USDA Dry Skim Milk Data Analysis

Ziyi Liu

2021/11/24

Contents

Summary

I will use the time series methods from PSTAT 174 course to analysis the real-world USDA Dry Skim Milk Data in this final project. The data set was collected from the USDA website, accessed in 2021/11/24. I use the data before 2018 to train the SARIMA model and use the data after 2018 to test the model fit, checking if the 95% confidence interval of the prediction contains the test set. First I do data visualization and determine suitable transformations to make the data meet the assumption of SARIMA model. Then I use sample ACF and PACF plot to choose candidate models. Select the best model by AICc score and check the model residual ACF, PACF, normaltiy, independence, characteristic roots, and conclude that it behaves quite like white noise. Then I make the forecast and find that the test sets are inside the 95% CI.

Introduction

The data was collected from the website: https://www.ers.usda.gov/data-products/dairy-data/. The Dairy Data set includes data covering domestic supply and demand and trade of various dairy products, and I use the skim milk production data. The data on the U.S. dairy situation and commercial disappearance is a monthly data, and it can help people have a better understanding of the dairy production and comsumption in the US. The main goal of this project is using the SARIMA model to predict the seasonality and future trends.

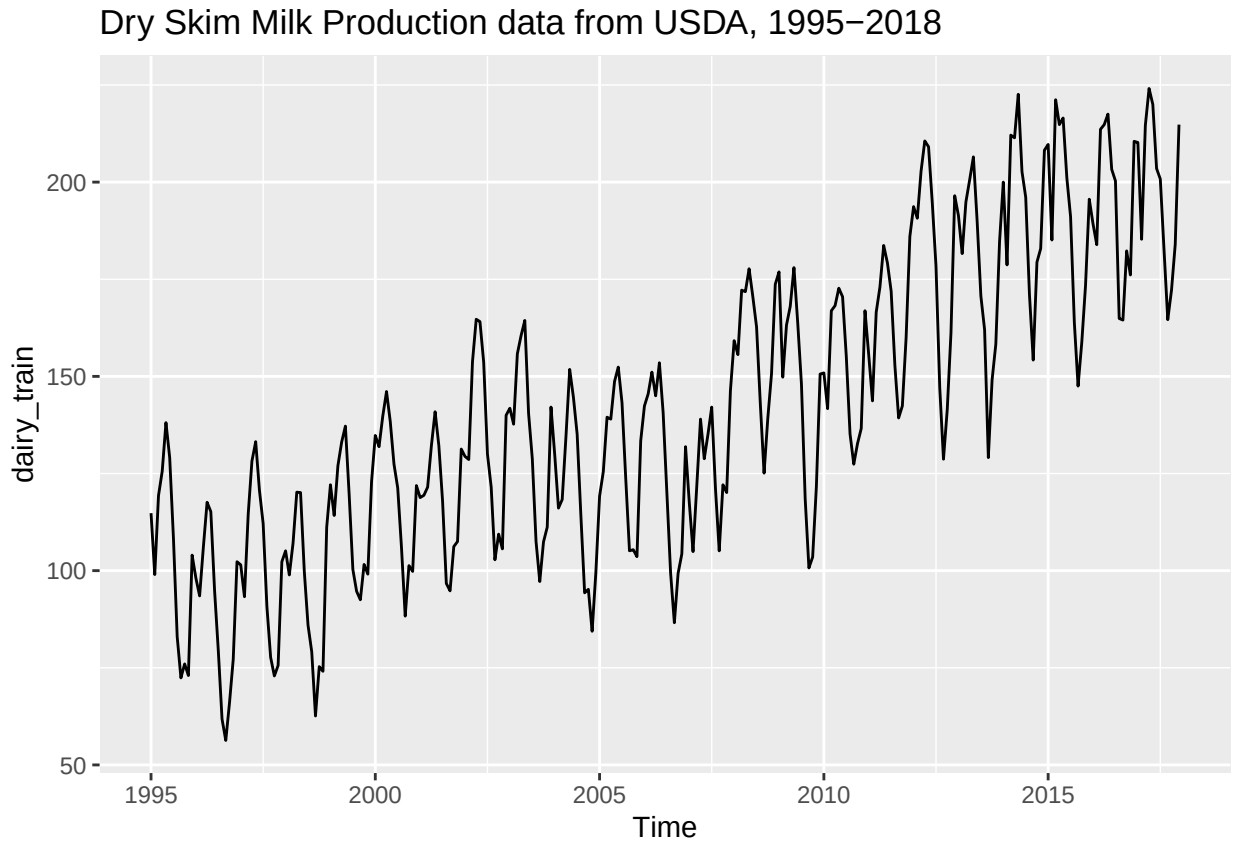
Data description

The data ranges from 1995 Jan to 2021 Sept. I split the data set into two parts: 1995-2017 Dec as the training set, and 2018 Jan - 2021 Sept as testing data set. The first 5 rows of the data is presented as:

```
##        Jan   Feb   Mar   Apr   May   Jun
## 1995 114.8  99.0 119.3 125.6 138.1 129.1
```

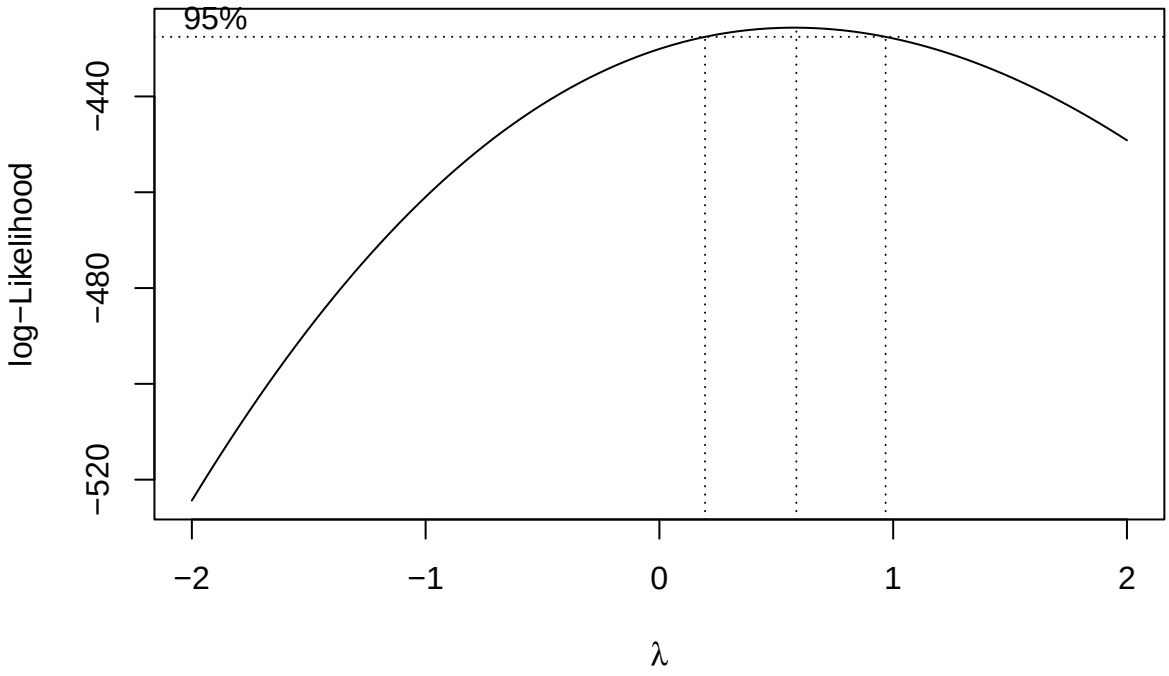
Exploratory analysis

The time series plot of the data:



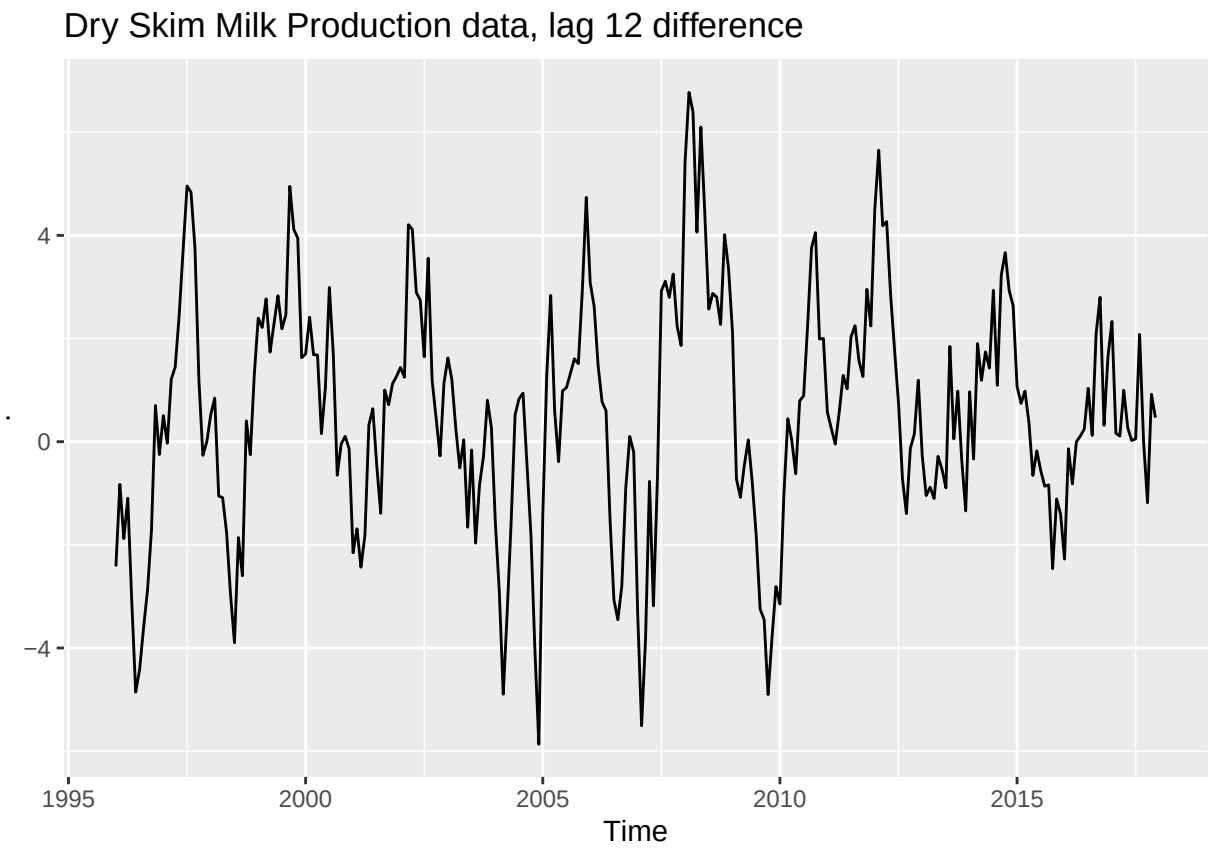
From the time series plot, the data has significant seasonal trend, not stationary. Also it may suffer from non-equal variance. So first we apply the Box-Cox transformation and check whether we need such a transformation to make the variance equal.

Transformation

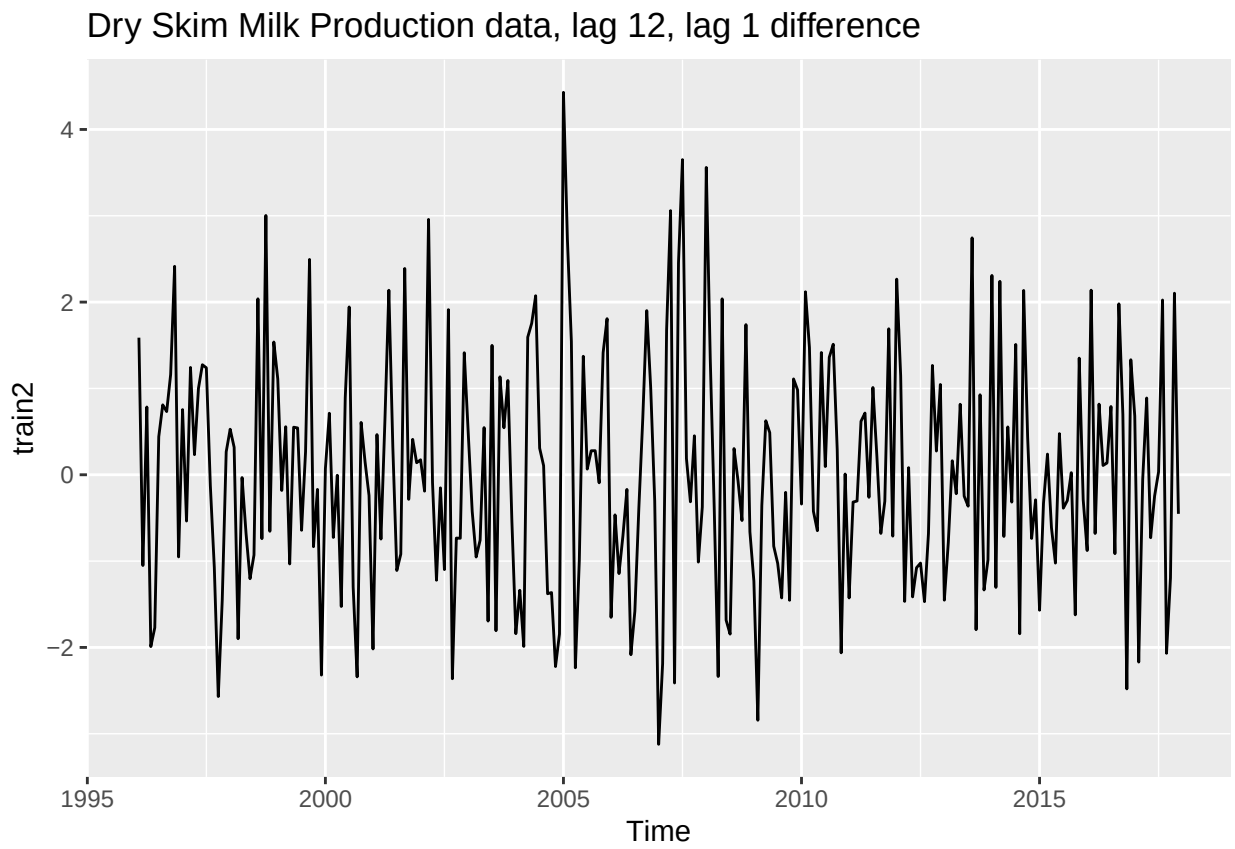


As the 95% confidence interval of λ does not contains 1 or 0, so we will use the optimal λ indicated by R to do the Box-Cox transformation.

The optimal lambda is 0.586, after applying the optimal lambda, then I apply a lag 12 difference to remove the seasonality and check whether the series is stationary.



From the plot above apparently it is not stationary, so we need additional lag 1 difference:

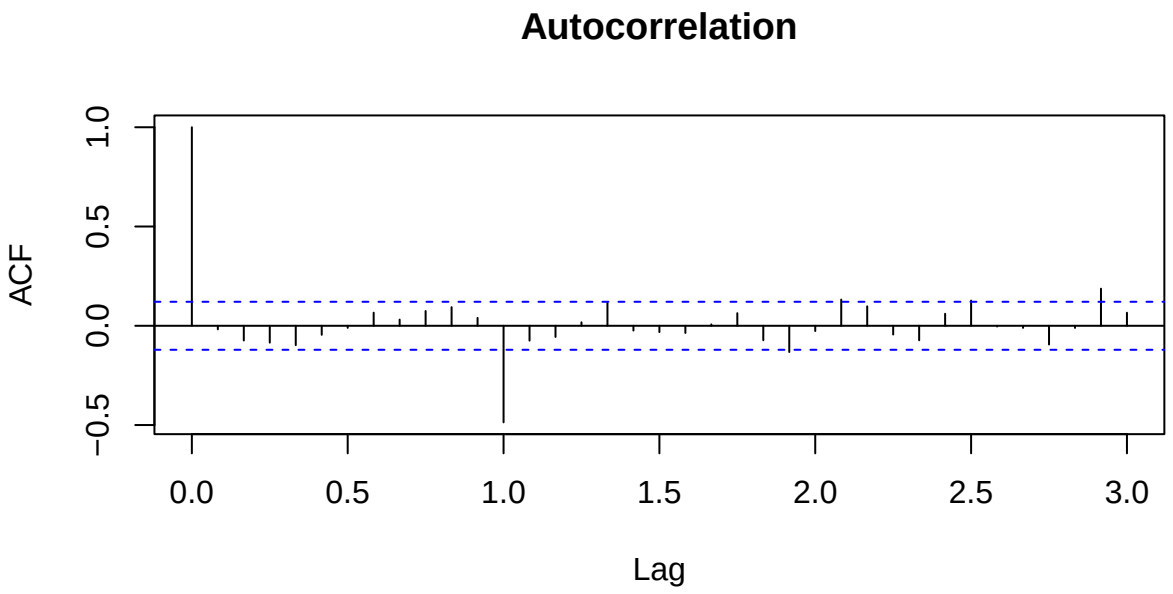


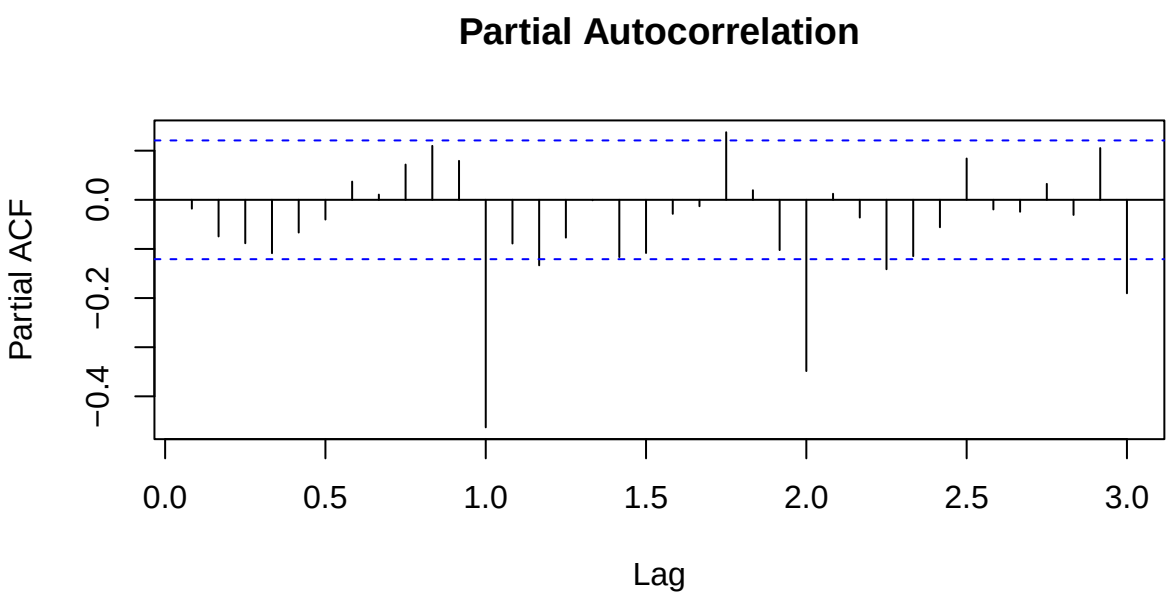
Now it seems quite stationary, but to make sure I apply the adf test to check stationary. The null hypothesis is that the series is not stationary, alternative is the series is stationary, the r result shows that p-value <= 0.01, thus we reject the null and conclude the series is stationary and ready to fit SARIMA model.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train2
## Dickey-Fuller = -6.8809, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Sample ACF and PACF plot

Now I can check the sample ACF and PACF plot to propose some candidate models.





The ACF plot has no significant spike from lag 1 to lag 11, and there is significant spike in lag 12, marginally significant spike at lag 23 and 35. The PACF also is not significant in the non seasonal part, and for the seasonal part it seems to be tailed off, so it suggest SMA 1 part. $p = 0, q = 0, P = 0, Q = 1$

Beside this most probably candidate model, I also fit all possible model with parameters p, q, P, Q range from 0 to 2, report their AICc. The model that cannot be fit in R will have AICc as NA. The final model will be the model with smallest AICc.

Model selection

Table 1: AICcs of the candidate models

	(0,1,0)	(1,1,0)	(2,1,0)	(0,1,1)	(1,1,1)	(2,1,1)	(0,1,2)	(1,1,2)	(2,1,2)
SARIMA(0,1,0)	907.3871	834.2302	801.6061	771.0320	772.5107	774.2180	772.4600	774.3265	775.7807
SARIMA(1,1,0)	909.3327	835.5837	801.5275	772.1711	773.5662	775.2915	773.5087	775.4437	777.0055
SARIMA(2,1,0)	909.9039	835.1587	800.3330	771.4991	772.7674	774.5322	772.7019	774.6748	776.0778
SARIMA(0,1,1)	909.3174	835.4056	800.8500	771.9159	773.2714	774.9987	773.2113	775.1615	776.4109
SARIMA(1,1,1)	904.3693	818.3679	786.0645	758.5123	760.0567	761.9091	760.0195	762.0288	763.1373
SARIMA(2,1,1)	891.2861	820.0750	NA	760.3989	762.0078	NA	761.9739	763.9979	768.3237
SARIMA(0,1,2)	909.2191	833.6394	797.5273	769.4759	770.4991	772.2299	770.4122	772.4046	773.7323
SARIMA(1,1,2)	891.3595	820.0647	788.1560	760.3894	761.9991	763.8780	761.9654	763.9839	765.1549
SARIMA(2,1,2)	886.6653	819.7275	789.6532	762.4755	764.0918	NA	764.0580	766.1084	766.2866

From this table the best model by AICc is $SARIMA(1,1,1)(0,1,1)_{12}$ with the smallest AICc 758.5123 (smaller than $SARIMA(0,1,0)(0,1,1)_{12} = 771.0320$). I think this is because the marginally significant spikes of lag 23, 35 make it suitable for AR1 and MA1 part.

So I get the model estimates as below:

```
## Series: train1
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.8288  -0.9867  -0.8433
## s.e.  0.0424   0.0246   0.0469
##
## sigma^2 estimated as 0.9569:  log likelihood=-375.18
## AIC=758.36   AICc=758.51   BIC=772.65
##
## Training set error measures:
##                       ME      RMSE      MAE       MPE     MAPE      MASE
## Training set 0.08392134 0.9494391 0.742133 0.2465636 2.652625 0.4103455
##                     ACF1
## Training set 0.01936663
```

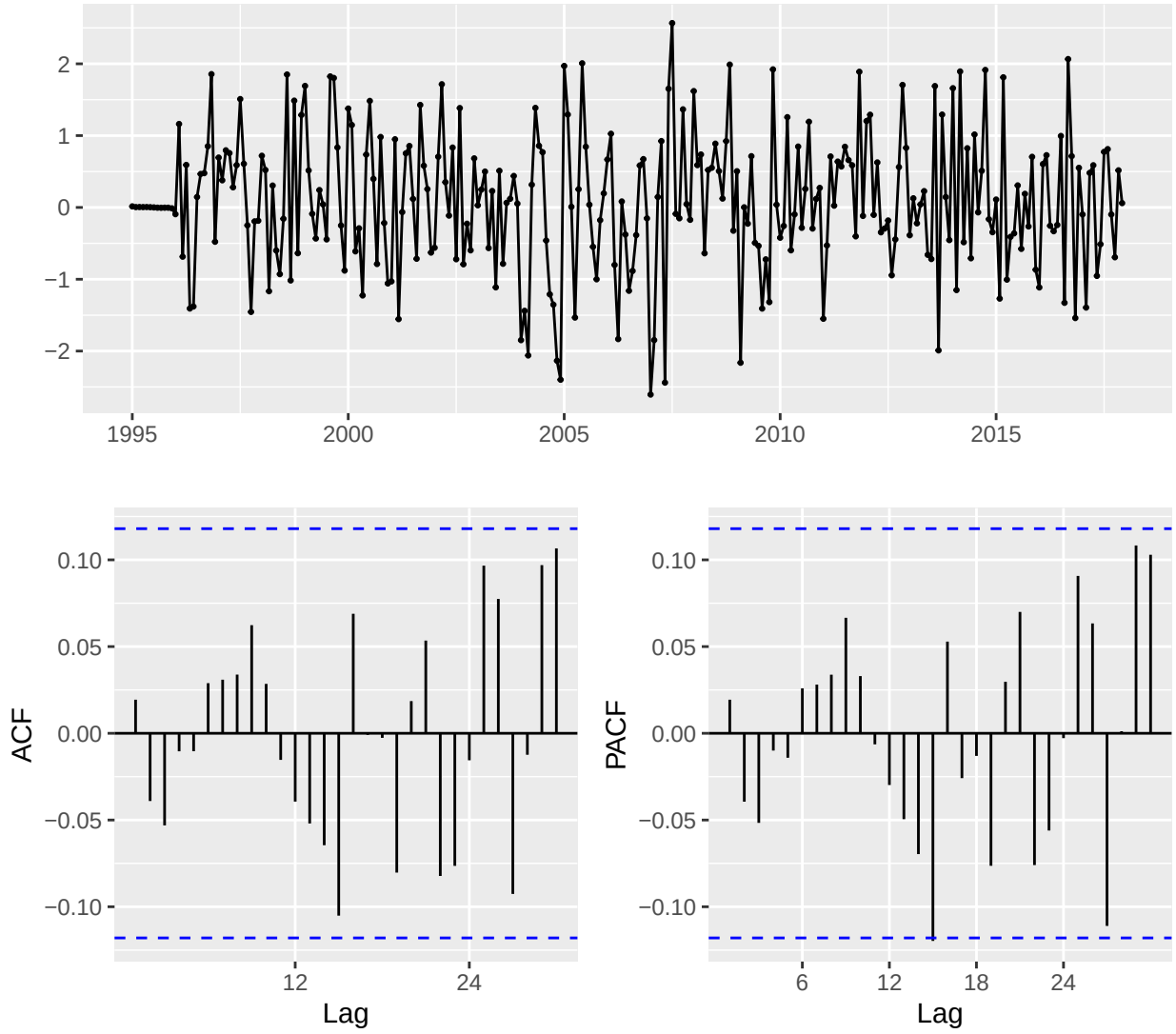
From the model summary, the ar1, ma1 and sma1 coefficient are all significant (absolute value bigger than 2 times standard error).

So the model is: (after Box cox transformation with $\lambda = 0.586$)

$$(1-B)(1-B^{12})(1-0.8288B)X_n = (1-0.9867B)(1-0.8433B^{12})Z_n,$$
$$Z_n \overset{iid}{\sim} N(0, 0.9569)$$

Model diagnostic

1. Check the residual time seris plot, ACF and PACF:



From the residual time series plot, it seems stationary and similar to white noise. All of the ACF and PACF lag are insignificant except 1 very marginal spike in PACF lag 15.

2. Check the residual independence:

Box - Pierce test: As the train data is length 276, the df should be 16 - 3 = 13.

Ljung - Box test: Same df as Box-Pierce, 13.

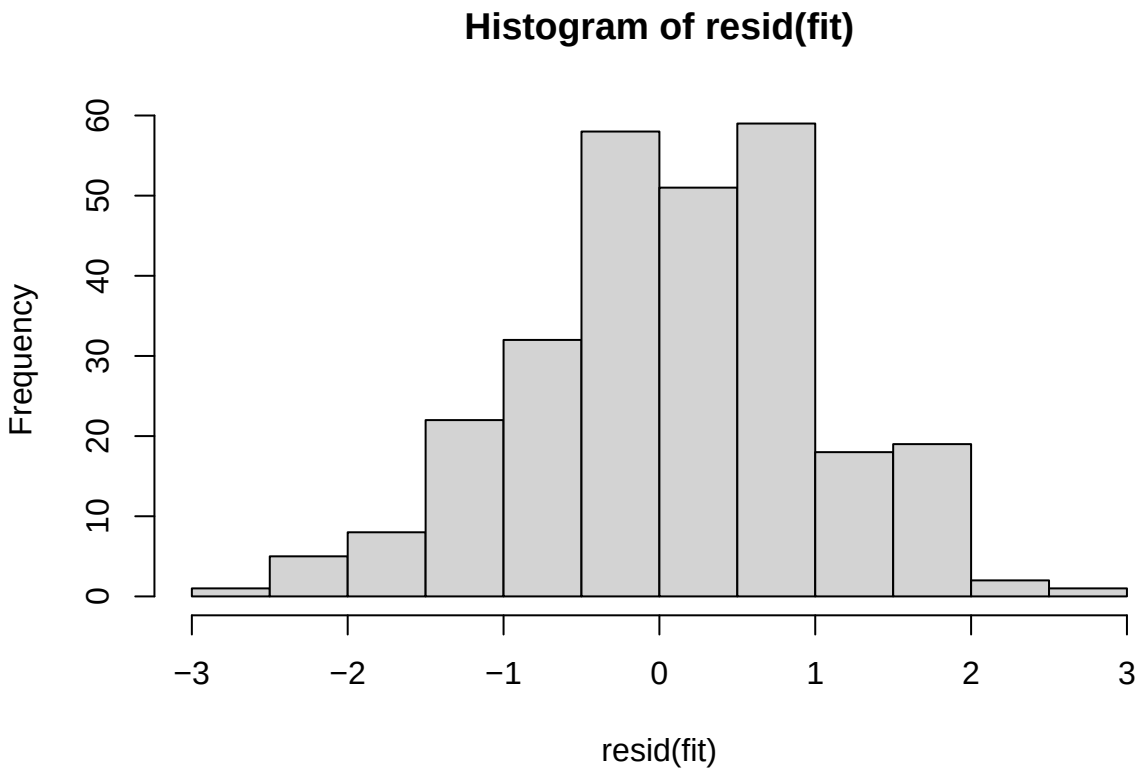
McLeod - Li test: Df is 16.

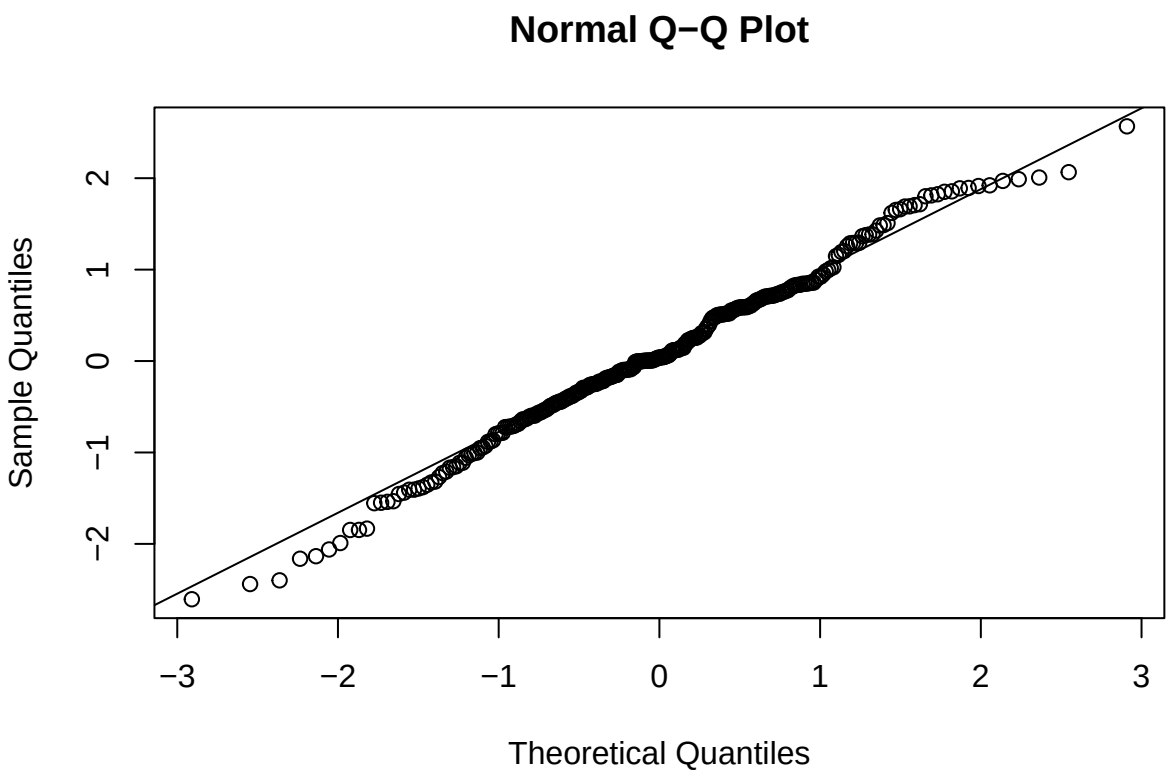
The result is as below:

Tests	p.value
Box-Pierce	0.6762769
Ljung-Box	0.6319650
McLeod-Li	0.0918462

As all of the p values are bigger than 0.05, all of the independence test are passed, so the model residual should be independent.

3. Check the residual normality using histogram, qqplot and Shapiro test.






```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit)
## W = 0.99272, p-value = 0.1964
```

The histogram and the qq plot seems very good. The null-hypothesis of this Shapiro test is that the population is normally distributed, and p value $= 0.19 > 0.05$ indicates there is no evidence that the residual is not normal.

4. Check the characteristic root.

The AR roots is solved from the equation:

$$(1 - 0.8288B) = 0$$

```
## [1] 1.206564+0i
```

The AR root is $1.207 > 1$, outside the unit circle, thus the model is stationary.

The MA roots is solved from the equation:

$$(1 - 0.9867B)(1 - 0.8433B^{12}) = 0$$

```
## [1] 1.013479+0i
```

```
## [1] 1.185818+0i
```

The MA root is 1.013479 > 1, the SMA root is 1.185818 > 1, all outside the unit circle, thus the model is invertible.

Now all of the tests are passed, I can conclude that the model is good and proceed to forecast. The model summary is here:

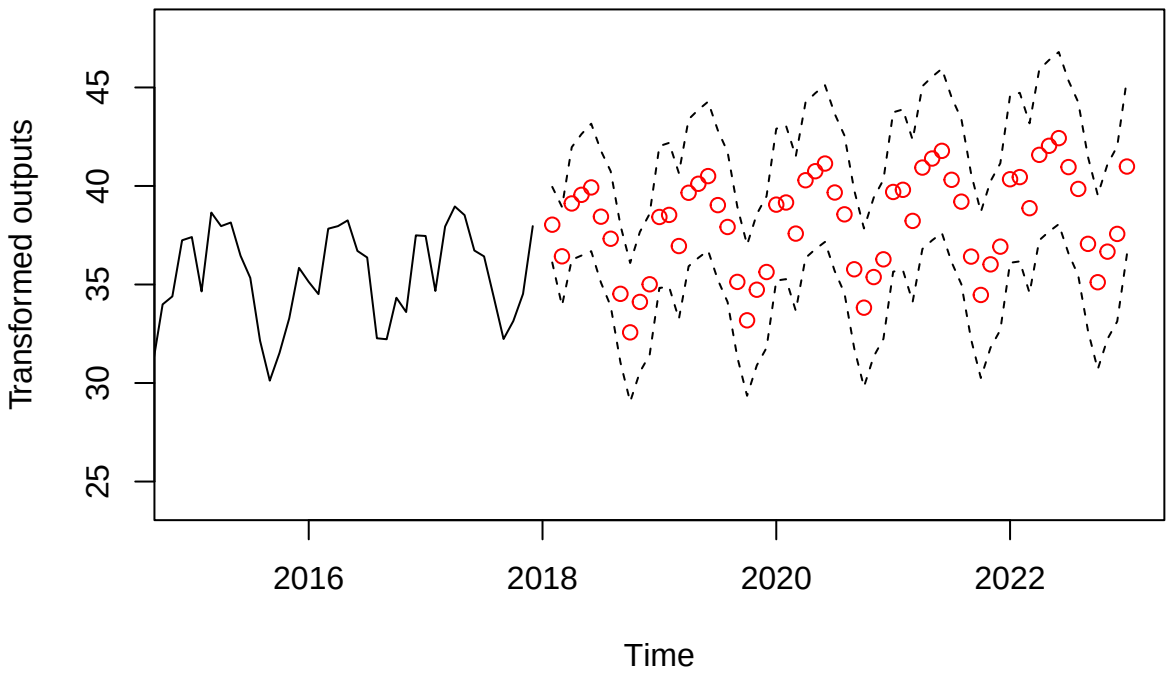
```
## Series: train1
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.8288  -0.9867  -0.8433
## s.e.  0.0424   0.0246   0.0469
##
## sigma^2 estimated as 0.9569:  log likelihood=-375.18
## AIC=758.36   AICc=758.51   BIC=772.65
##
## Training set error measures:
##                       ME      RMSE      MAE       MPE     MAPE      MASE
## Training set 0.08392134 0.9494391 0.742133 0.2465636 2.652625 0.4103455
##                     ACF1
## Training set 0.01936663
```

So the model is: (after Box cox transformation with $\lambda = 0.586$)

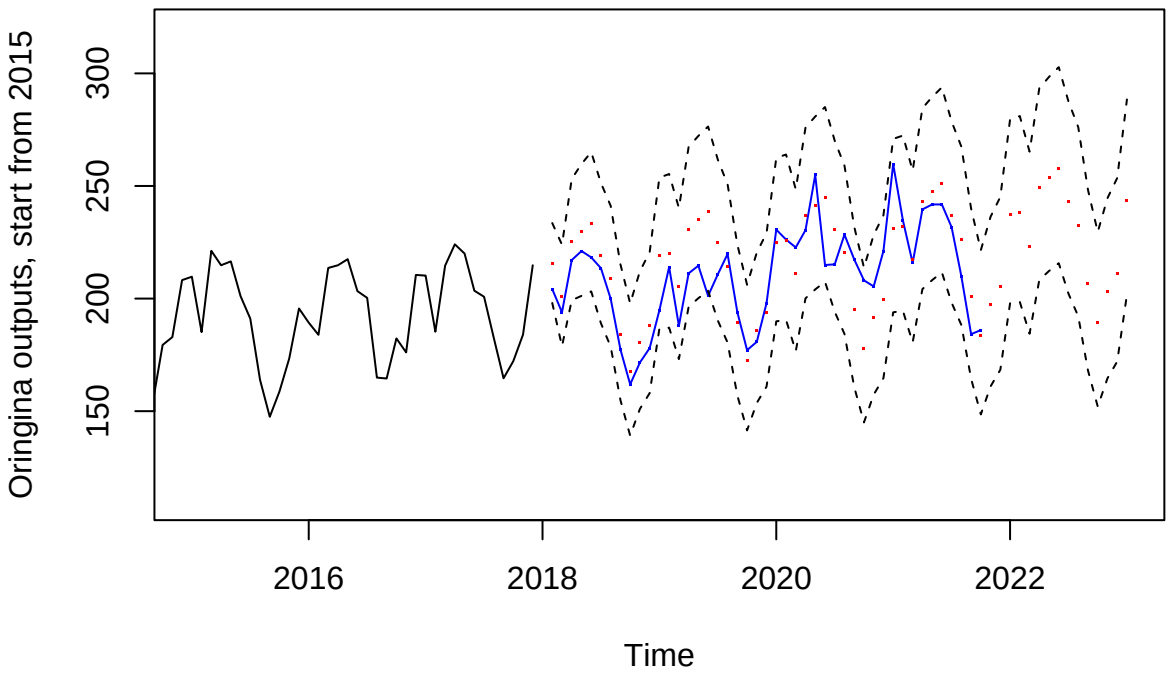
$$(1-B)(1-B^{12})(1-0.8288B)X_n = (1-0.9867B)(1-0.8433B^{12})Z_n,$$
$$Z_n \overset{iid}{\sim} N(0, 0.9569)$$

Forecasting

I could use the predict function to obtain the forecast. After Box-Cox transformation, we can compare the predicted value and the test set. Here is the forecast before the Box-Cox inverse transformation:



Then I transform it back and compare with the test data:



The forecast result shows that all of the test data (blue line) are inside the 95% confidence interval for the forecast value. So the forecast is very success.

Conclusion

In this project, I first apply Box-Cox transformation and then apply lag 1 and lag 12 difference, then choose some candidate model from the sample ACF and PACF, and calculate their AICc and choose the smallest AICc model: (after Box cox transformation with $\lambda = 0.586$)

$$(1-B)(1-B^{12})(1-0.8288B)X_n = (1-0.9867B)(1-0.8433B^{12})Z_n,$$
$$Z_n \overset{iid}{\sim} N(0, 0.9569)$$

Then I check the residual ACF, PACF, normality, independence and the residual pass all of the tests. The model characteristic roots shows that the model is invertible and stationary, so I can proceed to do the forecast.

In the forecast, the 95% CI for the prediction contains all of the test dataset from 2018 to 2021.

References

1. Dairy Data from USDA, accessed 2012/11/24. https://www.ers.usda.gov/data-products/dairy-data/
2. PSTAT 174 2021 Lecture slides, Labs
3. Special helps from Prof Raya Feldman and Jasmine Li

Appendix: Rcode

```
knitr::opts_chunk$set(echo = FALSE,message = FALSE,warning = FALSE
)
library(forecast)
library(MASS)
library(car)
library(tseries)
library(ggplot2)
library(dplyr)
data = read.csv("DrySkimmilkproduction.csv",header  =T)
data = ts(data, frequency = 12,start = c(1995,1))
dairy_train = ts(data[1:(12*23)],frequency = 12,start = c(1995,1))
dairy_test = data[-(1:(12*23))]
head(data)
autoplot(dairy_train) + ggtitle("Dry Skim Milk Production data from USDA, 1995-2018")
lambda = boxcox(dairy_train~1)
optlambda = lambda$x[which.max(lambda$y)]
train1 = (dairy_train^optlambda - 1)/optlambda
train1  %>% diff(lag=12) %>%
  autoplot() + ggtitle("Dry Skim Milk Production data, lag 12 difference")
train2 = train1 %>% diff(lag=12) %>% diff(lag=1)
autoplot(train2)+ ggtitle("Dry Skim Milk Production data, lag 12, lag 1 difference")
adf.test(train2)
# sample ACF PACF plots
par(mfrow = c(2,1))
acf(train2,main = "Autocorrelation",lag.max = 36)
pacf(train2,main = "Partial Autocorrelation",lag.max = 36)
# model selection via AICc
AICc = 9999
arimalist = list(c(0,1,0),c(1,1,0),c(2,1,0),c(0,1,1),c(1,1,1),c(2,1,1),c(0,1,2),c(1,1,2),c(2,1,2))
seasonallist = list(c(0,1,0),c(1,1,0),c(2,1,0),c(0,1,1),c(1,1,1),c(2,1,1),c(0,1,2),c(1,1,2),c(2,1,2))
AICcs = matrix(NA,length(arimalist),length(seasonallist))
for(i in 1:length(arimalist)){
  for(j in 1:length(seasonallist)){
    if((i == 6 & j == 3) | (i == 6 & j == 6) |
       (i == 9 & j == 6)){
      fit = list()
      fit$aicc = NA
    }
    else{
      fit = Arima(train1,order=arimalist[[i]],
                  seasonal = seasonallist[[j]],method = "CSS-ML")


    AICcs[i,j] = fit$aicc
    if(fit$aicc < AICc){
      AICc = fit$aicc
    }
    }
  }
}
rownames(AICcs) = c("SARIMA(0,1,0)","SARIMA(1,1,0)","SARIMA(2,1,0)",
```

```
 "SARIMA(0,1,1)","SARIMA(1,1,1)","SARIMA(2,1,1)",
 "SARIMA(0,1,2)","SARIMA(1,1,2)","SARIMA(2,1,2)")
colnames(AICcs) = c("(0,1,0)","(1,1,0)","(2,1,0)",
                    "(0,1,1)","(1,1,1)","(2,1,1)",
                    "(0,1,2)","(1,1,2)","(2,1,2)")
knitr::kable(AICcs, caption = "AICcs of the candidate models")
fit = Arima(train1,order=c(1,1,1),
                  seasonal = list(order=c(0,1,1),period=12),method = "CSS-ML")
summary(fit)
ggtsdisplay(resid(fit),lag.max = 30)
# independence tests
l = c()
l[1]=Box.test(resid(fit), lag = 16, type = c("Box-Pierce"),fitdf = 3)$p.value
l[2]=Box.test(resid(fit), lag = 16, type = c("Ljung-Box"),fitdf = 3)$p.value
l[3]=Box.test(resid(fit)^2, lag = 16, type = c("Ljung-Box"),fitdf = 0)$p.value
df = data.frame(Tests = c("Box-Pierce","Ljung-Box","McLeod-Li"), p.value = l)
knitr::kable(df)
# check normality
hist(resid(fit))
qqnorm(resid(fit))
qqline(resid(fit))
shapiro.test(resid(fit))
polyroot(c(1,-0.8288))
polyroot(c(1,-0.9867))
polyroot(c(1,-0.8433))
summary(fit)
# Forecast
pred = predict(fit, n.ahead=60)$pred
se = predict(fit, n.ahead=60)$se
ts.plot(train1, xlim = c(2015,2023),ylim = c(24,48), ylab = "Transformed outputs")
points(2018+(1:60)/12,pred,col = "red")
lines(2018+(1:60)/12,pred+1.96*se,lty=2)
lines(2018+(1:60)/12,pred-1.96*se,lty=2)
invbox = function(d,optlambda){
  return(exp(log(d * optlambda + 1)/optlambda))
}
predorigin = invbox(pred,optlambda)
predoriginlower = invbox(pred-1.96*se,optlambda)
predoriginupper = invbox(pred+1.96*se,optlambda)
ts.plot(dairy_train, xlim = c(2015,2023),ylim = c(110,320), ylab = "Oringina outputs, start from 2015")
points(2018+(1:45)/12,dairy_test,col = "blue",pch=".")
lines(2018+(1:45)/12,dairy_test,col = "blue")
points(2018+(1:60)/12,predorigin,col = "red",pch=".")
lines(2018+(1:60)/12,predoriginlower,lty=2)
lines(2018+(1:60)/12,predoriginupper,lty=2)
```