STA 9701 Project 1

Lily (Lizheng) Zhou

October 28, 2019

1 Introduction

Many of us may like to use `auto.arima()` function to fit best ARIMA model for univariate time series. This function returns a decent ARIMA model according to either AIC, AICc or BIC value. It is fast, easy and convenient to use, can provide us candidate models or give us suggestions of p and q values according to smallest AIC, AICc or BIC values. It saved much of our time.

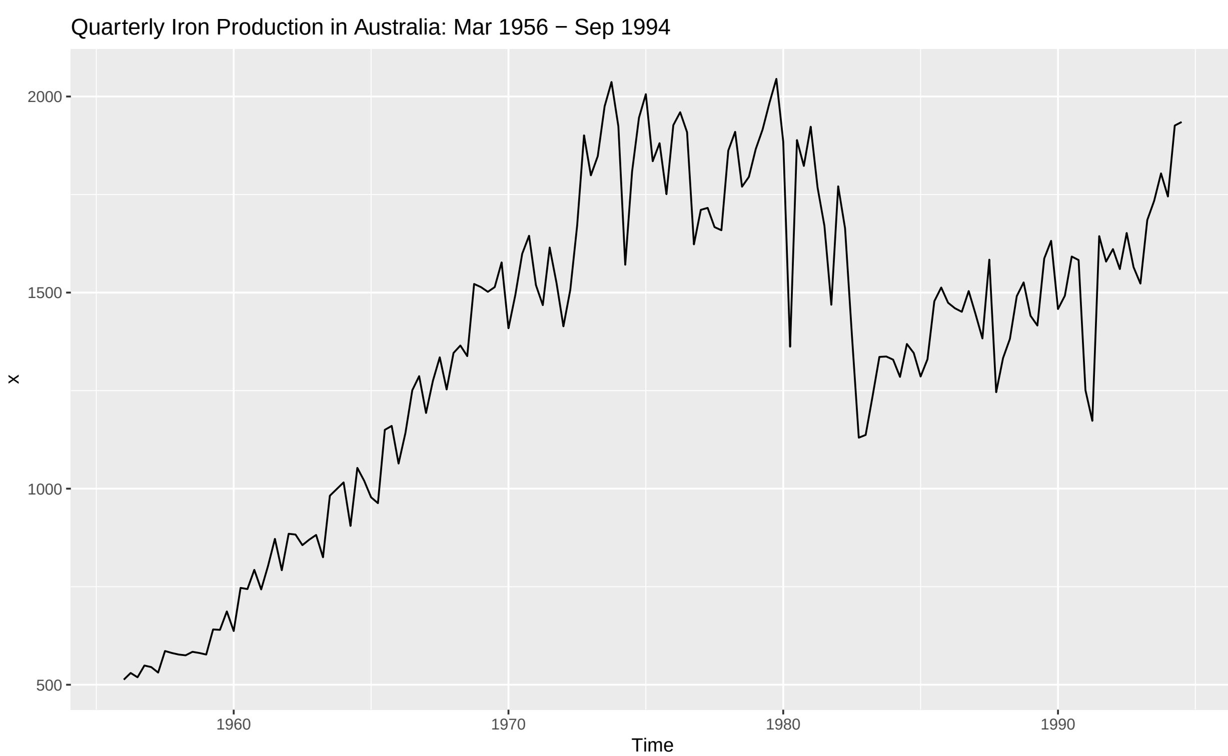
However, I wonder if this function always provides well models. Imagine such a situation, if an original dataset has both seasonal pattern and other patterns, while seasonal pattern is not that strong, I am curious that whether `auto.arima()` would choose to keep seasonal pattern or not.

To further investigate this question, I need a seasonal dataset but this seasonality should not be that strong.

There is a great time series dataset package called “Time Series Data Library (tsdl)” (https://pkg.yangzhuoranyang.com/tsdl/), which contains a list of 648 time series datasets. I used a `for` loop to auto print the datasets, dataset attributes and plot the datasets. Finally, I found that the 97th dataset in the package fulfills my need.

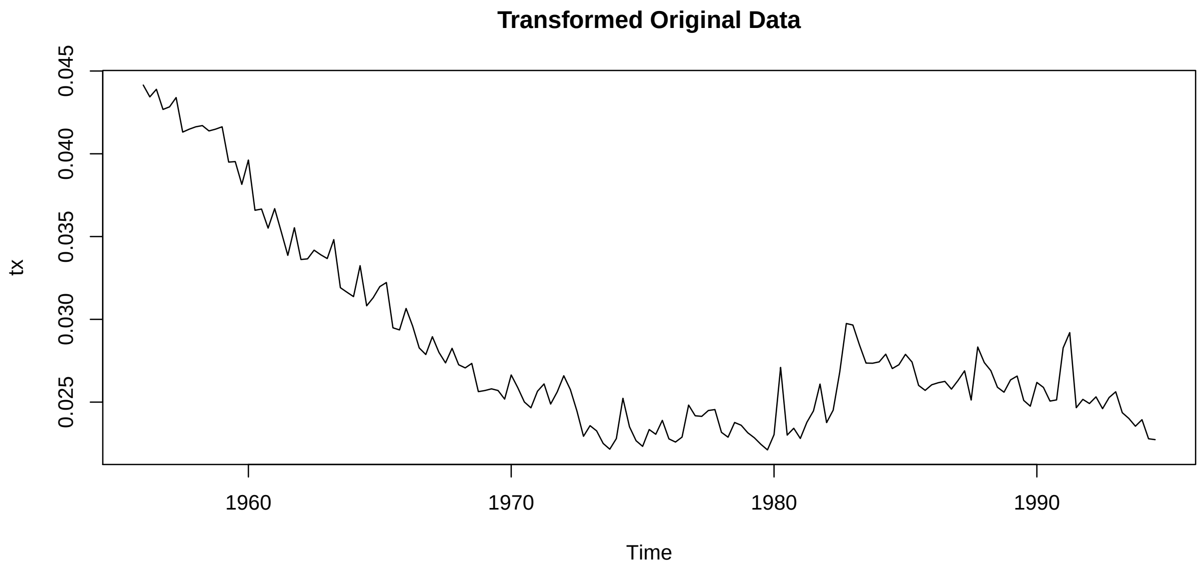
2 Exploratory Data Analysis

The 97th dataset of **tsdl** package is: Basic quarterly iron production in Australia: thousand tonnes from March 1956 to September 1994, giving a total of n = 155 observations. Source: Australian Bureau of Statistics (https://www.abs.gov.au/)

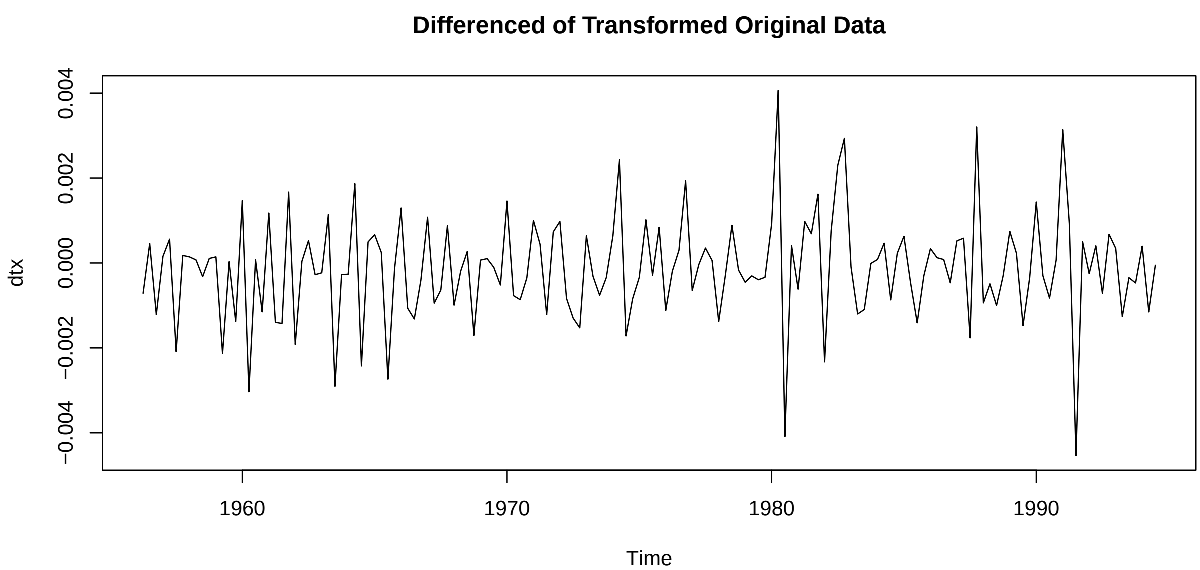


When look at the plot of original data plot, we see that there is a seasonal pattern, wile this seasonality is not that strong, this satisfies my need.

The time series plot indicates that there are some the variance differences, so original may not be stationary. To investigate the required transformation, we use BoxCox procedure. `BoxCox.lambda()` function gives us lambda = -0.3854732 which is close to -0.5. Thus, we take 1 over square root transformation.



Also took a first order diferencing according to the trend of original data.



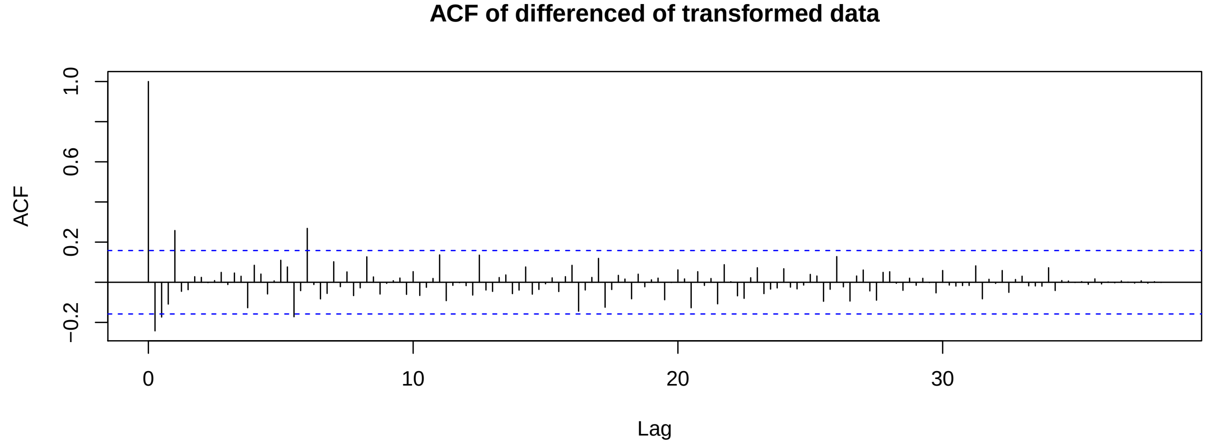
To test if the differenced of transformed original data is stationary, use `adf.test()` function.

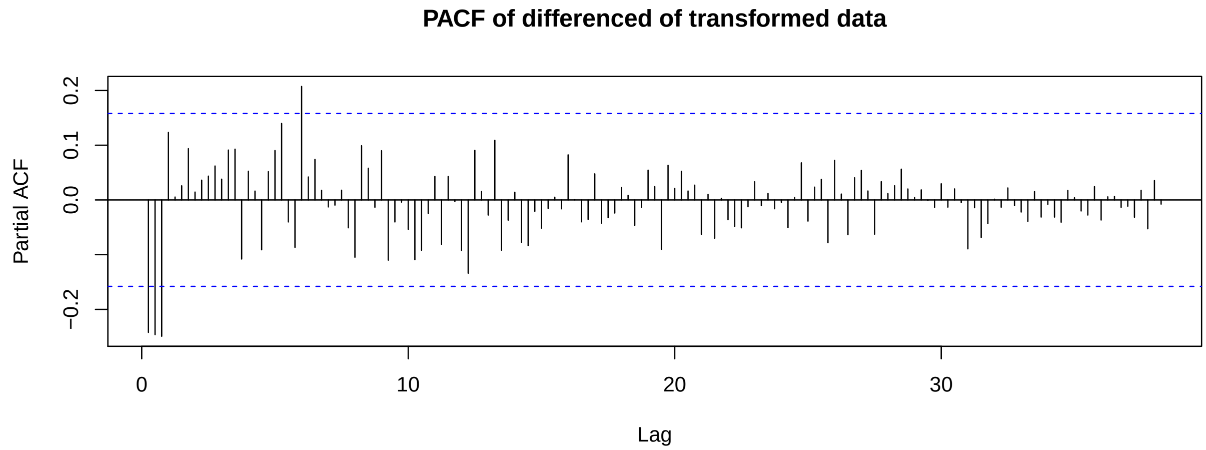
```
##
##  Augmented Dickey-Fuller Test
##
## data:  dtx
```

```
## Dickey-Fuller = -5.7733, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

From the result, we see it is stationary. Conbine with the plot shown above, I decide to use differenced of transformed original data (dtx) to do further analysis.

The sample ACF and sample PACF of differenced of the transformed data are shown below:





There is a seasonal pattern, while most are not significant, so I feel ACF and PACF all cut off actually. Thus, the ACF and PACF plots suggest a tentative model may be an AR(3), AR(4), MA(2), MA(4) of differenced of transformed data.

3 ARIMA Modeling

Fit 4 models: AR(3), AR(4), MA(2), MA(4) of differenced of transformed data, AIC and BIC shown below.

Model	AIC	BIC
AR(3)	-1645.103	-1629.918
AR(4)	-1645.439	-1627.217
MA(2)	-1639.416	-1627.268
MA(4)	-1645.617	-1627.395

we notice that AIC and BIC both are very close, respectively. I simply choose the smallest BIC here, so the model is AR(3) of differenced of transformed data.

At first, fit the AR(3) model of differenced of transformed data with constant, the estimated coeffcients and the corresponding p-values are given by `sarima()` function:

Parameter	Estimate	Standard Error	t-value	p-value
$\hat{\phi}_1$	-0.3619	0.0780	-4.6428	0.0000
$\hat{\phi}_2$	-0.3181	0.0788	-4.0367	0.0001
$\hat{\phi}_3$	-0.2470	0.0777	-3.1805	0.0018
$\hat{\mu}$	-0.0001	0.0001	-1.4597	0.1465

Since the constant term is not significant, re-fit the model without constant. The following Table shows the parameter estimates of the model of AR(3) of differenced of transformed data without constant.

Parameter	Estimate	Standard Error	t-value	p-value
$\hat{\phi}_1$	-0.3240	0.0787	-4.1148	0.0001
$\hat{\phi}_2$	-0.2789	0.0796	-3.5061	0.0006
$\hat{\phi}_3$	-0.2105	0.0785	-2.6796	0.0082

4 `auto.arima()` Modeling

Go back to the purpose of this project, I want to know if `auto.arima` function would always keep the seasonal pattern when seasonal pattern is not strong. Thus, use `auto.arima` function to get the model.

```
## Series: dtx
## ARIMA(0,0,2)(1,0,0)[4] with non-zero mean
##
## Coefficients:
##           ma1      ma2    sar1    mean
##       -0.3427  -0.1778  0.2621  -1e-04
## s.e.   0.0776   0.0759  0.0818   1e-04
##
## sigma^2 estimated as 1.267e-06:  log likelihood=828.83
## AIC=-1647.65   AICc=-1647.25   BIC=-1632.47
##
## Training set error measures:
##                         ME        RMSE          MAE      MPE     MAPE      MASE
## Training set -2.133764e-07 0.001111112 0.0007998581 116.8174 213.1336 0.7091858
##                     ACF1
## Training set 0.005004072
```

We see that although we already know that this dataset doesn't have strong seasonality pattern, but `auto.arima` function still chooses to provide us a seasonal model ARIMA(0,1,2)(1,0,0)[4].

I will compare this seasonal model **ARIMA(0,1,2)(1,0,0)[4]** with the model **ARIMA(3,1,0) without constant** we fit manually.

5 Model Comparison

5.1 AIC and BIC Comparison

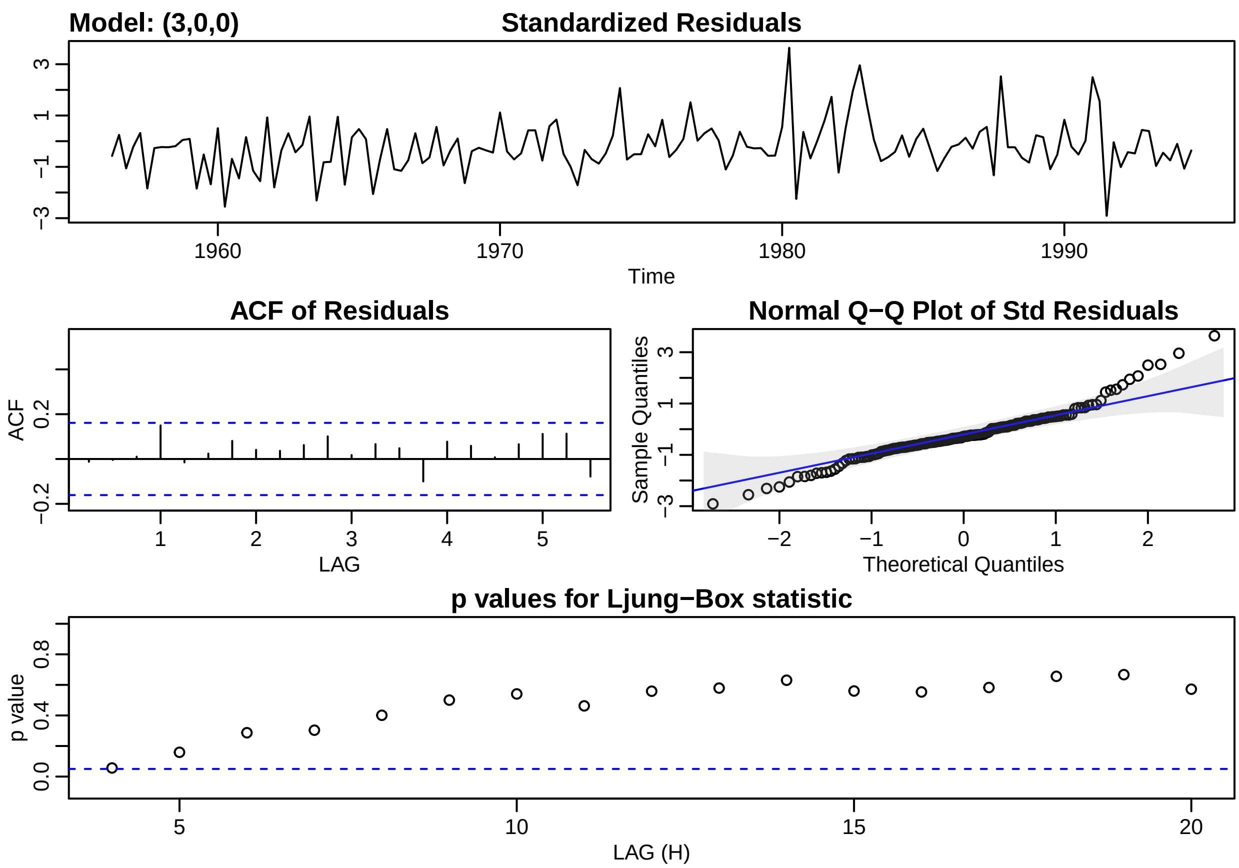
we first compare the AIC and BIC of two models.

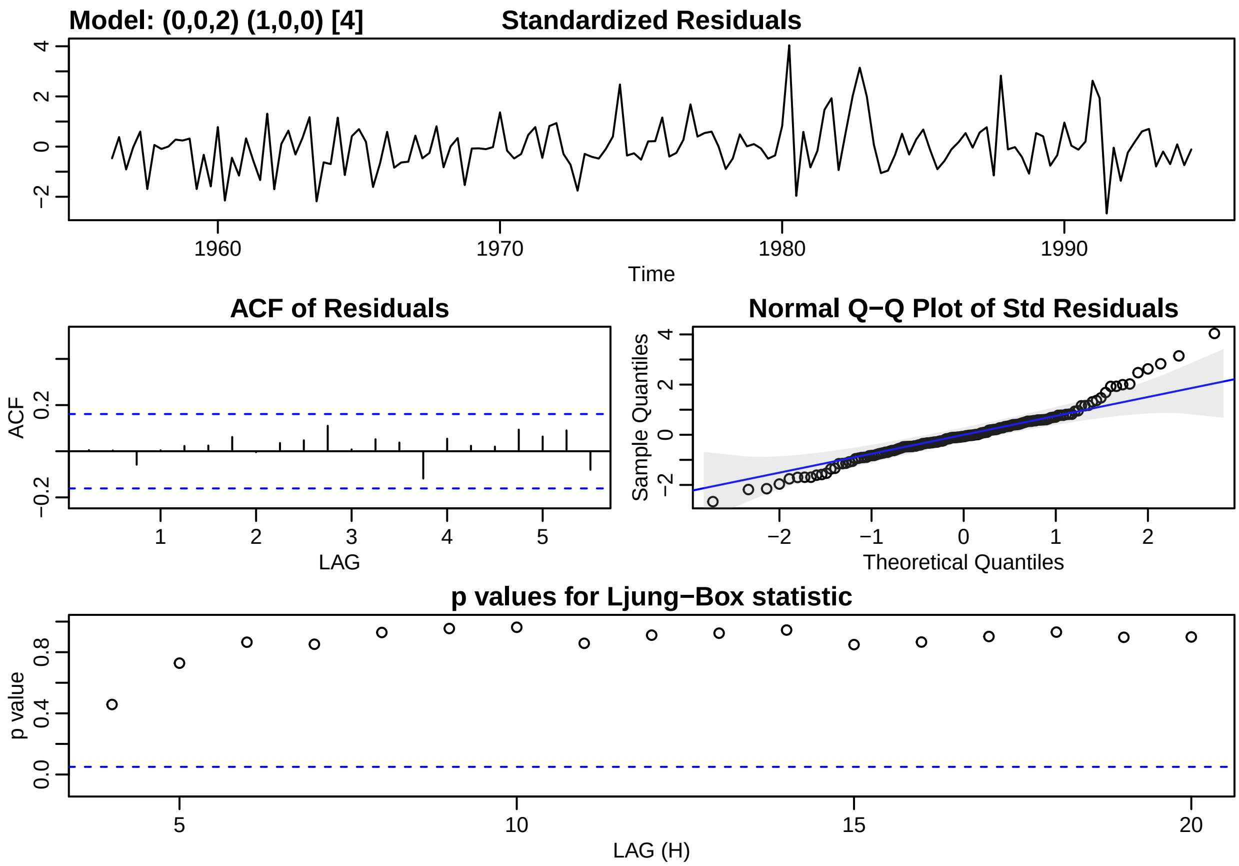
Model	AIC	BIC
ARIMA(3,1,0) without constant	-1639.42	-1627.272
ARIMA(0,1,2)(1,0,0)[4]	-1647.65	-1632.466

I feel that two models' AIC and BIC are still very close, respectively.

5.2 Diagnostic Checking and Comparing

Then, compare the diagnostic plots two models. Diagnostic plots shown below:





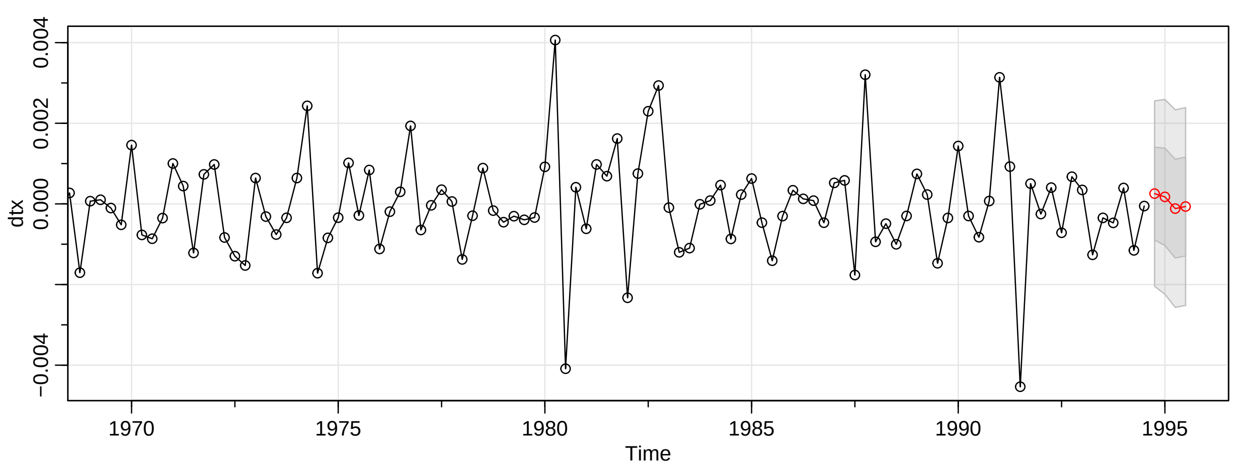
Look at the two diagnostic plots, we see that: **Standardized Residuals** totally same; **ACF of Residuals** are all not significant far away from 0; **Normal Q-Q Plot of Std Residuals** look similar, most parts are ok while all have outliers because of 1980s; **p values for Ljung-Box statistic** have no problems, all significant.

Thus, two diagnostic plots of two models also are very similar and satisfying.

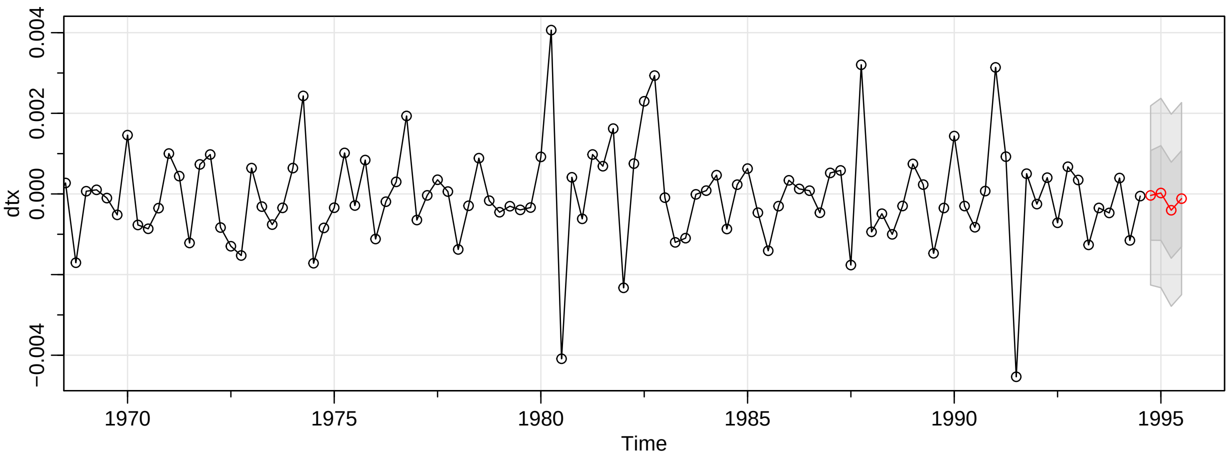
5.3 Forecasting Comparison

I use the two models to forecast the differenced of transformed data for next 4 quarters.

ARIMA(3,1,0) without constant forecasting plot:



ARIMA(0,1,2)(1,0,0)[4] forecasting plot:



Compare with forecasting of two models, I found that they are still very close, magnitudes are both very small.

I compared the AIC, BIC, diagnostic plots, forecasting of two models, I feel it is not necessary to keep the seasonal pattern for this dataset, the model `auto.arima()` function provides is kind of complex, AR model is enough.

6 Conclusion

From this proect, I used a seasonal dataset while seasonality is not strong to do a standard arima model building and get an AR model, later compare with the seasonal model got from `auto.arima()` function. By comparing with the AIC, BIC, diagnostic plots and forecasting of two models, I feel that we don't have to keep seasona pattern for this mode, thus I feel `auto.arima()` function might not always provide the most appropriate and simplest model.

7 Appendix

7.1 Dataset

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1956  513  530  519  549
## 1957  545  531  586  581
## 1958  577  575  584  581
## 1959  577  641  640  687
## 1960  637  747  744  793
## 1961  743  803  872  792
## 1962  885  883  856  870
## 1963  882  825  982  999
## 1964 1016  905 1053 1020
## 1965  978  963 1150 1160
## 1966 1064 1142 1251 1287
## 1967 1193 1275 1335 1253
## 1968 1346 1365 1338 1522
## 1969 1514 1502 1514 1577
## 1970 1409 1494 1599 1645
```

```
## 1971 1519 1468 1615 1524
## 1972 1414 1507 1671 1901
## 1973 1799 1848 1975 2037
## 1974 1924 1571 1809 1946
## 1975 2006 1835 1881 1751
## 1976 1927 1960 1909 1623
## 1977 1711 1716 1667 1659
## 1978 1862 1910 1770 1795
## 1979 1866 1916 1984 2045
## 1980 1885 1362 1889 1823
## 1981 1923 1768 1670 1469
## 1982 1771 1664 1391 1130
## 1983 1137 1235 1336 1337
## 1984 1329 1285 1369 1346
## 1985 1286 1330 1478 1513
## 1986 1474 1460 1451 1504
## 1987 1445 1383 1584 1246
## 1988 1333 1382 1491 1526
## 1989 1441 1416 1587 1632
## 1990 1458 1492 1592 1583
## 1991 1251 1173 1644 1579
## 1992 1611 1560 1652 1565
## 1993 1523 1685 1734 1804
## 1994 1745 1926 1935
```

7.2 Dataset Attribute

```
## $tsp
## [1] 1956.0 1994.5    4.0
##
## $class
## [1] "ts"
##
## $source
## [1] "Australian Bureau of Statistics"
##
## $description
## [1] "Basic quarterly iron production in Australia: thousand tonnes. Mar 1956 <U+FFFD> Sep 1994"
##
## $subject
## [1] "Production"
```

7.3 R Code

```
knitr::opts_chunk$set(echo = TRUE)
# Load library
library(forecast)
library(astsa)
library(tseries)
library(tsdl)
# Dataset plot
```

```r
par(mfrow=c(1,1))
x = tsdl[[97]]
autoplot(x,main='Quarterly Iron Production in Australia: Mar 1956 - Sep 1994')
# Transformation
BoxCox.lambda(x)
tx = 1/(sqrt(x))
plot(tx,main='Transformed Original Data')
# Differencing
dtx = diff(tx)
plot(dtx,main='Differenced of Transformed Original Data')
# adf test
adf.test(dtx)
# ACF and PACF of dtx
par(mfrow=c(2,1))
acf(dtx,155,main='ACF of differenced of transformed data')
pacf(dtx,155,main='PACF of differenced of transformed data')
# ARIMA test
ARIMA1 = arima(dtx, order=c(3,0,0))
ARIMA2 = arima(dtx, order=c(4,0,0))
ARIMA3 = arima(dtx, order=c(0,0,2))
ARIMA4 = arima(dtx, order=c(0,0,4))
# Check model AIC and BIC
AIC(ARIMA1);BIC(ARIMA1)
AIC(ARIMA2);BIC(ARIMA2)
AIC(ARIMA3);BIC(ARIMA3)
AIC(ARIMA4);BIC(ARIMA4)
# AR3 with constant
summary(ARIMA1)
sarima(dtx, 3,0,0)
# AR3 without constant
AR3_no_c = arima(dtx, order=c(3,0,0), include.mean=FALSE)
summary(AR3_no_c)
sarima(dtx, 3,0,0, no.constant=TRUE)
AIC(AR3_no_c);BIC(AR3_no_c)
# Use auto.arima function
ARIMAfit_dtx = auto.arima(dtx, approximation=FALSE,trace=FALSE)
summary(ARIMAfit_dtx)
# AR3 without constant
AIC(AR3_no_c);BIC(AR3_no_c)
# ARIMA(0,1,2)(1,0,0)[4]
AIC(ARIMAfit_dtx);BIC(ARIMAfit_dtx)
# Compare Diagnostic plots
sarima(dtx, 3,0,0, no.constant=TRUE)
sarima(dtx, 0,0,2, 1,0,0, 4)
# AR(3) forecasting
sarima.for(dtx, 4, 3,0,0, no.constant=TRUE)
# ARIMA(0,1,2)(1,0,0)[4] forecasting
sarima.for(dtx, 4, 0,0,2, 1,0,0, 4)
# Dataset
x
# Dataset Attribute
attributes(x)
```