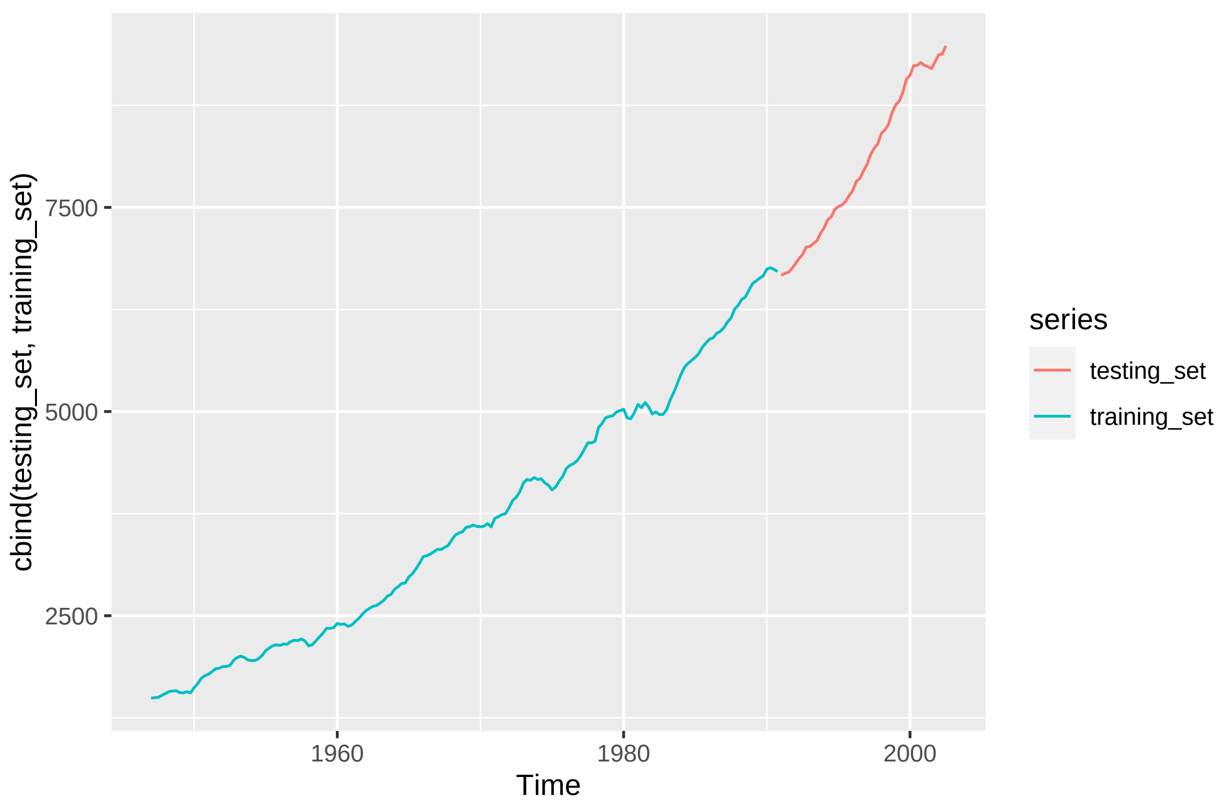
Dopasowanie modelu ARIMA do szeregu gnp

Paweł Matławski i Michał Liszkowski

20 06 2021

Podział danych na część uczącą i testową

```
training_set <- window(df, end = c(1990,4))
testing_set <- window(df, start = c(1991,1))
autoplot(cbind(testing_set, training_set))
```

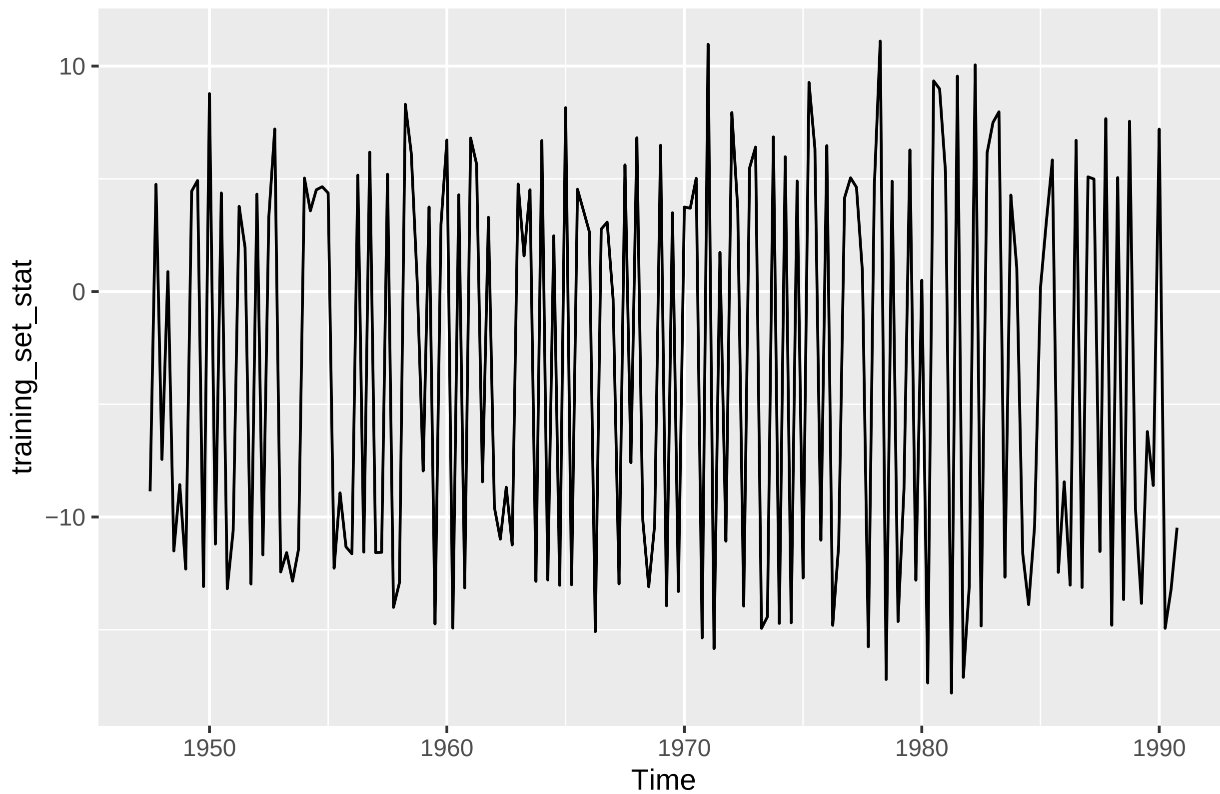


Przekształcenie do postaci stacjonarnej szeregu

```
lambda <-  BoxCox.lambda(training_set)
ndiffs(training_set)
```

```
## [1] 2
```

```r
training_set_stat <- diff(training_set, differences = 2)
training_set_stat <- BoxCox(training_set_stat, lambda = lambda)
autoplot(training_set_stat)
```



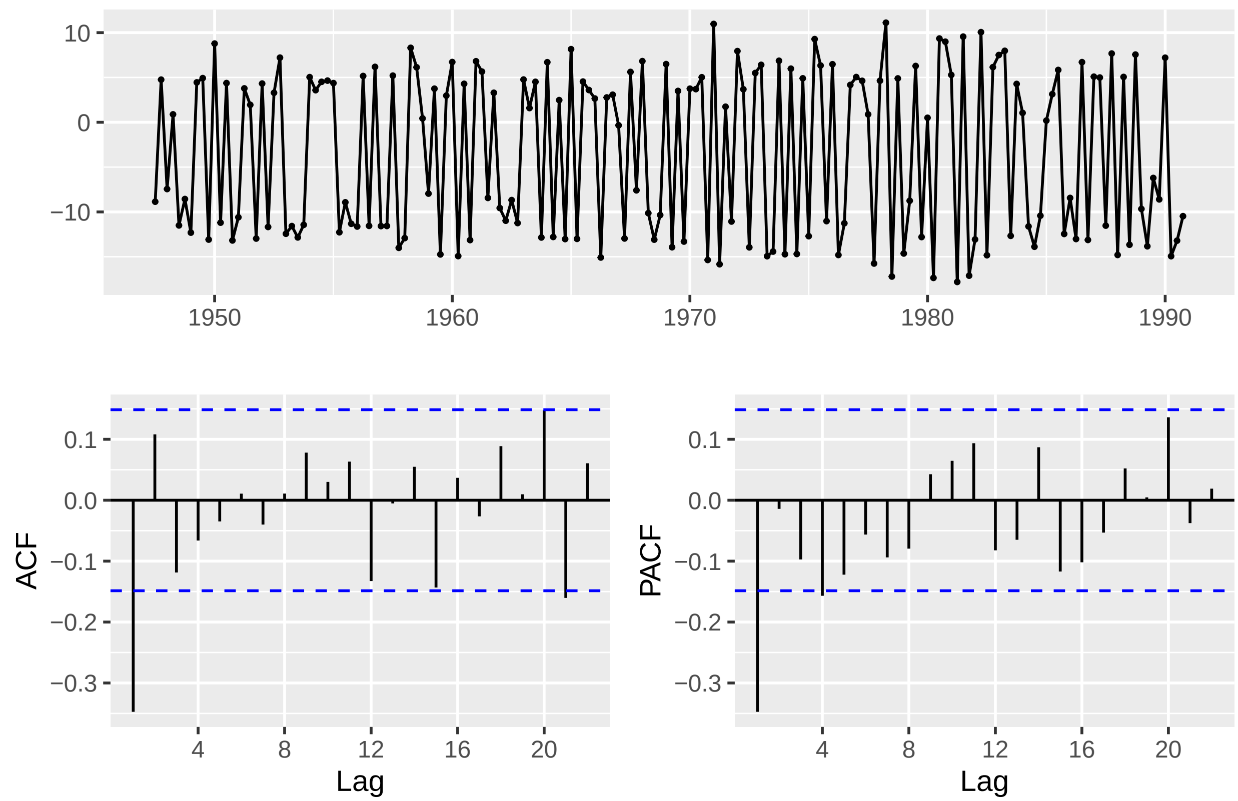

```r
Box.test(training_set_stat)
```

```
##
##  Box-Pierce test
##
## data:  training_set_stat
## X-squared = 20.99, df = 1, p-value = 4.617e-06
```

Wstępna identyfikacja modeli stacjonarnych

Wybór AR i MA na podstawie ACF i PACF

```r
ggtsdisplay(training_set_stat)
```




```r
ma21.fit <- Arima(training_set, order = c(0,1,21))
ar20.fit <- Arima(training_set, order = c(20,1,0))
```

Na podstawie wykresów funkcji autokorelacji i funkcji cząstkowej autokorelacji będę rozważał w dalszych etapach nasz szereg jako szereg $MA(21)$ lub $AR(20)$.

Wybór rzędów dla modeli ARMA na podstawie kryteriów informacyjnych

```r
arima.fit <- auto.arima(training_set)
arima.fit
```

```
## Series: training_set
## ARIMA(1,2,1)(0,0,2)[4]
##
## Coefficients:
##          ar1      ma1     sma1     sma2
##       0.3440  -0.9717  -0.0323  -0.1348
## s.e.  0.0759   0.0196   0.0793   0.0717
##
## sigma^2 estimated as 1345:  log likelihood=-872.97
## AIC=1755.94   AICc=1756.3   BIC=1771.74
```

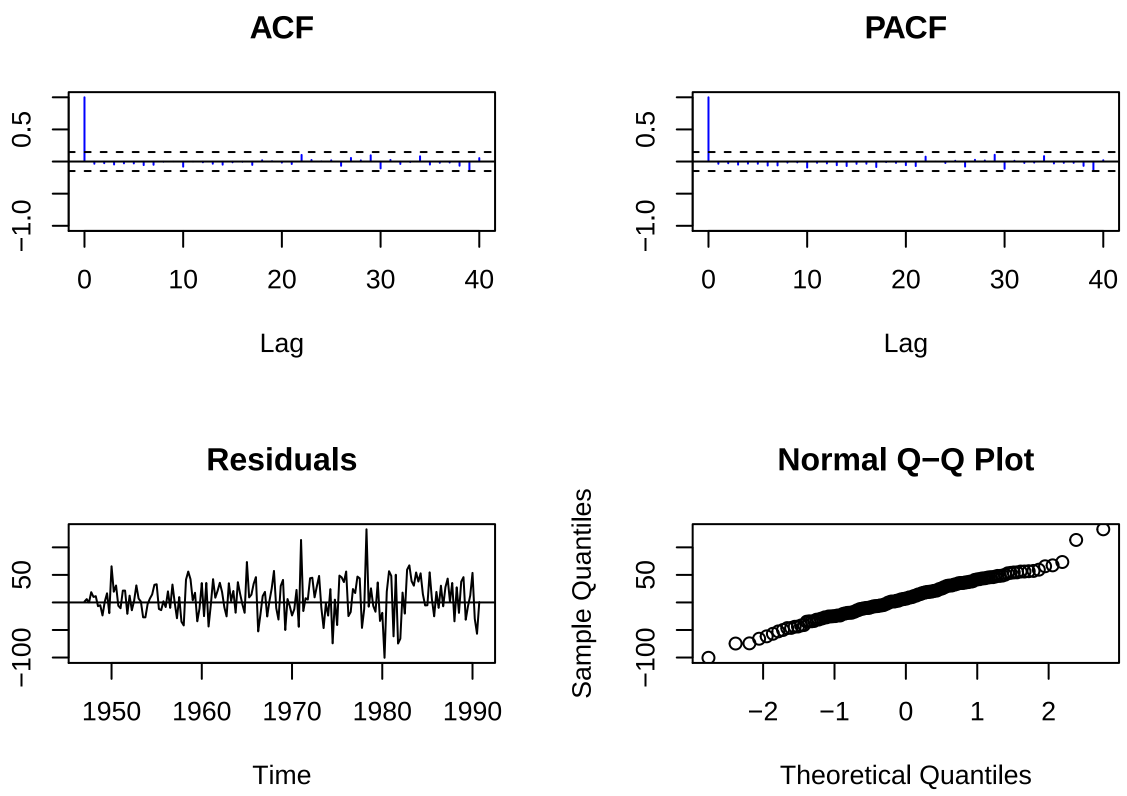
Automatyczne dopasowanie dało nam model $ARMA(1,2,1)$

Badanie poprawności dopasowania modeli na podstawie analizy reszt

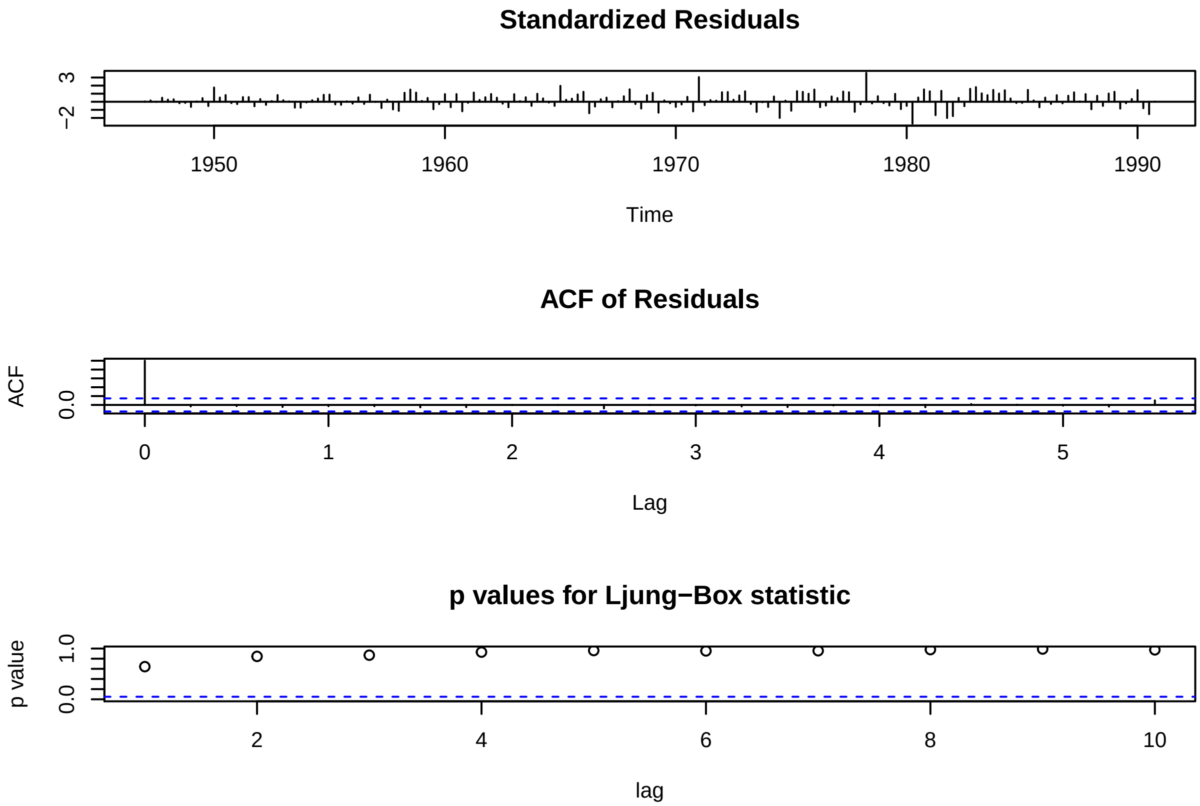
MA(21)

```
test(residuals(ma21.fit))
```

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)      4.71    0.9998
## McLeod-Li Q                Q ~ chisq(20)     31.71    0.0465 *
## Turning points T    (T-116)/5.6 ~ N(0,1)       120    0.4723
## Diff signs S       (S-87.5)/3.8 ~ N(0,1)        92    0.2413
## Rank P           (P-7700)/390.8 ~ N(0,1)      8102    0.3036
```




```
tsdiag(ma21.fit)
```

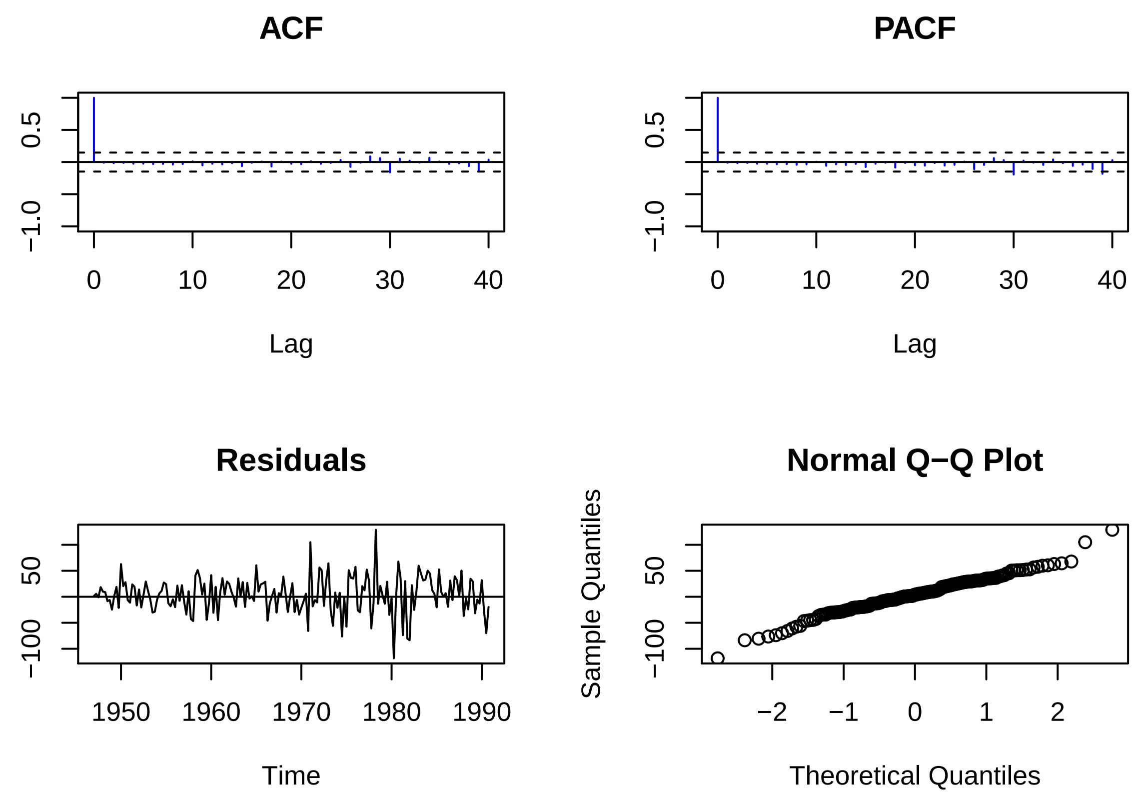


Analizując reszty dla tego modelu, nie mamy podstaw do odrzucenia hipotezy zerowej, że reszty są szumem i.i.d, zatem model nasz wydaje się być sensownym dopasowaniem.

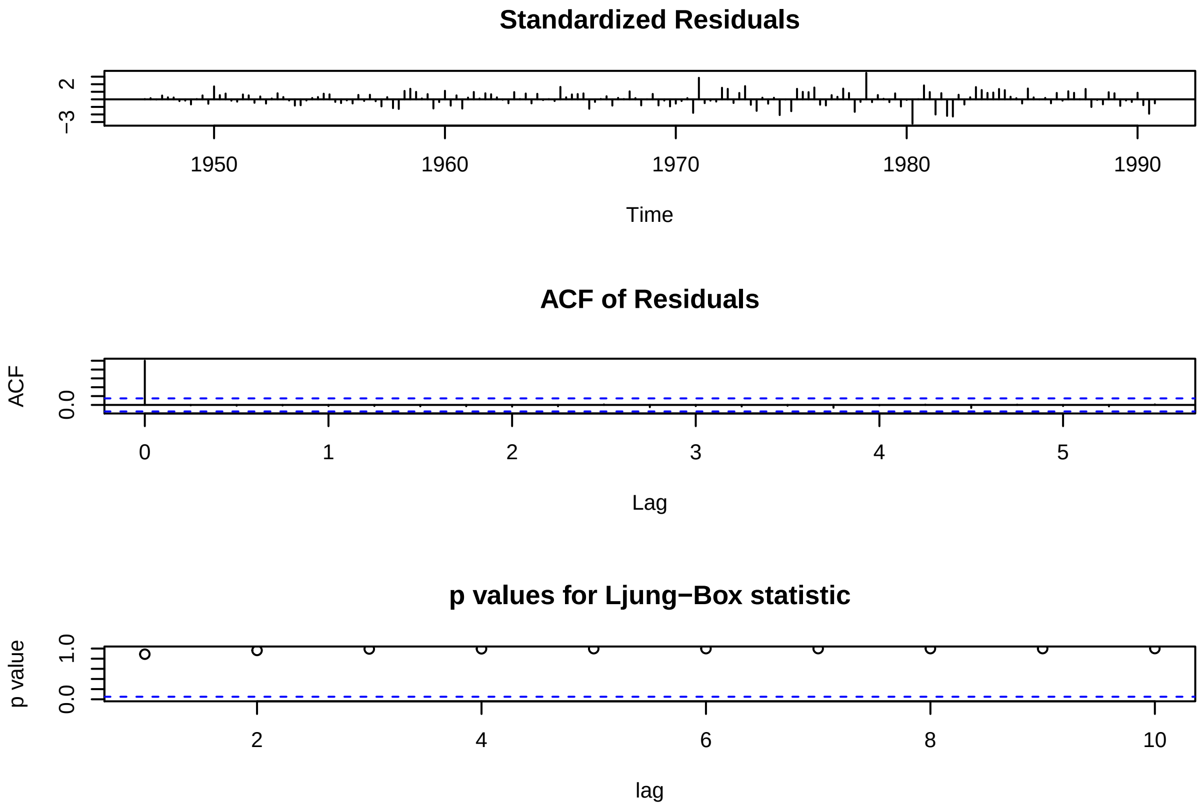
AR(20)

```
test(residuals(ar20.fit))
```

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)      3.99         1
## McLeod-Li Q                Q ~ chisq(20)     46.77     6e-04 *
## Turning points T    (T-116)/5.6 ~ N(0,1)       122    0.2809
## Diff signs S       (S-87.5)/3.8 ~ N(0,1)        83    0.2413
## Rank P          (P-7700)/390.8 ~ N(0,1)      7802    0.7941
```




```
tsdiag(ar20.fit)
```

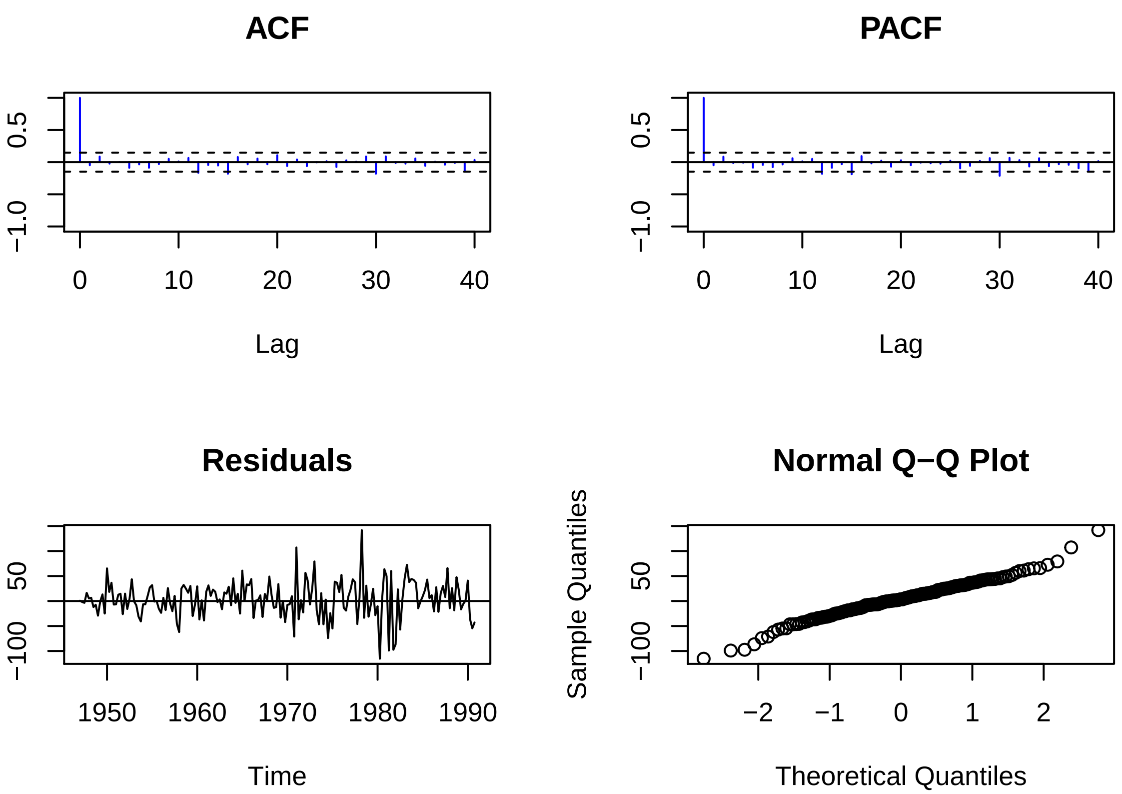


Analizując reszty dla tego modelu, nie mamy podstaw do odrzucenia hipotezy zerowej, że reszty są szumem i.i.d, zatem model nasz wydaje się być sensownym dopasowaniem.

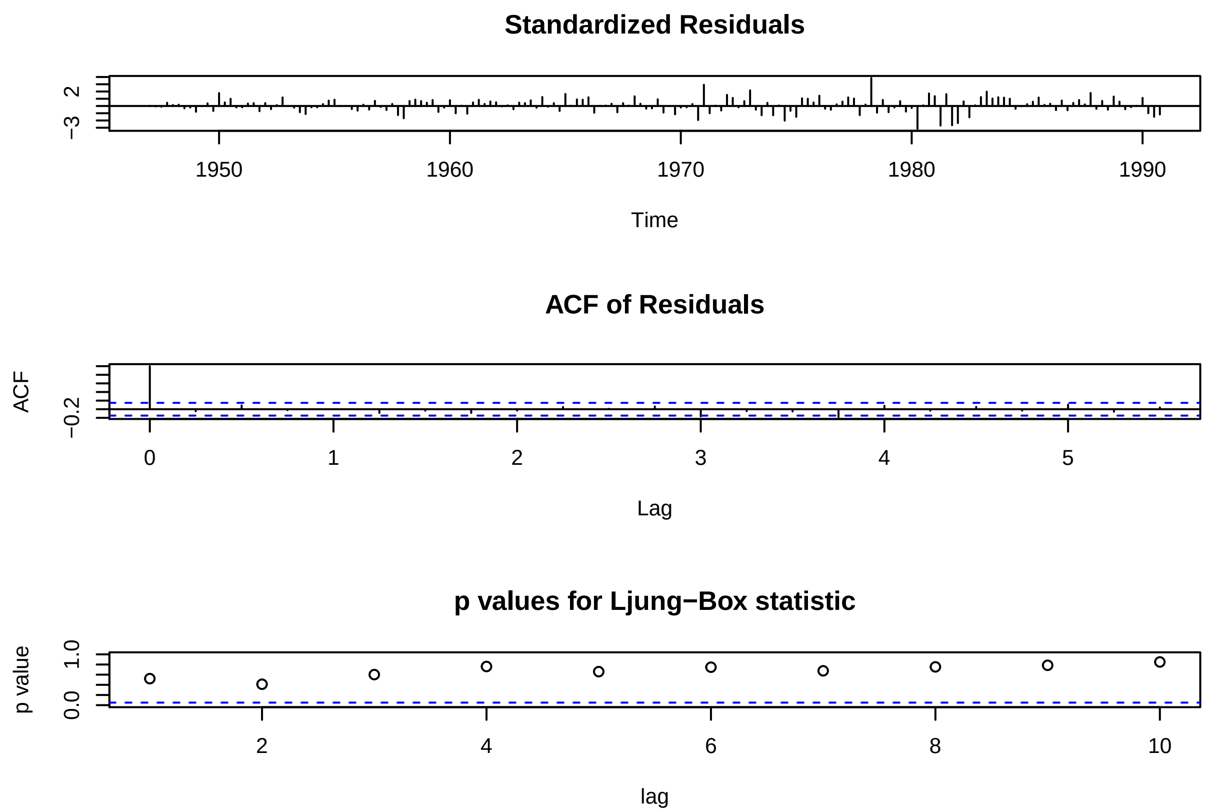
ARMA(1,2,1)

```
test(residuals(arima.fit))
```

```
## Null hypothesis: Residuals are iid noise.
## Test                          Distribution Statistic   p-value
## Ljung-Box Q                  Q ~ chisq(20)      23.4    0.2697
## McLeod-Li Q                  Q ~ chisq(20)     54.88         0 *
## Turning points T    (T-116)/5.6 ~ N(0,1)       125    0.1058
## Diff signs S       (S-87.5)/3.8 ~ N(0,1)        92    0.2413
## Rank P          (P-7700)/390.8 ~ N(0,1)      8008    0.4306
```




```
tsdiag(arima.fit)
```



Analizując reszty dla tego modelu, nie mamy podstaw do odrzucenia hipotezy zerowej, że reszty są szumem i.i.d, zatem model nasz wydaje się być senownym dopasowaniem.

Porównanie jakości dopasowania modeli w oparciu o kryteria informacyjne

```
AIC(ma21.fit)
```

```
## [1] 1790.639
```

```
AIC(ar20.fit)
```

```
## [1] 1782.12
```

```
AIC(arima.fit)
```

```
## [1] 1755.943
```

```
AIC(ma21.fit, k = log(length(training_set_stat)))
```

```
## [1] 1860.139
```

```
AIC(ar20.fit, k = log(length(training_set_stat)))
```

```
## [1] 1848.46
```

```
AIC(arima.fit, k = log(length(training_set_stat)))
```

```
## [1] 1771.738
```

Widzimy, że wśród wszystkich kryteriów model dopasowany automatycznie przez funkcję *auto.arima* ma najlepsze wyniki.

Ocena istotności współczynników

MA(21)

```
coeftest(ma21.fit)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1   0.412912   0.076316  5.4105 6.283e-08 ***
## ma2   0.358904   0.110657  3.2434  0.001181 **
## ma3   0.236163   0.089005  2.6534  0.007969 **
## ma4   0.174528   0.105764  1.6502  0.098909 .
## ma5   0.137494   0.099844  1.3771  0.168483
## ma6   0.197594   0.109913  1.7977  0.072219 .
## ma7   0.083705   0.096611  0.8664  0.386264
## ma8  -0.096193   0.103754 -0.9271  0.353862
## ma9   0.089847   0.101225  0.8876  0.374754
## ma10  0.248564   0.093884  2.6476  0.008107 **
## ma11  0.207346   0.093414  2.2197  0.026442 *
## ma12 -0.021998   0.089644 -0.2454  0.806156
## ma13  0.106205   0.106324  0.9989  0.317854
## ma14  0.080004   0.087093  0.9186  0.358303
## ma15 -0.122626   0.091451 -1.3409  0.179954
## ma16  0.151182   0.095623  1.5810  0.113873
## ma17  0.141965   0.085679  1.6569  0.097530 .
## ma18  0.044468   0.104323  0.4263  0.669921
## ma19  0.052982   0.107822  0.4914  0.623154
## ma20  0.277125   0.100258  2.7641  0.005708 **
## ma21  0.148790   0.078815  1.8878  0.059048 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Funkcja *coeftest* pokazuje nam, że dla modelu $MA(21)$ powinniśmy usunąć zmienne $1, 2, 3, 10, 11, 12$.

Sprawdźmy, jak wyzerowanie współczynników wpłynie na model.

```
ma.fixed <- numeric(21)
ma.fixed[1] <- NA
ma.fixed[2] <- NA
ma.fixed[3] <- NA
ma.fixed[10] <- NA
ma.fixed[11] <- NA
ma.fixed[20] <- NA

ma.zeros <- Arima(training_set, order=c(0,1,21), fixed=ma.fixed)
ma.zeros
```

```
## Series: training_set
## ARIMA(0,1,21)
##
```

```
## Coefficients:
##          ma1     ma2     ma3  ma4  ma5  ma6  ma7  ma8  ma9    ma10    ma11
##       0.4182  0.2843  0.1482    0    0    0    0    0    0  0.1771  0.1518
## s.e.  0.0806  0.0743  0.0744    0    0    0    0    0    0  0.0799  0.0716
##       ma12  ma13  ma14  ma15  ma16  ma17  ma18  ma19    ma20  ma21
##          0     0     0     0     0     0     0     0  0.1741     0
## s.e.     0     0     0     0     0     0     0     0  0.0746     0
##
## sigma^2 estimated as 1491:  log likelihood=-885.35
## AIC=1784.71   AICc=1785.38   BIC=1806.86
```

```
ma21.fit
```

```
## Series: training_set
## ARIMA(0,1,21)
##
## Coefficients:
##          ma1     ma2     ma3     ma4     ma5     ma6     ma7      ma8     ma9
##       0.4129  0.3589  0.2362  0.1745  0.1375  0.1976  0.0837  -0.0962  0.0898
## s.e.  0.0763  0.1107  0.0890  0.1058  0.0998  0.1099  0.0966   0.1038  0.1012
##         ma10    ma11     ma12    ma13    ma14     ma15    ma16    ma17    ma18
##       0.2486  0.2073  -0.0220  0.1062  0.0800  -0.1226  0.1512  0.1420  0.0445
## s.e.  0.0939  0.0934   0.0896  0.1063  0.0871   0.0915  0.0956  0.0857  0.1043
##         ma19    ma20    ma21
##       0.0530  0.2771  0.1488
## s.e.  0.1078  0.1003  0.0788
##
## sigma^2 estimated as 1378:  log likelihood=-873.32
## AIC=1790.64   AICc=1797.3   BIC=1860.26
```

Widzimy, że według kryterium AIC model po usunięciu zmiennych ma trochę lepsze własności, a więc dalej będziemy używać modelu zmodyfikowanego do prognoz.

subsection{AR(20)}

```
coeftest(ar20.fit)
```

```
##
## z test of coefficients:
##
##        Estimate Std. Error z value  Pr(>|z|)
## ar1   0.3913343  0.0752289  5.2019 1.972e-07 ***
## ar2   0.1252265  0.0814110  1.5382   0.12400
## ar3   0.0336752  0.0812134  0.4147   0.67840
## ar4   0.0408706  0.0815604  0.5011   0.61630
## ar5  -0.0365426  0.0799969 -0.4568   0.64781
## ar6   0.0513473  0.0798643  0.6429   0.52027
## ar7  -0.0682383  0.0794150 -0.8593   0.39020
## ar8  -0.0623600  0.0790509 -0.7889   0.43019
## ar9   0.1595073  0.0781678  2.0406   0.04129 *
## ar10  0.0585325  0.0789016  0.7418   0.45818
## ar11  0.0955476  0.0795091  1.2017   0.22947
## ar12 -0.1725610  0.0791996 -2.1788   0.02935 *
## ar13  0.0551535  0.0809056  0.6817   0.49543
## ar14  0.0086723  0.0807641  0.1074   0.91449
## ar15 -0.1069655  0.0802360 -1.3331   0.18249
## ar16  0.1892501  0.0803355  2.3557   0.01849 *
```

```
## ar17  0.0035914  0.0819753  0.0438   0.96506
## ar18  0.1004416  0.0819682  1.2254   0.22044
## ar19 -0.0357851  0.0819836 -0.4365   0.66248
## ar20  0.0798473  0.0762591  1.0471   0.29508
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Funkcja *coeftest* pokazuje nam, że dla modelu $AR(20)$ powinniśmy usunąć zmienne $1, 9, 12, 16$.

Sprawdźmy, jak wyzerowanie współczynników wpłynie na model.

```
ar.fixed <- numeric(20)
ar.fixed[1] <- NA
ar.fixed[9] <- NA
ar.fixed[12] <- NA
ar.fixed[16] <- NA

ar.zeros <- Arima(training_set, order=c(0,1,20), fixed=ar.fixed)
ar.zeros
```

```
## Series: training_set
## ARIMA(0,1,20)
##
## Coefficients:
##          ma1  ma2  ma3  ma4  ma5  ma6  ma7  ma8     ma9  ma10  ma11    ma12
##       0.3920    0    0    0    0    0    0    0  0.1433     0     0  0.0241
## s.e.  0.0611    0    0    0    0    0    0    0  0.0687     0     0  0.0728
##       ma13  ma14  ma15    ma16  ma17  ma18  ma19  ma20
##          0     0     0  0.1562     0     0     0     0
## s.e.     0     0     0  0.0664     0     0     0     0
##
## sigma^2 estimated as 1736:  log likelihood=-899.44
## AIC=1808.88   AICc=1809.23   BIC=1824.7
```

```
ar20.fit
```

```
## Series: training_set
## ARIMA(20,1,0)
##
## Coefficients:
##          ar1     ar2     ar3     ar4      ar5     ar6      ar7      ar8     ar9
##       0.3913  0.1252  0.0337  0.0409  -0.0365  0.0513  -0.0682  -0.0624  0.1595
## s.e.  0.0752  0.0814  0.0812  0.0816   0.0800  0.0799   0.0794   0.0791  0.0782
##         ar10    ar11     ar12    ar13    ar14     ar15    ar16    ar17    ar18
##       0.0585  0.0955  -0.1726  0.0552  0.0087  -0.1070  0.1893  0.0036  0.1004
## s.e.  0.0789  0.0795   0.0792  0.0809  0.0808   0.0802  0.0803  0.0820  0.0820
##          ar19    ar20
##       -0.0358  0.0798
## s.e.   0.0820  0.0763
##
## sigma^2 estimated as 1354:  log likelihood=-870.06
## AIC=1782.12   AICc=1788.16   BIC=1848.58
```

W tym przypadku zmodyfikowany model ma gorsze własności według kryteriów informacyjnych.

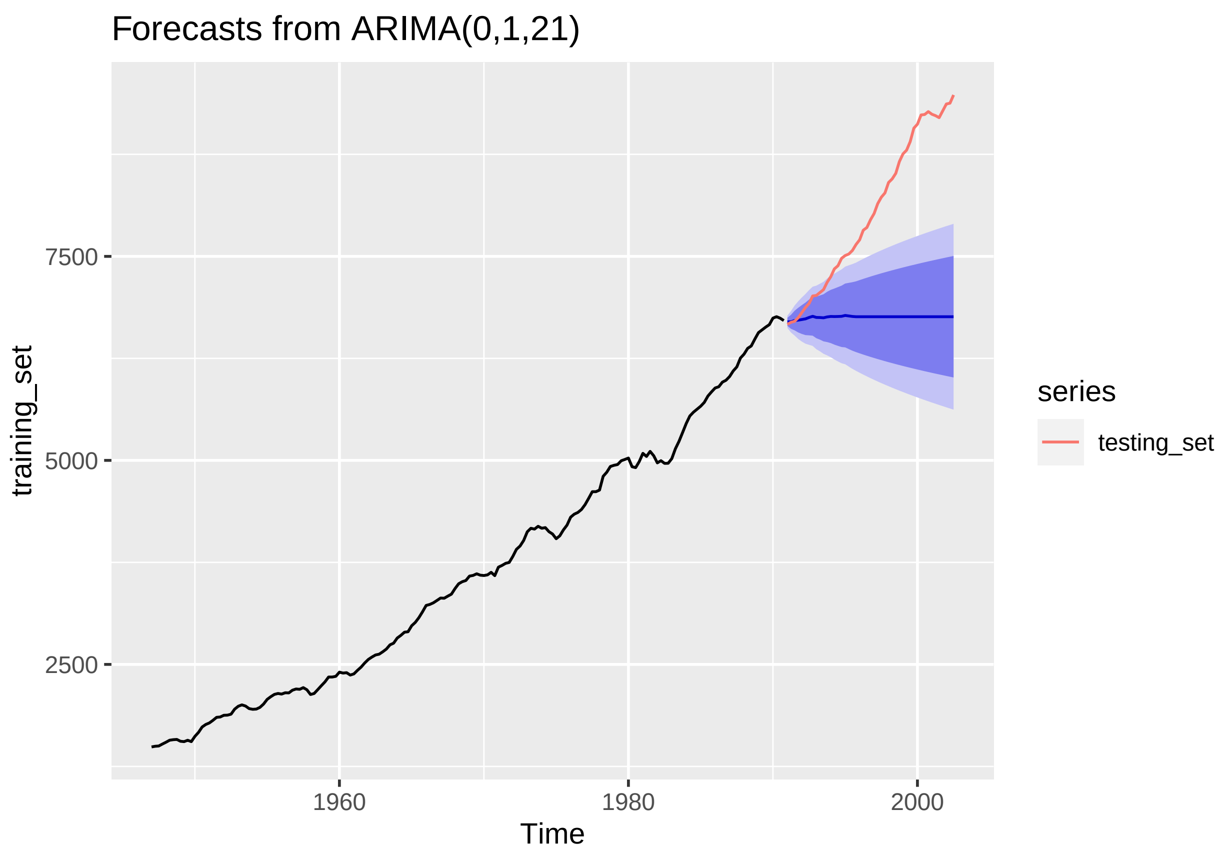
Wnioski na podstawie przeprowadzonej diagnostyki modeli ARIMA

Z trzech rozważanych modeli, zdecydowanie najlepszym kandydatem jest model $ARMA(1, 2, 1)$, którego wybór sugeruje zarówno analiza reszt, jak i kryteria AIC oraz BIC, zaraz potem $AR(20)$, a najmniej odpowiedni jest $MA(21)$.

Zastosowanie modeli do konstrukcji prognoz dla zbioru testowego

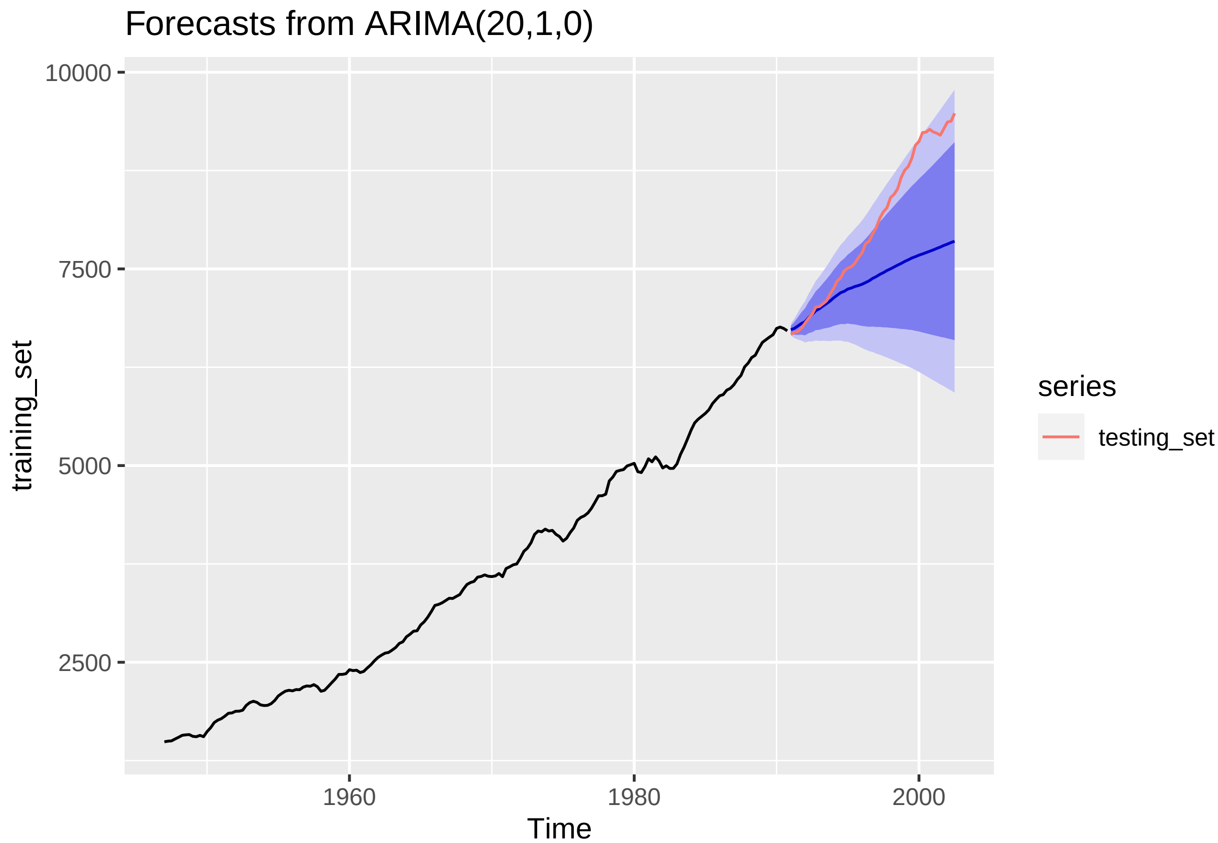
MA(21)

```
h <- length(testing_set)
ma21.forecast <- forecast::forecast(ma.zeros, h = h)
autoplot(ma21.forecast) + autolayer(testing_set)
```



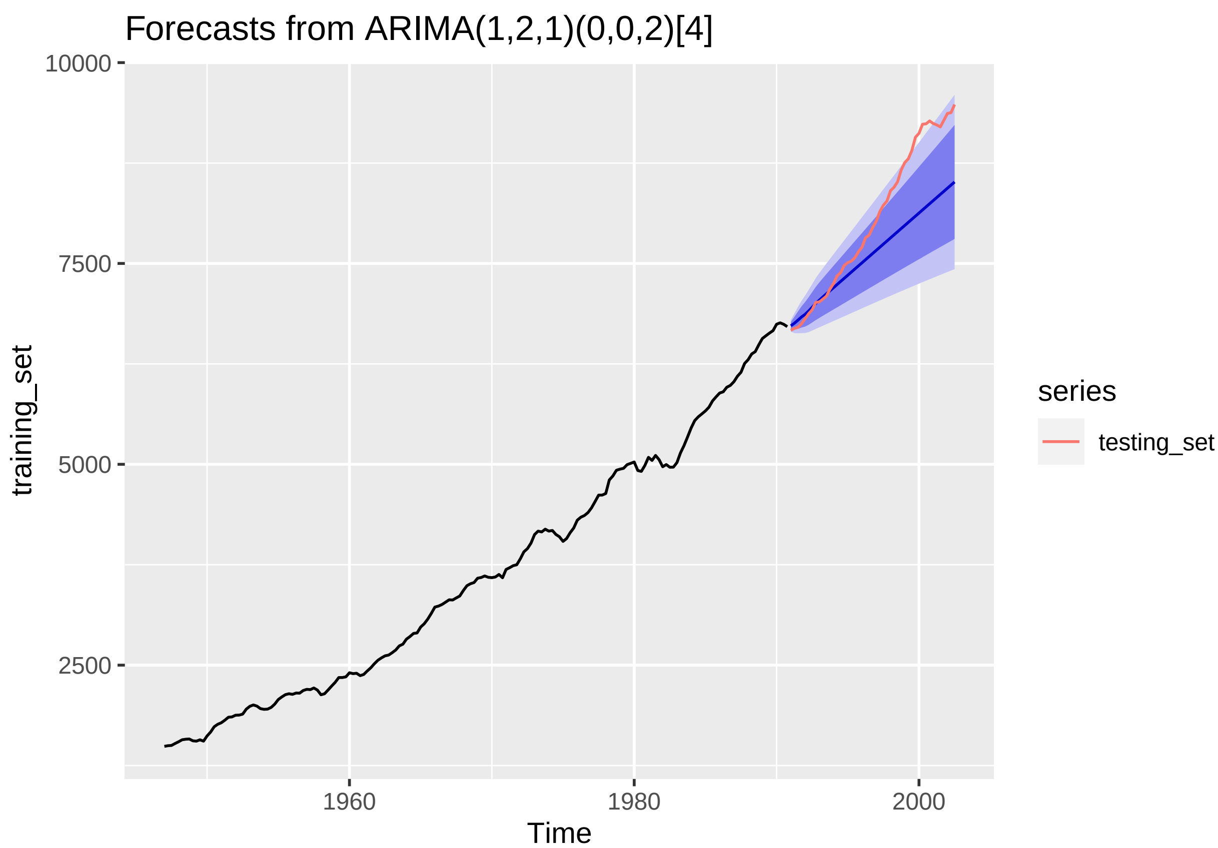
Widzimy, że model $MA(21)$, tak jak podejrzewaliśmy, nie ma zbyt dobrych własności predykcyjnych.

```
ar20.forecast <- forecast::forecast(ar20.fit, h = h)
autoplot(ar20.forecast) + autolayer(testing_set)
```



Model $AR(20)$ jest trochę lepszy, ale wyniki wciąż są daleko od zadowalających.

```
arima.forecast <- forecast::forecast(arima.fit, h = h)
autoplot(arima.forecast) + autolayer(testing_set)
```



Model dopasowany automatycznie ma najlepsze zdolności predykcyjne, ale wciąż fragmenty estymowanego szeregu czasowego wychodzą poza przedziały ufności.

Ocena dokładności

```
accuracy(ma21.forecast, testing_set)
```

```
##                       ME       RMSE        MAE        MPE       MAPE      MASE
## Training set   12.72106   37.83228   28.50615  0.3735088  0.8320866 0.2095082
## Test set     1290.02128 1579.36046 1291.98298 14.9387486 14.9681238 9.4955314
##                     ACF1 Theil's U
## Training set -0.08010559        NA
## Test set      0.94834698  19.57703
```

```
accuracy(ar20.forecast, testing_set)
```

```
##                      ME      RMSE      MAE       MPE      MAPE      MASE
## Training set   3.677295  34.53522  26.4001 0.1235633 0.7906076 0.1940296
## Test set     688.475517 911.87895 699.7302 7.8219564 7.9895618 5.1427223
##                    ACF1 Theil's U
## Training set -0.0104706        NA
## Test set      0.9527785  11.17093
```

```
accuracy(arima.forecast, testing_set)
```

```
##                      ME      RMSE       MAE       MPE     MAPE      MASE
## Training set   3.585384  36.04791  27.11557 0.1019004 0.796447 0.1992881
```

```
## Test set     422.892061 581.79252 443.81980 4.7621856 5.070063 3.2618889
##                     ACF1 Theil's U
## Training set -0.04788432        NA
## Test set      0.95804352  7.123378
```

Patrząc na współczynniki błędów, potwierdzają się słowa powyżej jeśli chodzi o poprawność prognozy. Najlepszym modelem jest model dopasowany automatycznie, później model autoregresji oparty na analizowaniu funkcji cząstkowej autokorelacji, a najgorzej wypada model ruchomej średniej wyznaczony poprzez analizę funkcji autokorelacji.