STA457A1

Yuhan Hu

2020/06/01

Theory

1.

A.

$E(Y|X=-1) = (-1) \times P(Y=-1|X=-1) + 0 \times P(Y=0|X=-1) + 1 \times P(Y=1|X=-1)$

$= (-1) \times \frac{0.05}{0.05+0.1+0.15} + 1 \times \frac{0.15}{0.05+0.1+0.15} = \frac{1}{3}$

$E(Y|X=0) = (-1) \times P(Y=-1|X=0) + 0 \times P(Y=0|X=0) + 1 \times P(Y=1|X=0)$

$= (-1) \times \frac{0.15}{0.15+0.15+0.1} + 1 \times \frac{0.1}{0.15+0.15+0.1} = -\frac{1}{8}$

$E(Y|X=1) = (-1) \times P(Y=-1|X=1) + 0 \times P(Y=0|X=1) + 1 \times P(Y=1|X=1)$

$= (-1) \times \frac{0.15}{0.15+0.15} + 1 \times \frac{0.15}{0.15+0.15} = 0$

Therefore, $E[(Y-g(X))^2] = E[E[Y-g(X)]^2|X]$

$= E[(Y-g(X=-1))^2|X=-1] \times P(X=-1) + E[(Y-g(X=0))^2|X=0] \times P(X=0) + E[(Y-g(X=1))^2|X=1] \times P(X=1)$

$= E[(Y-\frac{1}{3})^2] \times 0.3 + E[(Y-(-\frac{1}{8}))^2] \times 0.4 + E[(Y-0)^2] \times 0.3$

$E[Y^2 - \frac{2}{3}Y + \frac{1}{9}|X=-1] \times 0.3 + E[Y^2 + \frac{1}{4}Y + \frac{1}{64}|X=0] \times 0.4 + E[Y^2|X=1] \times 0.3$

where $E[Y] = -1 \times 0.35 + 0 \times 0.25 + 1 \times 0.4 = 0.05$,$E[Y^2|X=-1] = \frac{2}{3}, E[Y^2|X=0] = \frac{5}{8}, E[Y^2|X=1] = 1$

and $E[Y^2] = (-1)^2 \times 0.35 + 0^2 \times 0.25 + 1^2 \times 0.4 = 0.75$

Therefore, $E[(Y-g(X))^2] = 0.71$

b.

$E[(Y-\hat{Y})^2] = E[(Y-a-bX)^2]$

To find minimum value, take partial derivative of both a and b and set the result equals 0.

we have $E[(-2)(Y-a-bX)] = 0$

$E[(-2X)(Y-a-bX)] = 0$

Then we have $-2E[(Y-a-bX)] = 0 \rightarrow E[(Y-a-bX)] = 0 \rightarrow E[Y]-a-bE[X] = 0 \rightarrow a = E[Y]-bE[X]$

and $2E[(-X)(Y-a-bX)] = 0 \rightarrow E[(-X)(Y-a-bX)] = 0 \rightarrow E[XY]-aE[X]-bE[X^2] = 0$

subsititute a, we have $E[XY]-(E[Y]-bE[X])E[X]-bE[X^2] = 0 \rightarrow E[XY]-E[X]E[Y]+bE[X]^2-bE[X^2] = 0 \rightarrow b = \frac{cov(X,Y)}{var[X]}$

From Part A, we know $E[Y] = 0.05$,$E[Y^2] = 0.75$

also, $E[X] = 0$,$E[X^2] = 0.6$

then $Var[X] = 0.6 - 0^2 = 0.6$, $cov(X, Y) = E[XY] = 0.05 \times (-1)^2 + 0.15 \times (+1)^2 + (-1)(+1) \times 0.15 + (+1)(-1)0.15 = -0.1$

Then $a = 0.05, b = -\frac{1}{6}$

$E[(Y - \hat{Y})^2] = E[Y^2 - 2Y\hat{Y} + \hat{Y}^2] =$

2.

A.

$X^n_{n+1} = E[X_{n+1}|X_1...X_n] = E[\phi X_n + W_{n+1}|X_1...X_n] = \phi X_n$

$X^n_{n+2} = E[X_{n+2}|X_1...X_n] = E[\phi X_{n+1} + W_{n+1}|X_1...X_n] = E[\phi(\phi X_n + W_{n+1})|X_1...X_n] = \phi^2 X_n$

$Cov[(X_{n+1} - X^n_{n+1}), (X_{n+2} - X^n_{n+2})] = Cov[W_{n+1}, \phi W_{n+1} + W_{n+2}] = \phi Cov[W_{n+1}, W_{n+1}] = \phi\sigma^2_W$

b.

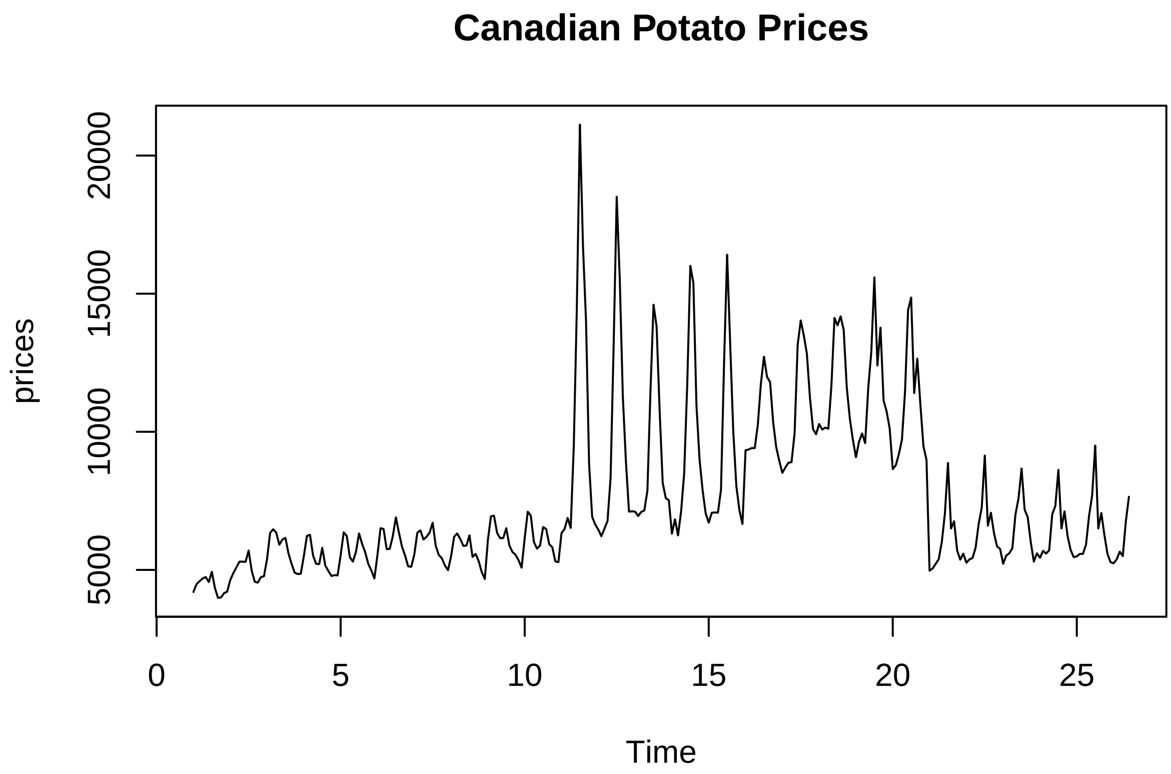
$X^{n-1}_n = E[X_n|X_1...X_{n-1}] = E[\phi X_{n-1} + W_n|X_1...X_{n-1}] = \phi X_{n-1}$

$Cov[(X_n - X^{n-1}_n), (X_{n+1} - X^n_{n+1})] = Cov[\phi X_{n-1} + W_n - (\phi X_{n-1}), \phi X_n + W_{n+1} - (\phi X_n)] = 0$

Practice

Exploratory data analysis and pre-processing

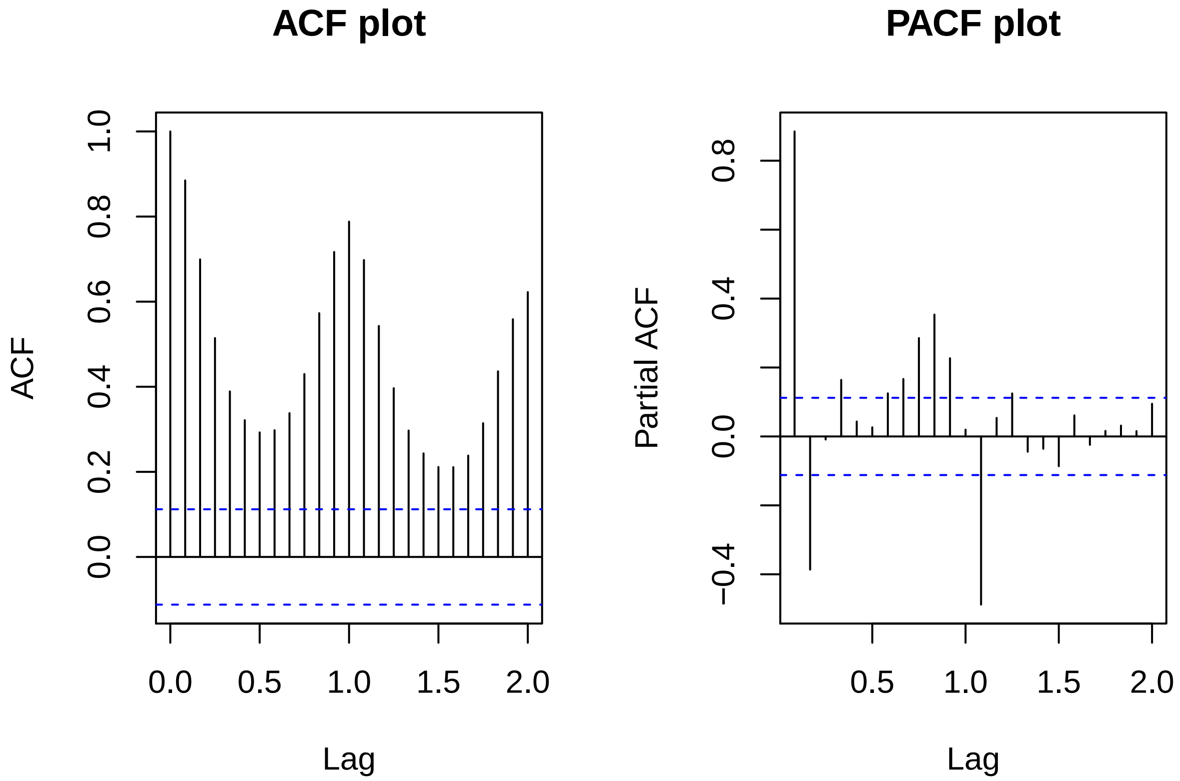
Here we are investigating change of Canadian potato prices. Though no extra information about the data(i.e. years the data was collected, frequency of data) was given, since the data measures prices changes nationwide while we have a large extent of change, it is unlikely to be collected in short time 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).

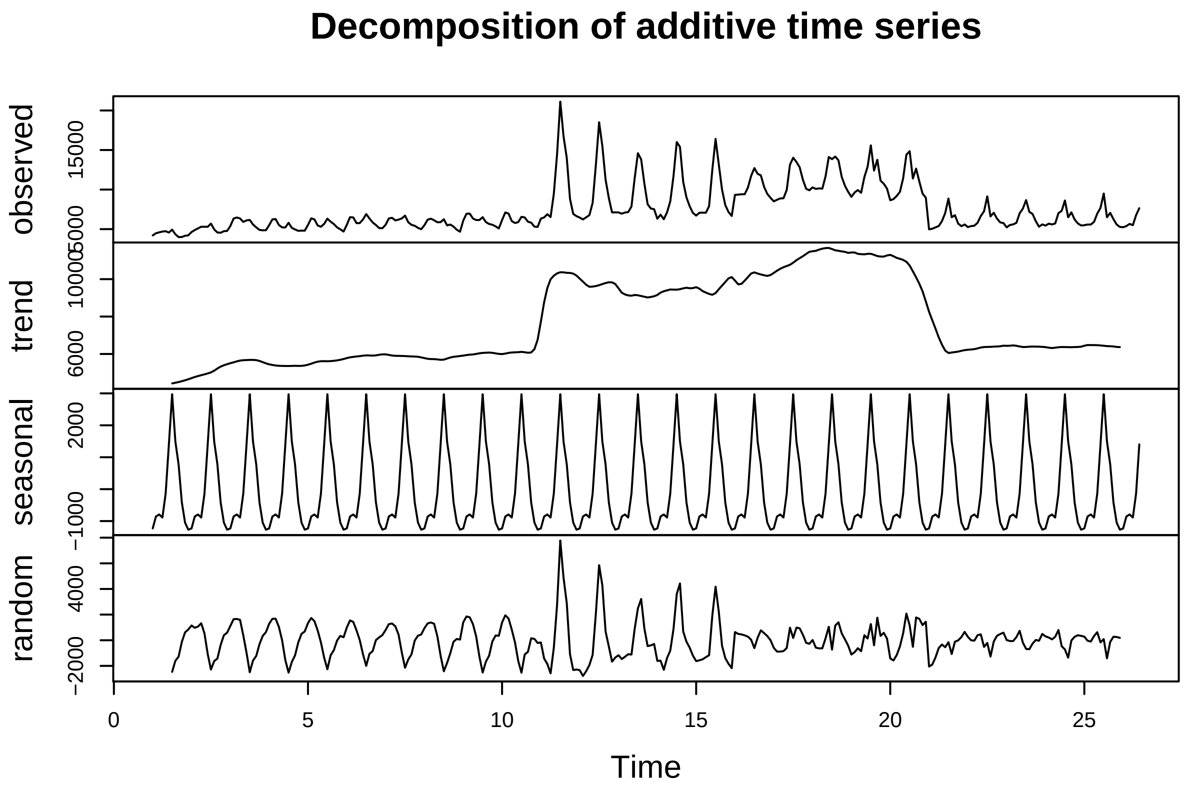


The ACF has repeating pattern that are fading as lag increases, this represent existence of seasonlity, which can also be found on the original time series plot. Also, trend can be found on the original plot since the data point are not flutuating around constant mean.

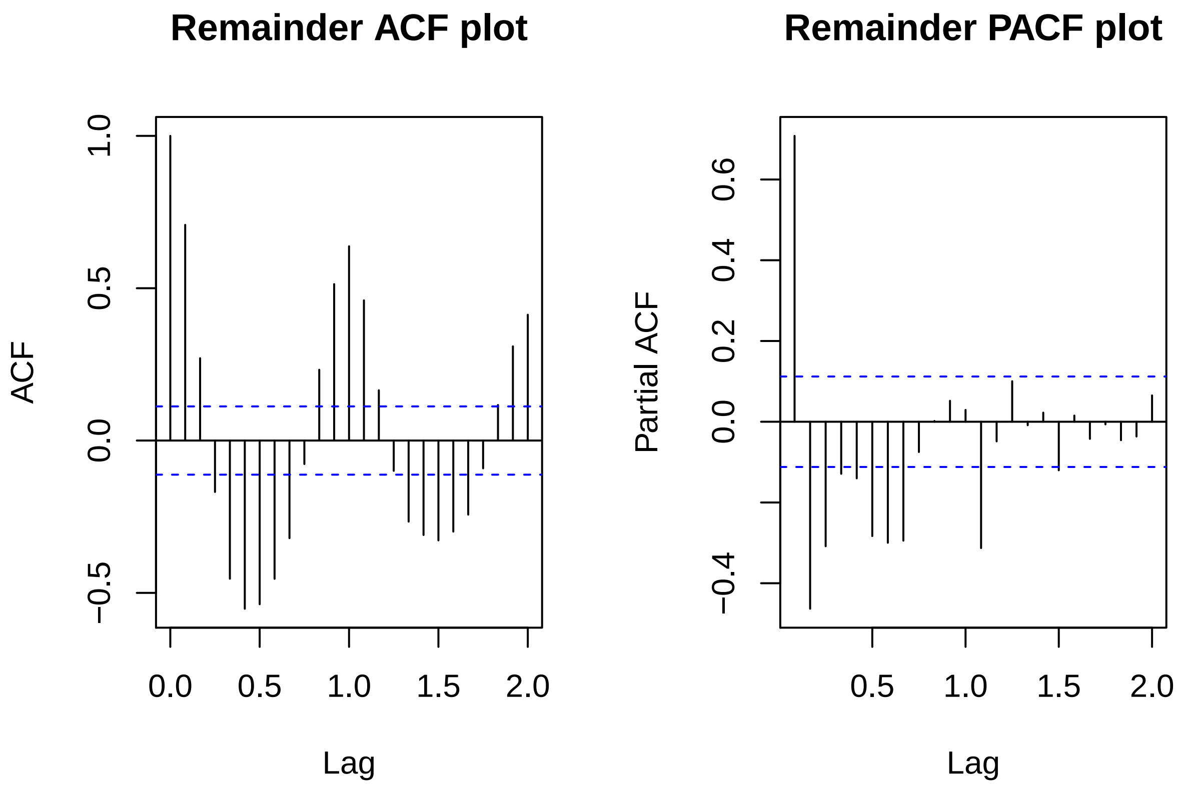
Since the variance is not constant, we may want to do a BoxCox tranformation. Corresponding table are shown in the Appendix. However, all returned values are close to 1; thus, BoxCox transformation is not feasible for our dataset.

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.


ACF plot
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
Lag
PACF plot
Partial ACF
0.8
0.4
0.0
−0.4
0.5
1.0
1.5
2.0
Lag


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

Model



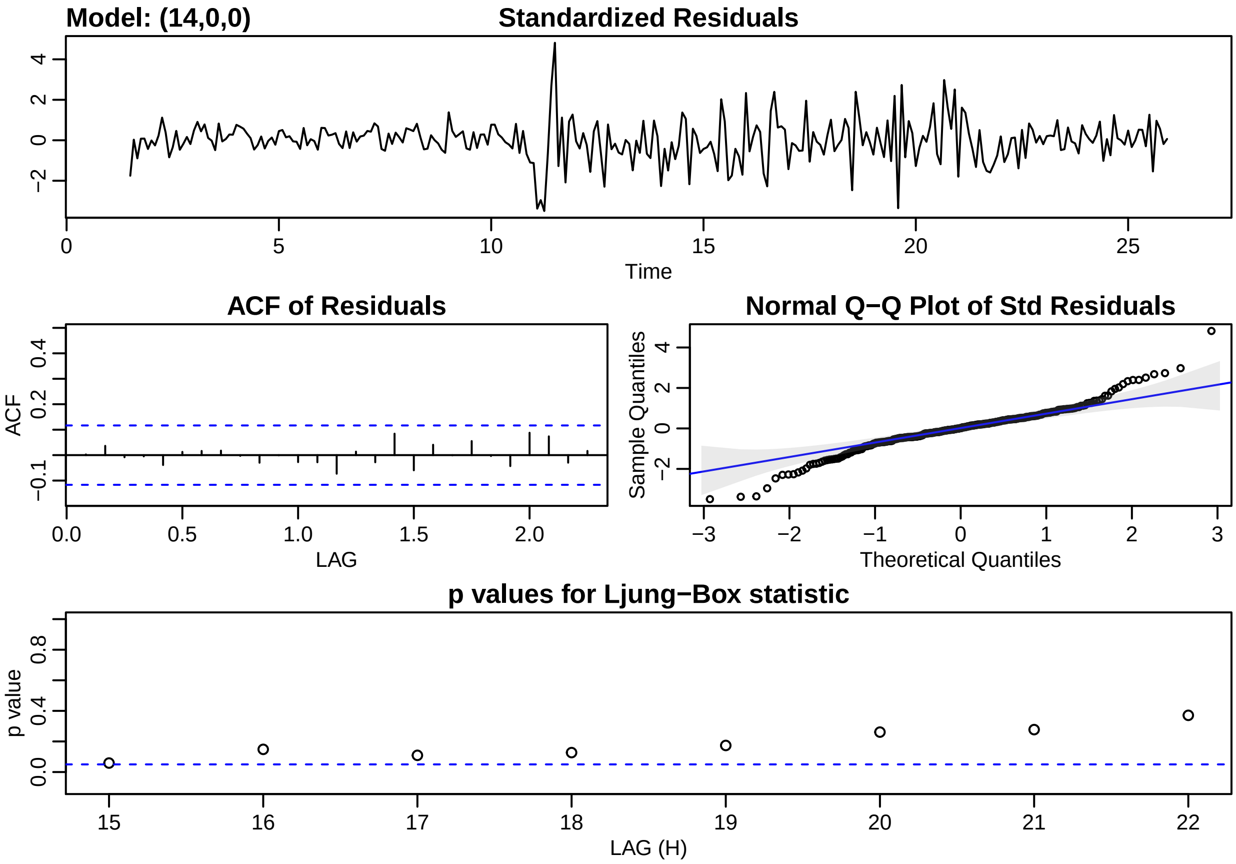
Since the original time series is not stationary, and simple transformation can not solve the problem, model is built on the remainder part from classical decomposition.AR(p) model is chosen here since ACF tails off and PACF cuts off. In addition, in order to compare goodness of fit, we also use built-in function auto.arima() to get an auto-fitted model.

Based on the p-value returend by performing Ljung-Box test, we can conclude that our AR(14) model is better than the auto-fitted AR(3) model, regardless of complexity.

```
## [1] 0.963734
```

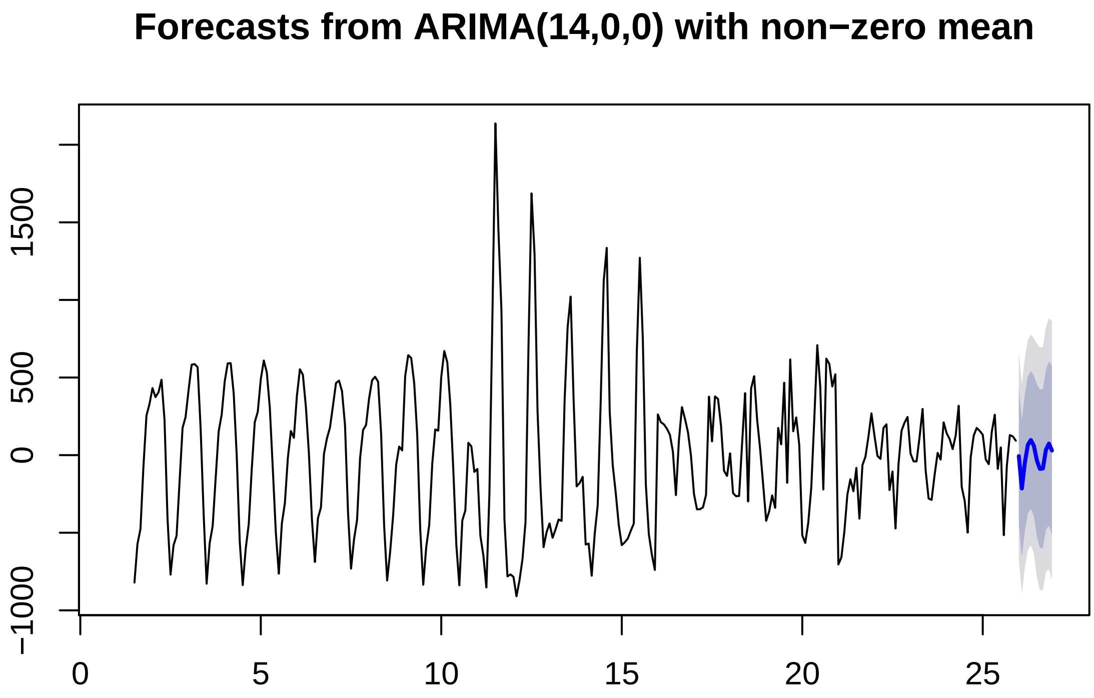
```
## [1] 0.7824435
```

Limitation



The residual ACF looks good since all auto-correlation are within the bound(except the auto-correlation beween the same data entry, which is always 1). The normal qq plot is not perfect, the normal qq line is close to y=x, but difference beween The residuals seems to have constant mean Despite model complexity introduced by the high order of model, this is a good fit.

Prediction


Forecasts from ARIMA(14,0,0) with non−zero mean
−1000
0
500
1500
0
5
10
15
20
25

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
```

```r
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
Remainder <- BoxCox(Remainder,lambda = 'auto')
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 = FALSE, stationary = TRUE)

Box.test(model.ar$residuals,type = c("Ljung-Box"))$p.value
Box.test(model.auto$residuals,type = c("Ljung-Box"))$p.value
sarima(Remainder,14,0,0)
fore <- forecast(model.ar,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')
```