STA457A1 Practice

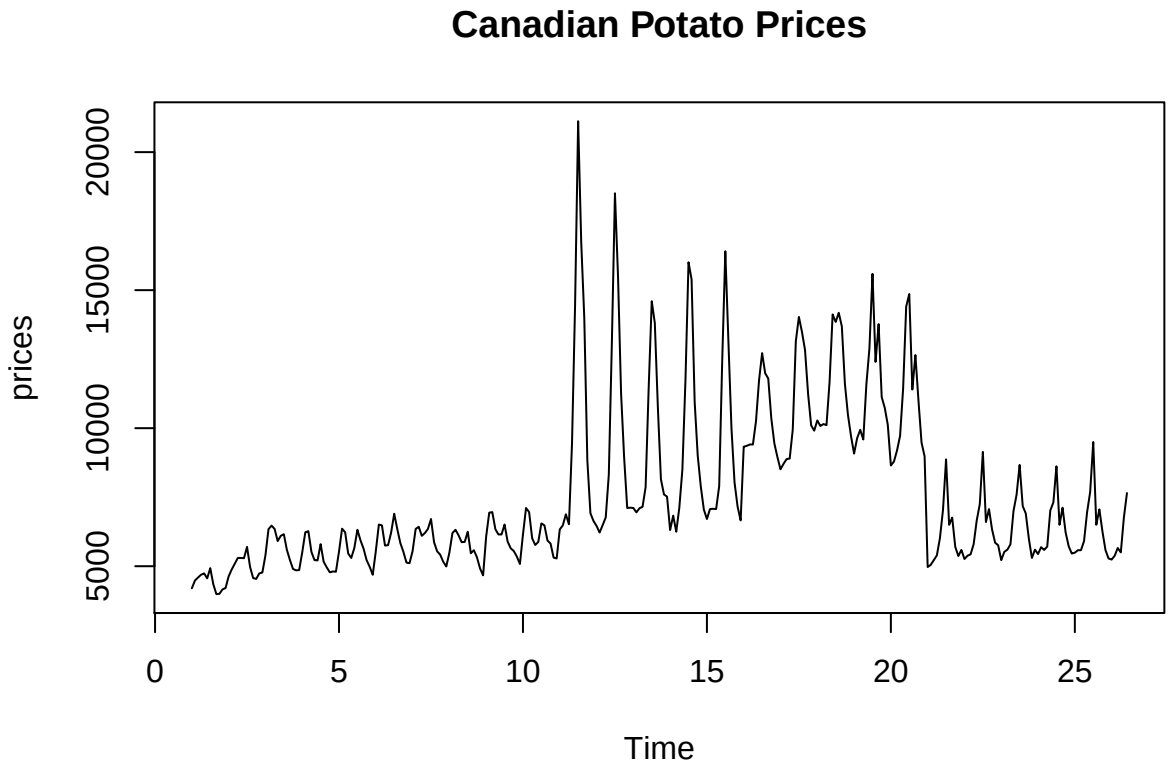
Yuhan Hu

2020/06/02

Practice

Exploratory data analysis and pre-processing

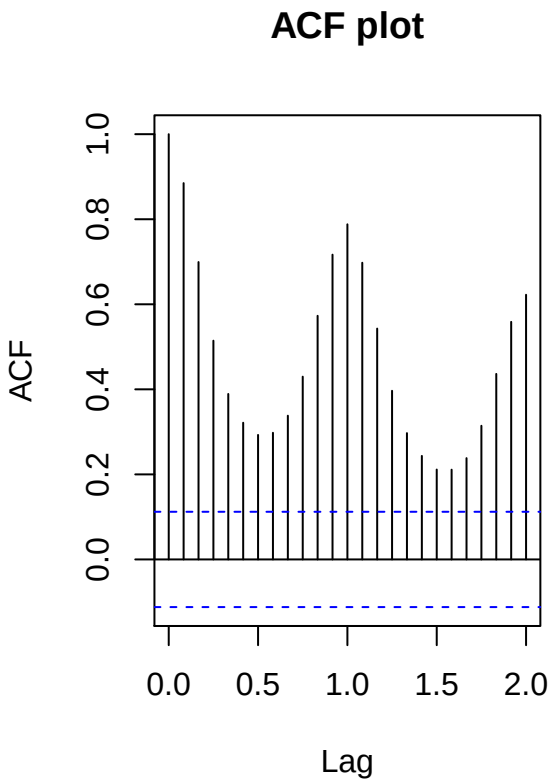
Here we are investigating change of Canadian potato prices. Though no extra information about the data(i.e. years the data was collected, frequency of data) was given, since the data measures prices changes nationwide while we have a large extent of change, it is unlikely to be collected in short time period. Therefore, this is monthly data. In addition, from the plot of time series, the data is clearly not stationary since the data is not varying around a horizontal line(i.e. the mean is not constant), and the extent of fluctuation is varying(i.e. non-constant variance).

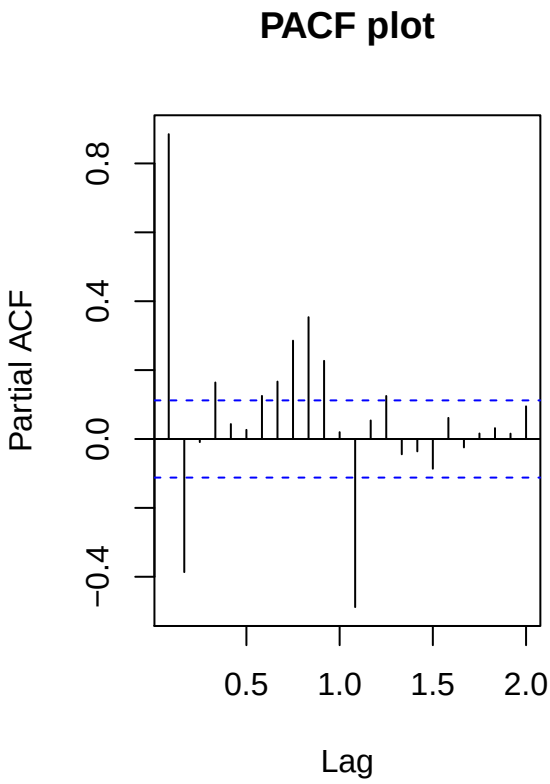


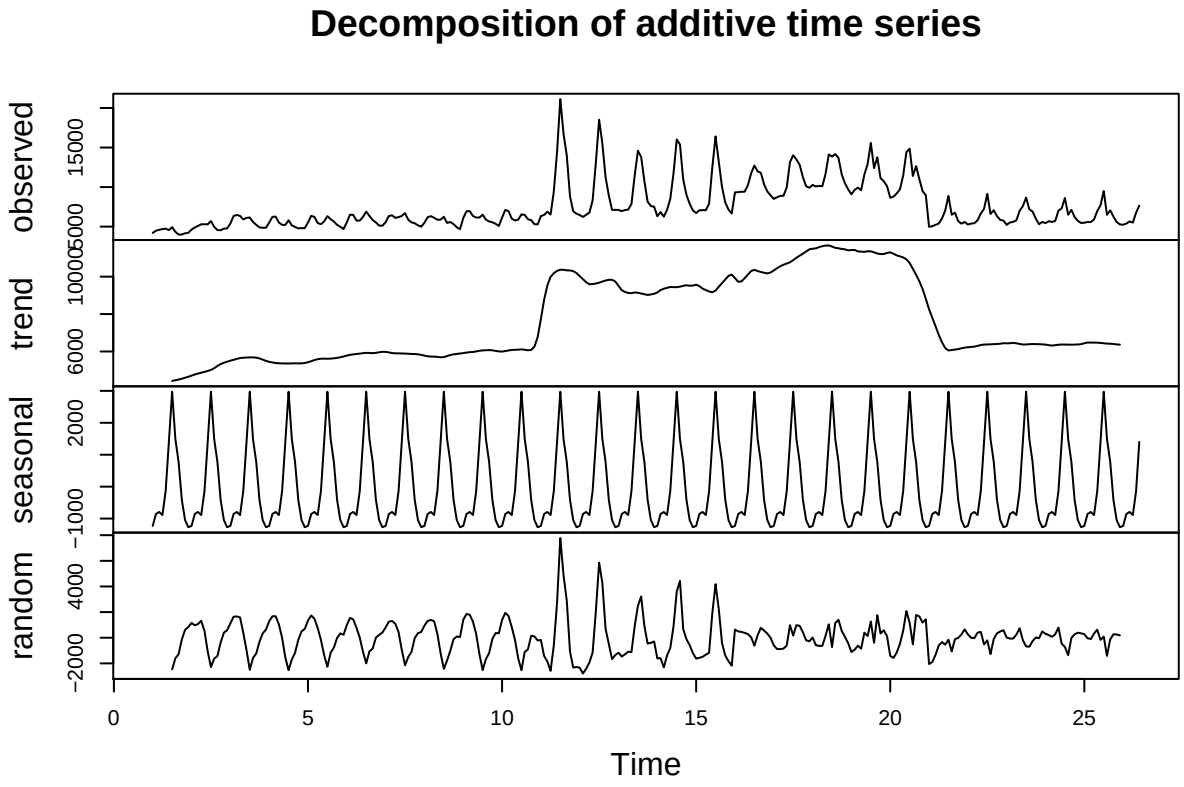
The ACF has repeating pattern that are fading as lag increases, this represent existence of seasonlity, which can also be found on the original time series plot. Also, trend can be found on the original plot since the data point are not flutuating around constant mean.

Since the variance is not constant, we may want to do a BoxCox tranformation. Corresponding table are shown in the Appendix. However, all returned values are close to 1; thus, BoxCox transformation is not feasible for our dataset.

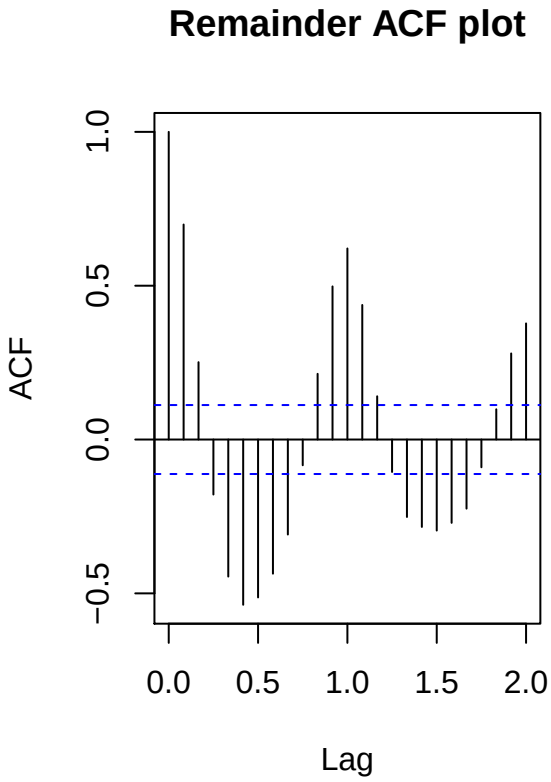
Then the data is decomposed using classical decomposition for a closer look on seasonality and trend. It is shown clearly that the data is seaonal and has trend since the trend plot is not horizontal line and the seasonal plot has repeating pattern. Therefore, we extract remainder to construct a more stationary time series for further analysis.

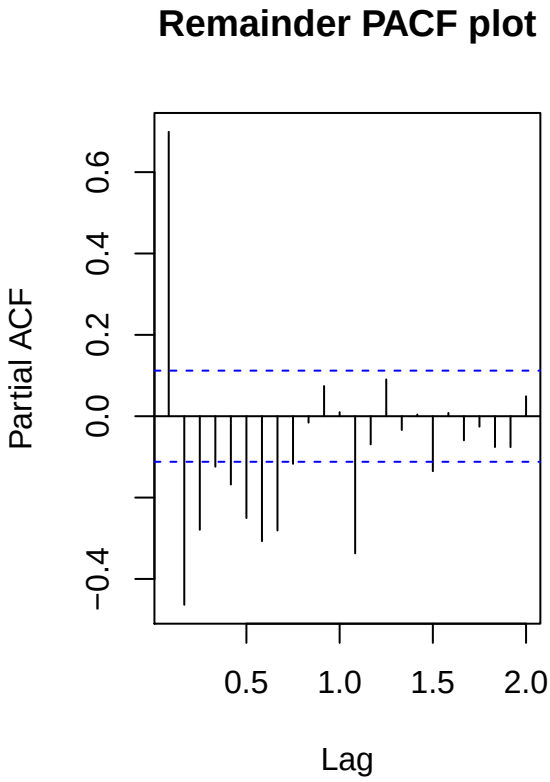





Decomposition of additive time series
observed
trend
seasonal
random
15000
5000
10000
6000
2000
−1000
4000
−2000
0
5
10
15
20
25
Time

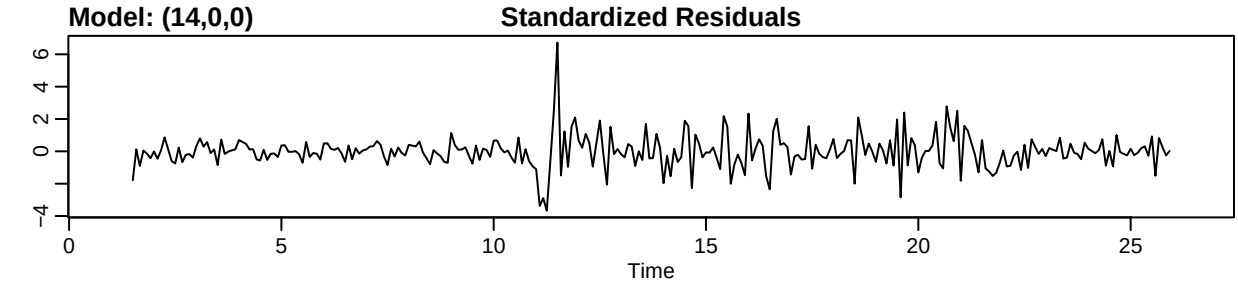
Model

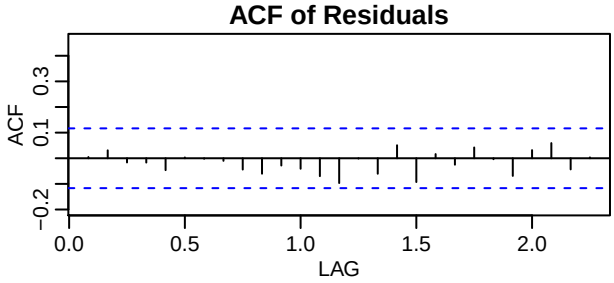


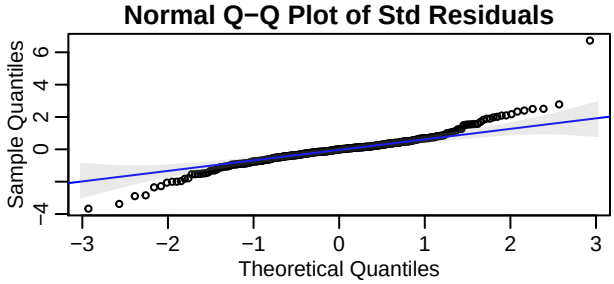


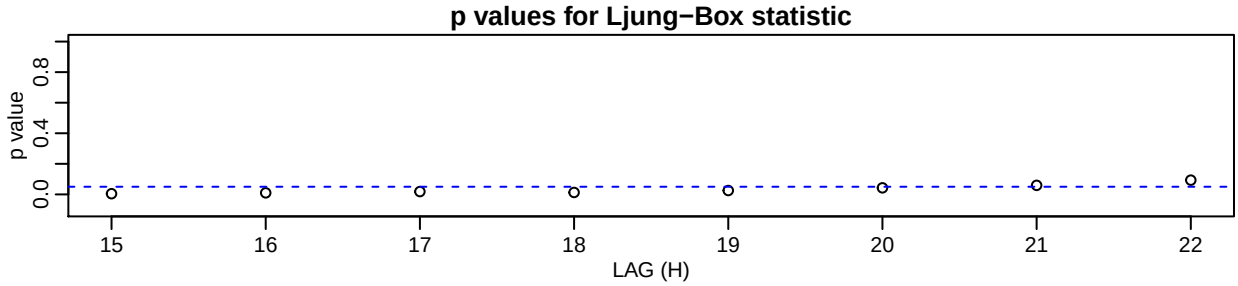
Since the original time series is not stationary, and simple transformation can not solve the problem, model is built on the remainder part from classical decomposition.AR(14) model is chosen here since ACF tails off and PACF cuts off after lag 14. However, as the ACF plot shown, seasonlity persists in the remainder component. There, in order to compare goodness of fit, we also use built-in function auto.arima() to get an auto-fitted model with seasonality for the remainder component. auto.arima() function gives arima model(2,0,1)(1,0,1)[12], where (2,0,1) is the orders of AR/MA terms, (1,0,1) is orders of seasonal AR/MA terms, and 12 is the span of the periodic seasonal behavior.

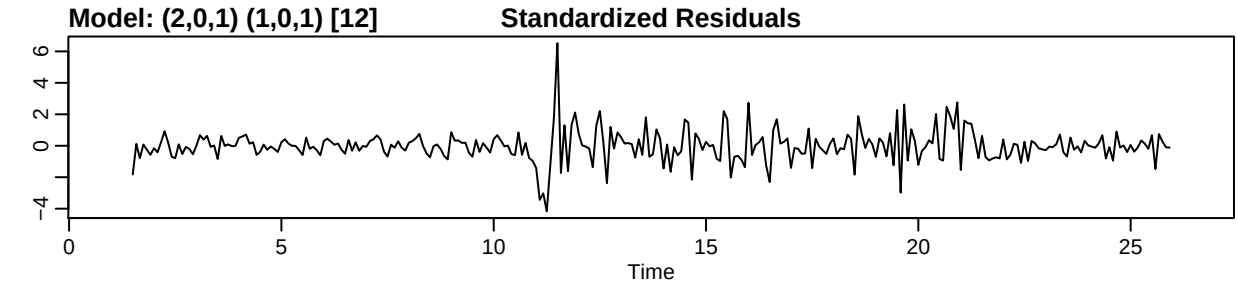
Model comparision and Limitation

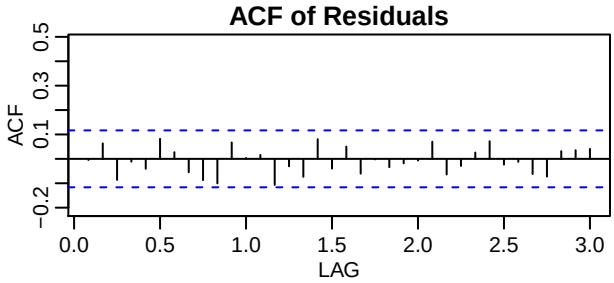


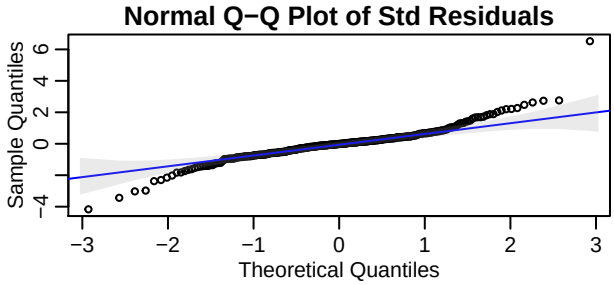


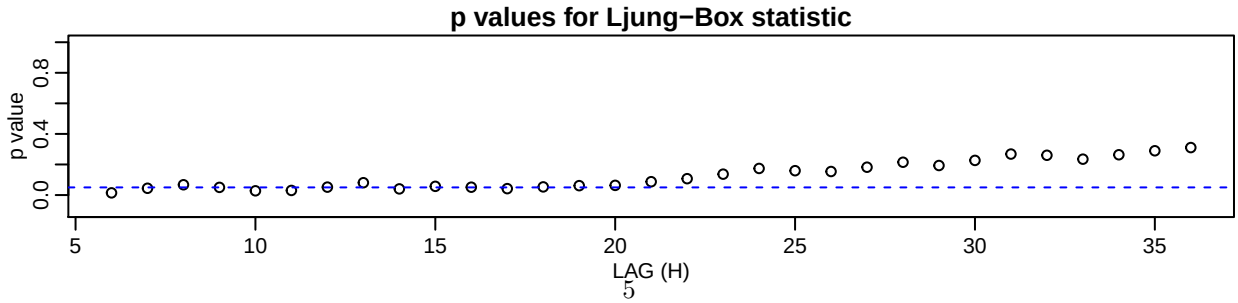










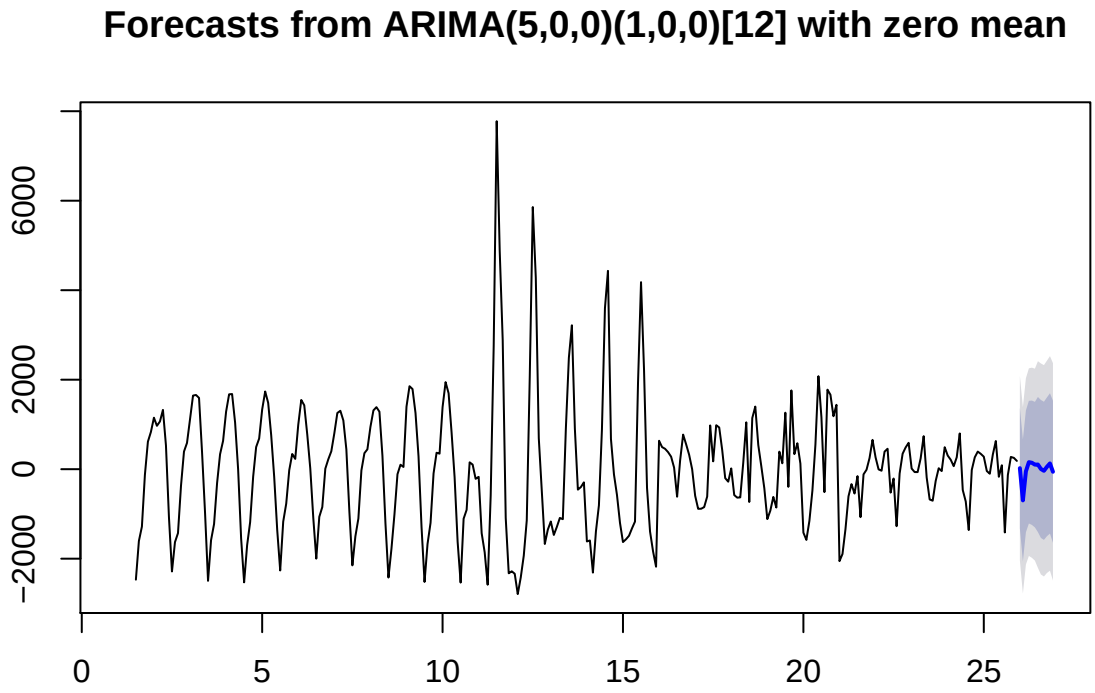


Compare the Ljung-Box plot for these two models, the AR(14) model has no Ljung-Box test value above the bound ,the null hypothese is rejected, the residuals are not independent, while most points on Ljung-Box plot of the auto-generated arima(2,0,1)(1,0,1)[12] model are above significant bound. Therefore, we prefer the auto-generated arima(2,0,1)(1,0,1)[12] model.

As for the chosen arima(2,0,1)(1,0,1)[12] model, the residual ACF looks good since all auto-correlation are within the bound. The normal qq plot is not perfect, the normal qq line is close to y=x, but the difference can be detected from the graph. Deviation from normality means the quality of fit may be further improved with proper transformation on data. The residuals seems to have constant 0-mean & nearly constant variance, and it has thiner tail than the normal, but otherwise symmetric. Overall, this is a good fit.

Prediction

Here we use forecast() function for 1-step to 12-step predictor, with predicted value, 95% CI and 80% CI shown on the graph.



Appendix

BoxCox

```
##          Jan       Feb       Mar       Apr       May       Jun       Jul
## 1  0.9998375 0.9998524 0.9998577 0.9998624 0.9998646 0.9998563 0.9998728
## 2  0.9998596 0.9998707 0.9998792 0.9998869 0.9998869 0.9998866 0.9999002
## 3  0.9998901 0.9999179 0.9999211 0.9999182 0.9999064 0.9999117 0.9999133
## 4  0.9998938 0.9999151 0.9999162 0.9998945 0.9998840 0.9998837 0.9999032
## 5  0.9998948 0.9999184 0.9999151 0.9998925 0.9998869 0.9998990 0.9999174
## 6  0.9998951 0.9999220 0.9999213 0.9999017 0.9999020 0.9999149 0.9999307
## 7  0.9998954 0.9999182 0.9999201 0.9999117 0.9999143 0.9999182 0.9999266
## 8  0.9998935 0.9999143 0.9999174 0.9999122 0.9999053 0.9999056 0.9999156
## 9  0.9999112 0.9999316 0.9999320 0.9999179 0.9999130 0.9999130 0.9999220
## 10 0.9999130 0.9999350 0.9999320 0.9999092 0.9999023 0.9999056 0.9999230
## 11 0.9999177 0.9999213 0.9999303 0.9999223 0.9999699 1.0000065 1.0000284
## 12 0.9999208 0.9999149 0.9999216 0.9999279 0.9999555 0.9999995 1.0000217
## 13 0.9999350 0.9999318 0.9999348 0.9999358 0.9999484 0.9999884 1.0000072
## 14 0.9999172 0.9999292 0.9999156 0.9999352 0.9999580 0.9999904 1.0000132
## 15 0.9999266 0.9999342 0.9999344 0.9999342 0.9999491 0.9999956 1.0000148
## 16 0.9999685 0.9999688 0.9999694 0.9999694 0.9999781 0.9999905 0.9999971
## 17 0.9999582 0.9999609 0.9999631 0.9999633 0.9999752 0.9999997 1.0000044
## 18 0.9999784 0.9999765 0.9999772 0.9999768 0.9999900 1.0000049 1.0000035
## 19 0.9999655 0.9999719 0.9999751 0.9999714 0.9999893 0.9999981 1.0000116
## 20 0.9999601 0.9999619 0.9999669 0.9999728 0.9999883 1.0000063 1.0000084
## 21 0.9998744 0.9998776 0.9998840 0.9998901 0.9999095 0.9999336 0.9999629
## 22 0.9998855 0.9998897 0.9998915 0.9999032 0.9999255 0.9999377 0.9999663
## 23 0.9998840 0.9998945 0.9998971 0.9999029 0.9999328 0.9999439 0.9999603
## 24 0.9998918 0.9998999 0.9998967 0.9999005 0.9999332 0.9999389 0.9999597
## 25 0.9998935 0.9998964 0.9998964 0.9999064 0.9999320 0.9999456 0.9999704
## 26 0.9998848 0.9998897 0.9998990 0.9998938 0.9999277 0.9999449
##          Aug       Sep       Oct       Nov       Dec
## 1  0.9998457 0.9998250 0.9998256 0.9998352 0.9998381
## 2  0.9998744 0.9998568 0.9998553 0.9998646 0.9998660
## 3  0.9998967 0.9998840 0.9998715 0.9998694 0.9998698
## 4  0.9998818 0.9998740 0.9998664 0.9998677 0.9998673
## 5  0.9999073 0.9998983 0.9998840 0.9998748 0.9998624
## 6  0.9999182 0.9999044 0.9998945 0.9998807 0.9998799
## 7  0.9999053 0.9998951 0.9998911 0.9998818 0.9998752
## 8  0.9998928 0.9998964 0.9998877 0.9998724 0.9998615
## 9  0.9999059 0.9998986 0.9998954 0.9998887 0.9998788
## 10 0.9999213 0.9999070 0.9999038 0.9998873 0.9998862
## 11 1.0000160 1.0000043 0.9999624 0.9999311 0.9999250
## 12 1.0000112 0.9999869 0.9999637 0.9999350 0.9999352
## 13 1.0000033 0.9999826 0.9999530 0.9999441 0.9999427
## 14 1.0000107 0.9999844 0.9999647 0.9999489 0.9999338
## 15 0.9999998 0.9999758 0.9999513 0.9999360 0.9999255
## 16 0.9999923 0.9999909 0.9999791 0.9999699 0.9999639
## 17 1.0000016 0.9999978 0.9999867 0.9999767 0.9999748
## 18 1.0000052 1.0000027 0.9999896 0.9999803 0.9999727
## 19 0.9999950 1.0000031 0.9999858 0.9999826 0.9999769
## 20 0.9999880 0.9999966 0.9999848 0.9999702 0.9999643
## 21 0.9999218 0.9999277 0.9999002 0.9998894 0.9998967
## 22 0.9999241 0.9999342 0.9999177 0.9999047 0.9999020
```

```
## 23 0.9999364 0.9999309 0.9999090 0.9998869 0.9998971
## 24 0.9999218 0.9999352 0.9999149 0.9999008 0.9998925
## 25 0.9999218 0.9999340 0.9999159 0.9998964 0.9998862
## 26
```

```
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)
library(cansim)
library(tidyverse)
library(forecast)
library(Metrics)
library(seasonal)
library(ggplot2)
library(forecast)
library(astsa)

#data import and exploratory plot
data <- read.csv('1001311626.csv')
prices <- ts(data$Canadian.Potato.Prices,frequency = 12)
plot(prices,main = 'Canadian Potato Prices')
BoxCox(prices,lambda = 'auto')
par(mfrow=c(1,2))
acf(prices, main = 'ACF plot')
pacf(prices, main = 'PACF plot')

#Decomposition
dc <- decompose(prices)
plot(dc)
Remainder <- dc$random
par(mfrow=c(1,2))
acf(Remainder,na.action = na.pass, main = 'Remainder ACF plot')
pacf(Remainder,na.action = na.pass, main = 'Remainder PACF plot')
model.ar <- arima(Remainder, c(14,0,0))
model.auto <- auto.arima(Remainder, seasonal = TRUE)
model.ar
model.auto
a <- sarima(Remainder,14,0,0)
sarima(Remainder,2,0,1,P=1,D=0,Q=1,S=12)
fore <- forecast(model.auto,h=12)
plot(fore)
out <- c(prices,as.numeric(fore$mean)+as.numeric(tail(dc$seasonal,n=12))+as.numeric(tail(na.omit(dc$tre
write.csv(out,'submission/1001311626.csv',row.names = FALSE,col.names = FALSE)
BoxCox(prices,lambda = 'auto')
```