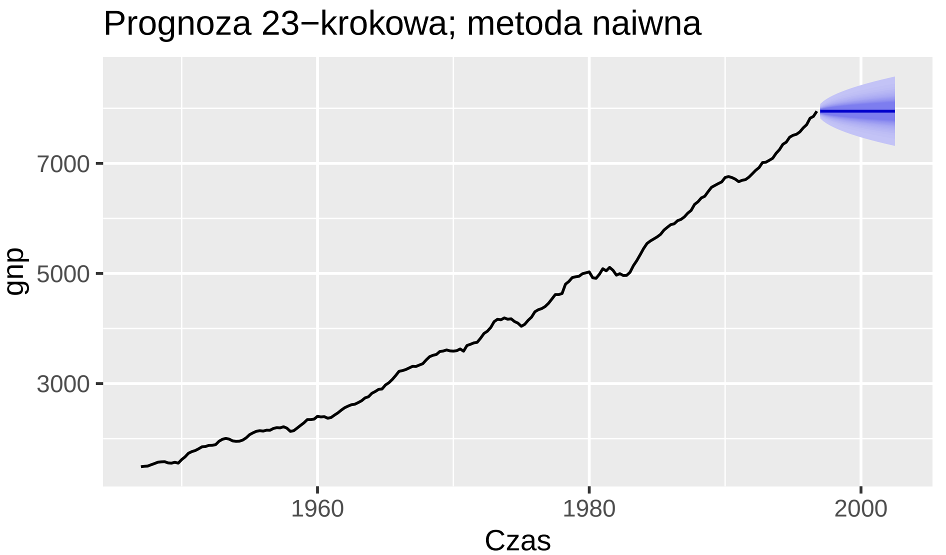
- Prosta średnia ruchoma



Rysunek 20: 23-krokowa prognoza metodą prostej średniej ruchomej dla części testowej szeregu czasowego gnp

Jak mogliśmy się tego spodziewać, prosta średnia ruchoma działa **fatalnie**.

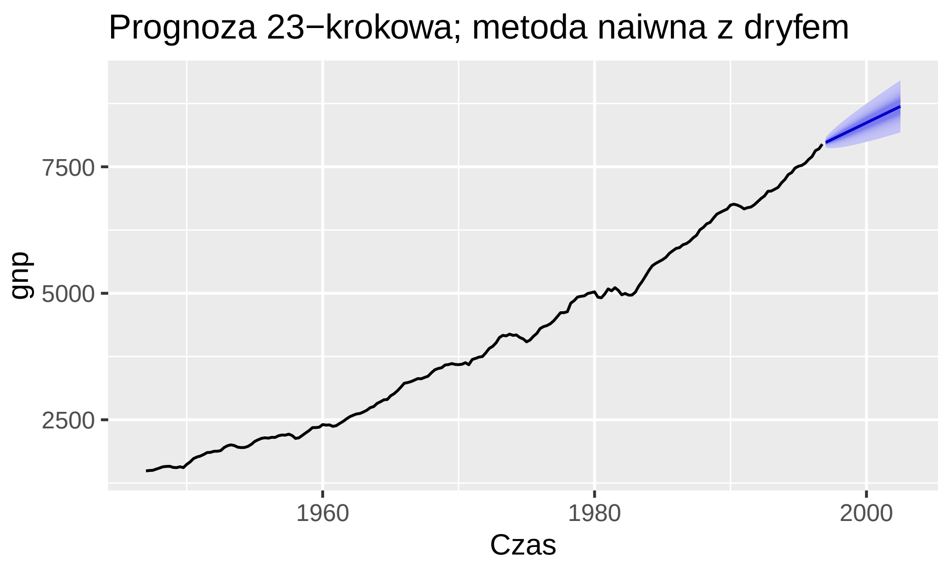
- Metoda naiwna



Rysunek 21: 23-krokowa prognoza metodą naiwną dla części testowej szeregu czasowego gnp

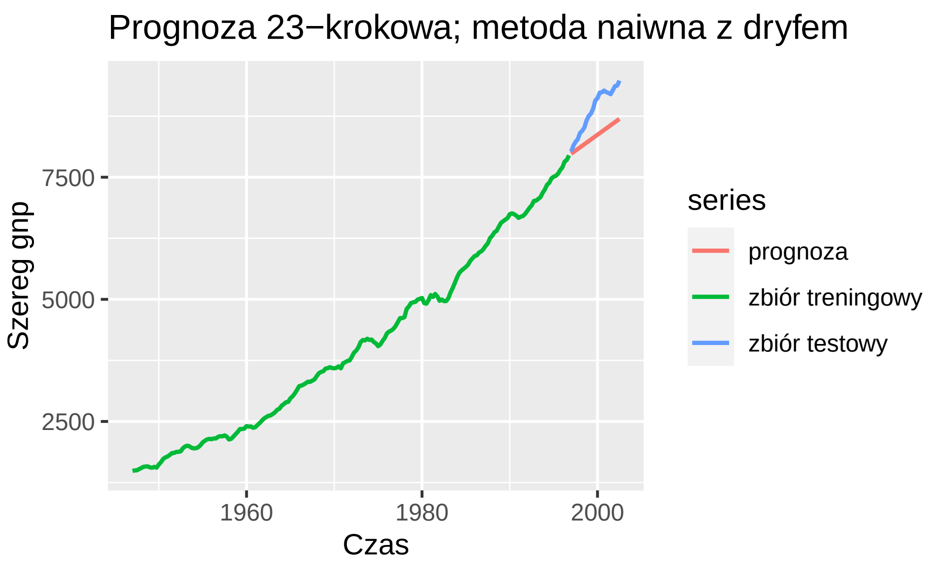
Metoda naiwna także nie działa dobrze i prowadzi do tzw. płaskich prognoz.

- Metoda naiwna uwzględniająca dryf



Rysunek 22: 23-krokowa prognoza metodą naiwną uwzględniającą dryf dla części testowej szeregu czasowego gnp

Ta metoda działa najlepiej z prostych metod prognozwania. Mimo to także nie działa dobrze, popatrzmy w tym celu na poniższy wykres.



Rysunek 23: Porównanie 23-krokowej prognozy dla zbioru testowego metodą naiwną z dryfem vs obserwowane wartości szeregu czasowego

Wyznaczona prognoza jest daleka od bycia dobrą, mimo iż na pierwszy rzut oka mogło wydawać się inaczej.

Wniosek 10 *Użycie bardziej skomplikowanych narzędzi w celu wyznaczenia prognoz dla badanego szeregu czasowego jest jak najbardziej uzasadnione.*

1.10 (i)

Tak, jak zapowiadaliśmy szybciej, ocenimy teraz dokładność skonstruowanych prognoz na bazie zbioru testowego i uczącego, i porównany je między sobą na tej płaszczyźnie.

- Dla Arima(2,1,2) z dryfem

```
##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 89.04332 165.0757 144.4849 1.017266 1.610199 0.8513196  1.924252
```

- Dla ARIMA(1,1,0) z dryfem

```
##                 ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set 58.90946 152.1949 133.0473 0.6803137 1.473759 0.8542605  1.761259
```

- Dla ARIMA(0,1,2) z dryfem

```
##                 ME     RMSE     MAE       MPE     MAPE      ACF1 Theil's U
## Test set 69.77665 156.1216 137.281 0.8021882 1.524485 0.8544852   1.81159
```

Uzyskane wyniki świadczą o tym, że najlepiej dopasowane prognozy zostały wyznaczone dla modelu ARIMA(1,1,0) z dryfem oraz dla ARIMA(0,1,2) z dryfem. Co prawda, współczynnik *U Theila* jest większy od 1 dla każdej z metod, co być może nie jest zbyt optymistyczne, ale nie przejmujemy się tym - może to mieć związek z długim horyzontem prognozy.

Wniosek 11 *Do konstrukcji prognoz najlepiej jest wykorzystać modele ARIMA(1,1,0) z dryfem oraz ARIMA(0,1,2) z dryfem, tym bardziej, że nie są to modele złożone. Co prawda model ARIMA(2,1,2) z dryfem także radzi sobie dobrze oraz został wybrany przez dwa z trzech kryteriów informacyjnych, ale jest on cięższej interpretowalny oraz wymaga większej ilości współczynników estymacji.*