BAN 673 Time Series Analytics

Spring Semester 2019

Final Project –

US Candy Production Time Series Analysis

Submitted to: Dr. Zinovy Radovilsky, instructor of BAN 673

Submitted by:

Wenxuan Liang（gf5548), Jing Zhang (js9673), Lu Dai (qx8329),

Hsin-Yi Chiu (sj8767), Chian Long (sy6252)

Table of Contents

Summary

We can see that candy is a mainstream food in many festivals in the United States, and at the same time people are increasingly paying attention to the effects of sugar on body weight and health. Our group is very interested in predicting the trend of candy and the characteristics of sales based on U.S candy production data over the past 45 years, such as whether seasonality is obvious.

Our dataset tracks industrial production every month from January 1972 to December 2016 to predict the future 2017 and 2018 production trend. The entire dataset is 540 periods.

There are three main forecasting models used in our report: Holt-Winter's Exponential Smoothing Model (HW), Two-Level Forecast Model (Regression + AR (1) models), and Auto ARIMA Model. Before developing the forecasting models, we defined the partition series of training and validation period.

Introduction

Confectionery market comprises variety of products such as chocolates, and various sugar-based products. In addition, due to the health issue is rising up in past few years, more and more people start paying attention to the risks that caused by sugar. Hence, other than traditional confectionery products, it also includes therapeutic and dietetic confectioneries that differ in formulations from traditional confections. The global Confectionery market is growing at a steady pace owing to high demand. Besides, U.S is the sixth biggest exporting country of confectionery industry. Therefore, we would like to know probable demand in the future by using variety of forecasting methods.

The dataset contains U.S candy monthly production of 45 years data from Jan. 1972 to Dec 2016. The are 540 data points in total based on volume of MKG (meter-kilograms). On the basis of the dataset, we will predict the next 24 periods into the future from 2017 and 2018.

Our data is retrieved from Kaggle dataset of US Candy Production by Month. This dataset original data is from FRED, Federal Reserve Bank of St. Louis; (US), Industrial Production: Nondurable Goods: Sugar and confectionery product [IPG3113N]. Data has 2 variables:

Main Chapter

Define Goal

According to the research, we consider U.S holiday candy consumption should reflect affect industrial production volume which also reflects strong seasonality or trend. To prove our hypothesis, we got the U.S candy production monthly time series data of 45 years data from Jan. 1972 to Dec 2016, the total is 540 monthly records with the unit of production volume based on M KG (meter-kilograms). We also like to find is there any trend or pattern in this time series data? Whether American consumption more candy than before or the candy production hinders by the growing health consciousness in recent years? Which months have the highest candy production? Based on this long-term and numerous historical data, can we precisely forecast the candy production in the next 2 years, how is the accuracy?

Get Data and Pre-process Data

Our data is retrieved from Kaggle dataset of US Candy Production by Month. This dataset original data is from FRED, Federal Reserve Bank of St. Louis; (US), Industrial Production: Nondurable Goods: Sugar and confectionery product [IPG3113N]. Data has 2 variables:

Observation_date: 540 observation records by month from Jan. 1972 to Dec. 2016

Volume_mkg: US candy production volume based on M KG (meter-kilograms)

To analyze time series data, we convert data into time series dataset (Candy.ts). Apply summary function on entire time series data (Candy.ts) to get the statistic description including mean, median, 25th and 75th quartiles, and min, max.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
339.1	583.5	683.6	673.1	768.3	936.5

To do further analyze and build model, we need to pre-process data:

Step 1: Check missing value:

There is no missing value.

```
> # Check for missing values
> sum(is.na(Candy.ts))
[1] 0
```

Step 2: Check the frequency and the cycle of the time series:

This is time series data with an apparent cycle of 12 months.

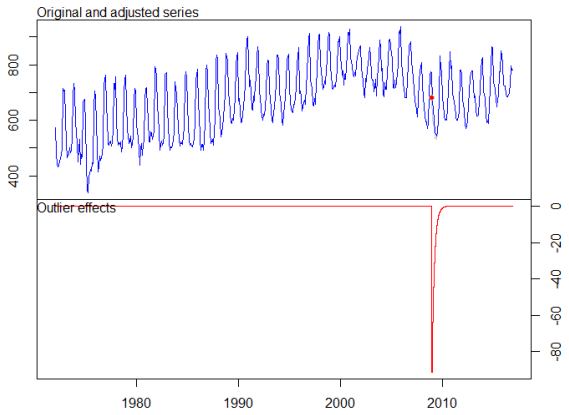
```
> # Check the frequency of the time series data
> frequency(Candy.ts)
[1] 12
> hist(Candy.ts)
> # Check the cycle of the time series
> cycle(Candy.ts)
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1972   1   2   3   4   5   6   7   8   9  10  11  12
1973   1   2   3   4   5   6   7   8   9  10  11  12
1974   1   2   3   4   5   6   7   8   9  10  11  12
1975   1   2   3   4   5   6   7   8   9  10  11  12
1976   1   2   3   4   5   6   7   8   9  10  11  12
1977   1   2   3   4   5   6   7   8   9  10  11  12
1978   1   2   3   4   5   6   7   8   9  10  11  12
1979   1   2   3   4   5   6   7   8   9  10  11  12
1980   1   2   3   4   5   6   7   8   9  10  11  12
1981   1   2   3   4   5   6   7   8   9  10  11  12
1982   1   2   3   4   5   6   7   8   9  10  11  12
```

Step 3: Check outliers:

We can find out there is only one record is considered as the outlier, which is Dec. 2008 data at No. 444 row. In time series models, outliers should be further investigated to ensure they are not part of a seasonal (or other cyclical) trend that pops up every so often and may appear as an

aberrant value. This outlier may be real values that should be further investigated. Consider only one outlier may not significantly affect the entire data set, we decide not to eliminate the outlier.

```
> data.ts.outliers
Series: Candy.ts
Regression with ARIMA(1,0,1)(0,1,2)[12] errors

Coefficients:
         ar1      ma1     sma1     sma2     TC444
      0.9170  -0.2431  -0.6048  -0.1114  -91.2204
s.e.  0.0217   0.0539   0.0457   0.0433   21.6512

sigma^2 estimated as 606.7:  log likelihood=-2442.72
AIC=4897.43   AICc=4897.59   BIC=4923.05

Outliers:
  type ind    time coefhat  tstat
1   TC 444 2008:12  -91.22 -4.213
```

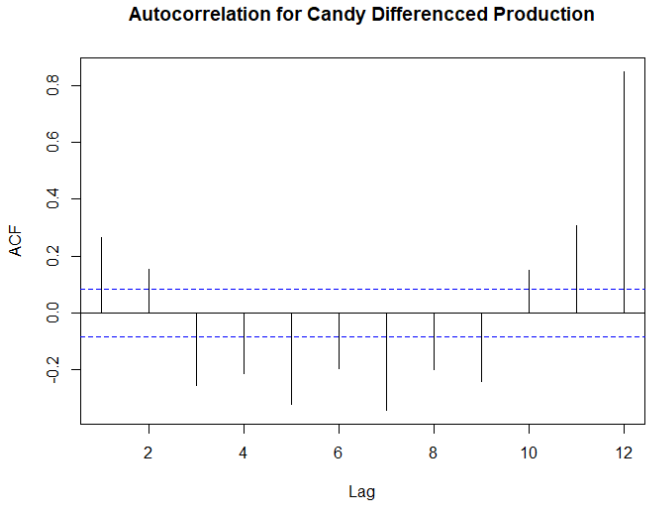


Step 4: Check Predictability

We used the first differencing of the entire data and Asf() function to prove the worldwide production is predictable. A partial output of the AR(1) model for Candy.ts time series data is presented below. ARIMA (1,0,0) is an autoregressive model with order 1 (lag-1), no differencing and no moving average model. The coefficient of the AR(1) variable is 0.8744, is well below 1. The upper value of this coefficient (the population value of this coefficient) will be 0.8744+ 2*0.0205 = 0.9154, which is still below 1, and not in the confidence of 95%. Therefore, Candy.ts time series is not a random walk and is predictable.

```
> Candy.ar1<- Arima(Candy.ts, order = c(1,0,0))
> summary(Candy.ar1)
Series: Candy.ts
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1      mean
      0.8744  673.8271
s.e.  0.0205   19.6309

sigma^2 estimated as 3428:  log likelihood=-3007.62
AIC=6021.24   AICc=6021.28   BIC=6034.16

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE     MASE      ACF1
Training set 0.2297317 58.44562 42.87655 -0.7715526 6.460161 1.241808 0.3014809
>
```

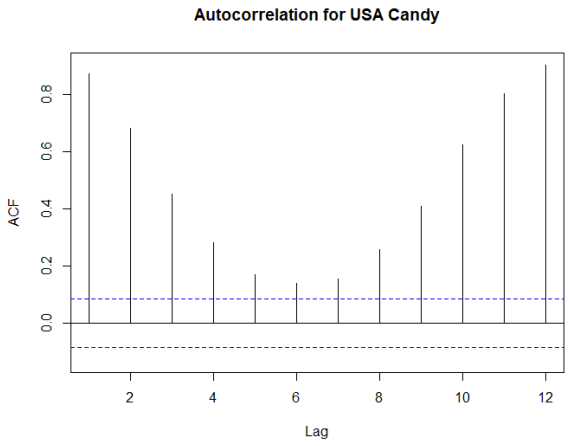
The autocorrelation plot of the first differencing for Candy.ts data is presented below.



All autocorrelation coefficients of the first-differenced data are statistically significant and not within the horizontal threshold. Therefore, using the first differencing, we can confirm that Candy.ts isn't a random walk and is predicable.

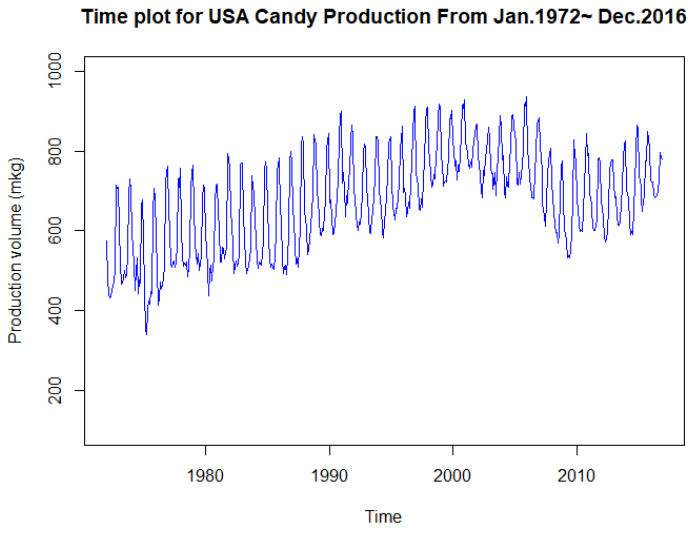
Explore Visualize Series

We apply the acf() function to identify possible time series components and get the autocorrelation for USA candy production with 12 lags. The autocorrelation chart is shown below:



All autocorrelation coefficient lags show positive and outside the band marked by the dotted blue lines are deemed to be statistically significantly (greater than zero). A positive autocorrelation coefficient in lag 1 is 0.874 which is close to one, which means that candy production at a given month is very similar to the next month that shows the significant upward and positive trend with data. In several lags gradually drops to zero but still outside the threshold, which is indicative of an upward trend component and of the level component in the candy dataset. The highest autocorrelation coefficient in lag 12 is 0.903, which is substantially higher than the horizontal threshold (significantly greater than zero), points to monthly seasonality.

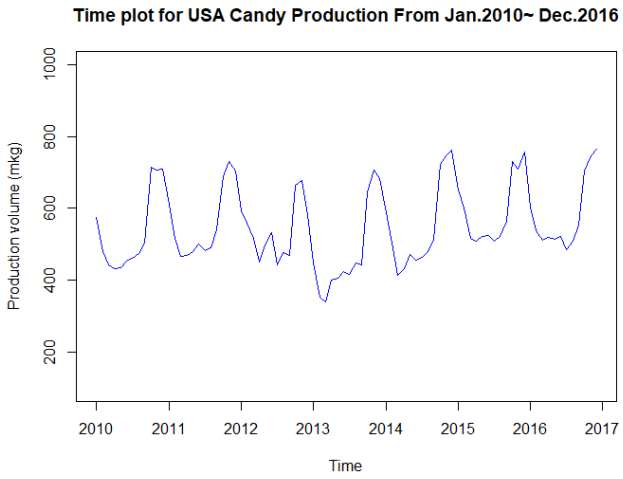
To visualize the entire data, we apply the plot() function to create a data plot with the historical data.



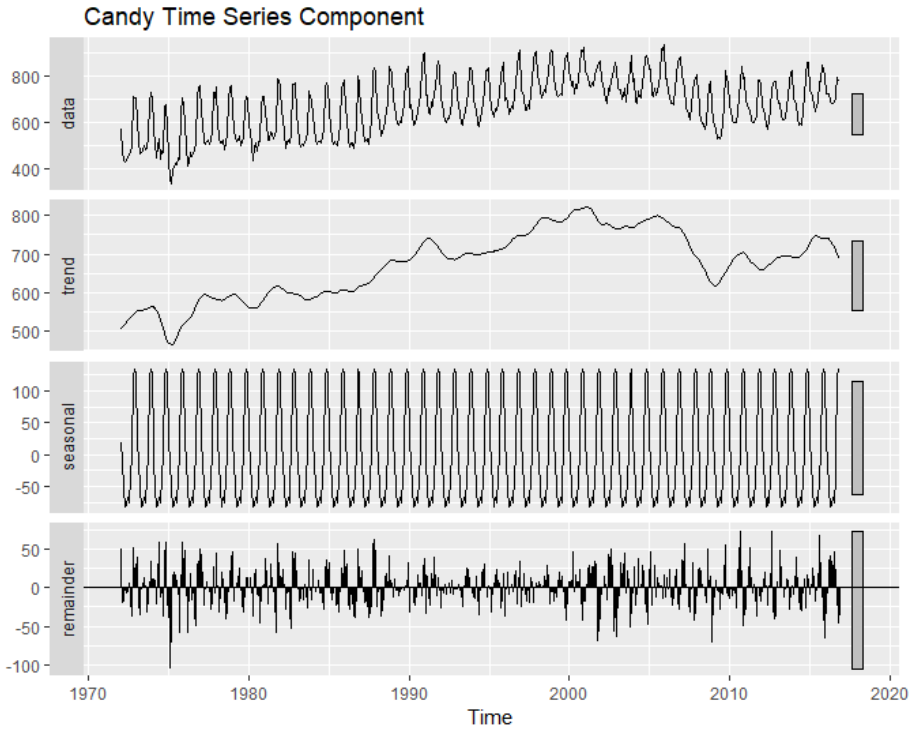
We can overview the trend and seasonality of entire data by the plot which shows an upward trend and seasonality is similar to the seasonal and trend components including small and noise fluctuations. The production volume at the beginning of the year 1972 is relatively low and slightly increased in several years and reach peak production until the year 2000. After that,

production rapidly decreased for 10 years long and then slightly increased at the end of the period.

To be closer view the monthly trend, we zoom the plot from year 2010~2016 shows below. The plot represents the pattern with the low production volume at the beginning of each year (approximately February-March) and peak production volume at the end of each year (approximately October-December).



The sales time series three components, trend, seasonal and the remainder are shown separately in the below graph. These components can be added together to reconstruct the original data. Notice that the seasonal component represents the data has seasonality with variations that repeat over. The trend shows that the sales pattern with the sales increasing steadily start from the year 1972, rapidly drop in 1975, but recover quickly with a steadily increasing trend until 2000. Then, the production experienced strongly recession till 2009. At the end of periods, the production keeps a slowly growing trend.



Partition Series

Before developing the forecasting model, we need to define the partition series of training and validation period. The main reason for partitioning is that we need to check our forecasting model corresponds too closely or exactly to a set of data or noise. Overfitting would let new data affect the forecasting model's performance. The earlier period is designated as the training period, and typically, is 70-80% of the whole time series data. Our entire data set is 540 periods, so set the training period = 432 and validation period =108.

Forecasting Methods

1. Holt-Winter's Exponential Smoothing Model (HW)

Create Holt-Winter's exponential smoothing (HW) for candy training data.

A summary of the Holt-Winter's (HW) model with the automated selection of the model options and automated selection of the smoothing parameters for the training period is shown below:

```
> hw.ZZZ <- ets(candytrain.ts, model = "ZZZ")
> hw.ZZZ
ETS(A,N,A)

Call:
 ets(y = candytrain.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.5542
    gamma = 0.259

  Initial states:
    l = 560.2014
    s = 155.6461 173.3181 131.7837 -33.7294 -49.2861 -70.5861
           -84.8021 -75.3606 -83.9517 -58.9033 -28.461 24.3321

  sigma:  25.5427

     AIC     AICc      BIC
5436.990 5438.144 5498.017
```

This HW model has the (A, N, A) options, which means additive error, no trend, and additive seasonality. The optimal value for exponential smoothing constant (alpha) is 0.5542, no smoothing constant for trend estimate (beta), and smoothing constant for seasonality estimate (gamma) is 0.259. The alpha value of this model indicates that the model's level component tends to be more local.

The HW model's forecast in the first 24 periods of the validation period is presented below:

```
> hw.ZZZ.pred <- forecast(hw.ZZZ, h =nValid , level = 0)
> hw.ZZZ.pred
         Point Forecast     Lo 0     Hi 0
Jan 2008       720.6324 720.6324 720.6324
Feb 2008       708.3373 708.3373 708.3373
Mar 2008       658.4758 658.4758 658.4758
Apr 2008       625.3877 625.3877 625.3877
May 2008       631.6809 631.6809 631.6809
Jun 2008       632.1418 632.1418 632.1418
Jul 2008       628.8727 628.8727 628.8727
Aug 2008       688.1305 688.1305 688.1305
Sep 2008       726.0724 726.0724 726.0724
Oct 2008       784.1660 784.1660 784.1660
Nov 2008       797.1341 797.1341 797.1341
Dec 2008       806.4964 806.4964 806.4964
Jan 2009       720.6324 720.6324 720.6324
Feb 2009       708.3373 708.3373 708.3373
Mar 2009       658.4758 658.4758 658.4758
Apr 2009       625.3877 625.3877 625.3877
May 2009       631.6809 631.6809 631.6809
Jun 2009       632.1418 632.1418 632.1418
Jul 2009       628.8727 628.8727 628.8727
Aug 2009       688.1305 688.1305 688.1305
Sep 2009       726.0724 726.0724 726.0724
Oct 2009       784.1660 784.1660 784.1660
Nov 2009       797.1341 797.1341 797.1341
Dec 2009       806.4964 806.4964 806.4964
```

Using the entire data set, we received the following Holt-Winter's model with the automated selection of the model options and optimal smoothing parameters:

```
> HW.ZZZ <- ets(Candy.ts, model = "ZZZ")
> HW.ZZZ
ETS(A,N,A)

Call:
 ets(y = Candy.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.5647
    gamma = 0.2663

  Initial states:
    l = 544.539
    s = 135.0834 151.9658 144.6042 -15.4781 -48.0933 -77.5638
           -72.8003 -81.3808 -99.7611 -30.3201 -36.0869 29.8312

  sigma:  26.57

     AIC     AICc      BIC
6955.427 6956.343 7019.801
```

Like the previous model for training data, this HW model also has the (A, N, A) options, which means additive error, no trend, and additive seasonality. The optimal value for

exponential smoothing constant (alpha) is 0.5642, no smoothing constant for trend estimate (beta), and smoothing constant for seasonality estimate (gamma) is 0.2663. The alpha value of this model indicates that the model's level component tends to be even more local than that in the previous model for training data. The gamma value is a little bit higher than that in previous model for training data.

The model's forecast in 24 months of 2017-2018 is given below:

```
> HW.ZZZ.pred <- forecast(HW.ZZZ, h = 24 , level = 0)
> HW.ZZZ.pred
         Point Forecast     Lo 0     Hi 0
Jan 2017       680.5494 680.5494 680.5494
Feb 2017       669.8835 669.8835 669.8835
Mar 2017       652.8397 652.8397 652.8397
Apr 2017       610.0296 610.0296 610.0296
May 2017       587.7277 587.7277 587.7277
Jun 2017       595.7319 595.7319 595.7319
Jul 2017       602.5135 602.5135 602.5135
Aug 2017       653.2016 653.2016 653.2016
Sep 2017       694.7166 694.7166 694.7166
Oct 2017       777.5155 777.5155 777.5155
Nov 2017       786.2865 786.2865 786.2865
Dec 2017       780.1746 780.1746 780.1746
Jan 2018       680.5494 680.5494 680.5494
Feb 2018       669.8835 669.8835 669.8835
Mar 2018       652.8397 652.8397 652.8397
Apr 2018       610.0296 610.0296 610.0296
May 2018       587.7277 587.7277 587.7277
Jun 2018       595.7319 595.7319 595.7319
Jul 2018       602.5135 602.5135 602.5135
Aug 2018       653.2016 653.2016 653.2016
Sep 2018       694.7166 694.7166 694.7166
Oct 2018       777.5155 777.5155 777.5155
Nov 2018       786.2865 786.2865 786.2865
Dec 2018       780.1746 780.1746 780.1746
```

2. Quadratic trend and seasonal model and AR(1) model for residuals

We developed regression models with the quadratic trend and seasonality, which is presented below. The regression model with quadratic trend and seasonality contains 13 independent variables: trend index (t), squared trend index (t2), and 11 seasonal dummy variables for season 2 to season 12.

```
> summary(train.trend.season)

Call:
tslm(formula = candytrain.ts ~ trend + I(trend^2) + season)

Residuals:
     Min       1Q   Median       3Q      Max
-171.685  -27.380    3.298   27.796  104.355

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.199e+02  9.234e+00  56.300  < 2e-16 ***
trend        9.788e-01  6.649e-02  14.720  < 2e-16 ***
I(trend^2)  -6.252e-04  1.487e-04  -4.204 3.21e-05 ***
season2     -3.177e+01  1.013e+01  -3.136  0.00184 **
season3     -8.128e+01  1.013e+01  -8.021 1.06e-14 ***
season4     -1.087e+02  1.013e+01 -10.725  < 2e-16 ***
season5     -1.024e+02  1.013e+01 -10.102  < 2e-16 ***
season6     -8.787e+01  1.013e+01  -8.671  < 2e-16 ***
season7     -1.004e+02  1.013e+01  -9.903  < 2e-16 ***
season8     -6.216e+01  1.013e+01  -6.134 1.99e-09 ***
season9     -3.039e+01  1.013e+01  -2.998  0.00288 **
season10     9.734e+01  1.013e+01   9.605  < 2e-16 ***
season11     1.150e+02  1.013e+01  11.346  < 2e-16 ***
season12     1.116e+02  1.014e+01  11.012  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42.99 on 418 degrees of freedom
Multiple R-squared:  0.8927,    Adjusted R-squared:  0.8894
F-statistic: 267.6 on 13 and 418 DF,  p-value: < 2.2e-16
```

Regression equation :

yt = 519.9 + 0.9788 t – 0.0006252 t2 – 31.77 D2 – 81.28 D3 -108.7 D4 -102.4 D5 – 87.87 D6 – 100.4 D7 - 62.16 D8 – 30.39 D9 + 97.34 D10 + 115 D11 + 111.6 D12

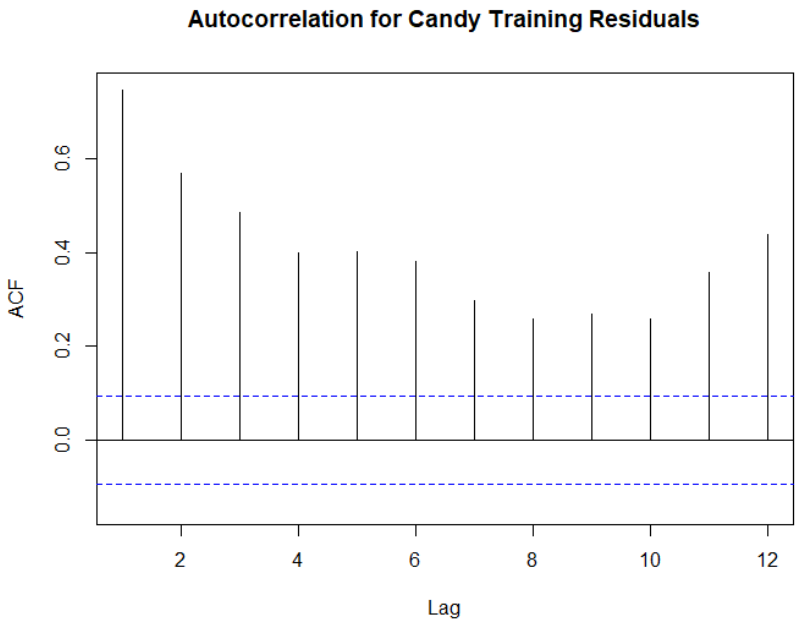
The model's summary shows a high R-squared of 0.8927, statistically significant F-statistic (p-value is substantially lower than 0.01), this model can explain 89.27% of the variance in time series data, which is good to fit the data with the quadratic trend and seasonal pattern. Furthermore, all regression coefficients and intercept are statistically significant (p-value <0.01). We conclude that this regression model may be applied for forecasting.

The production forecast of validation period (year 2008~2016) using this regression model based on training period is presented below:

```
> train.trend.season.pred
         Point Forecast     Lo 0     Hi 0
Jan 2008       826.4860 826.4860 826.4860
Feb 2008       795.1485 795.1485 795.1485
Mar 2008       746.0738 746.0738 746.0738
Apr 2008       719.1145 719.1145 719.1145
May 2008       725.8591 725.8591 725.8591
Jun 2008       740.7847 740.7847 740.7847
Jul 2008       728.7275 728.7275 728.7275
Aug 2008       767.3550 767.3550 767.3550
Sep 2008       799.5574 799.5574 799.5574
Oct 2008       927.7128 927.7128 927.7128
Nov 2008       945.7902 945.7902 945.7902
Dec 2008       942.8326 942.8326 942.8326
Jan 2009       831.6445 831.6445 831.6445
Feb 2009       800.2921 800.2921 800.2921
Mar 2009       751.2024 751.2024 751.2024
Apr 2009       724.2281 724.2281 724.2281
May 2009       730.9576 730.9576 730.9576
Jun 2009       745.8682 745.8682 745.8682
Jul 2009       733.7960 733.7960 733.7960
Aug 2009       772.4085 772.4085 772.4085
Sep 2009       804.5959 804.5959 804.5959
Oct 2009       932.7363 932.7363 932.7363
Nov 2009       950.7987 950.7987 950.7987
Dec 2009       947.8260 947.8260 947.8260
Jan 2010       836.6230 836.6230 836.6230
Feb 2010       805.2556 805.2556 805.2556
Mar 2010       756.1508 756.1508 756.1508
Apr 2010       729.1615 729.1615 729.1615
May 2010       735.8761 735.8761 735.8761
Jun 2010       750.7717 750.7717 750.7717
Jul 2010       738.6845 738.6845 738.6845
Aug 2010       777.2820 777.2820 777.2820
Sep 2010       809.4544 809.4544 809.4544
Oct 2010       937.5798 937.5798 937.5798
Nov 2010       955.6271 955.6271 955.6271
Dec 2010       952.6395 952.6395 952.6395
Jan 2011       841.4214 841.4214 841.4214
Feb 2011       810.0390 810.0390 810.0390
Mar 2011       760.9193 760.9193 760.9193
Apr 2011       733.9150 733.9150 733.9150
May 2011       740.6145 740.6145 740.6145
Jun 2011       755.4951 755.4951 755.4951
Jul 2011       743.3929 743.3929 743.3929
Aug 2011       781.9754 781.9754 781.9754
Sep 2011       814.1328 814.1328 814.1328
Oct 2011       942.2431 942.2431 942.2431
Nov 2011       960.2755 960.2755 960.2755
Dec 2011       957.2728 957.2728 957.2728
Jun 2012       760.0385 760.0385 760.0385
Jul 2012       747.9212 747.9212 747.9212
Aug 2012       786.4887 786.4887 786.4887
Sep 2012       818.6311 818.6311 818.6311
Oct 2012       946.7265 946.7265 946.7265
Nov 2012       964.7438 964.7438 964.7438
Dec 2012       961.7262 961.7262 961.7262
Jan 2013       850.4781 850.4781 850.4781
Feb 2013       819.0657 819.0657 819.0657
Mar 2013       769.9159 769.9159 769.9159
Apr 2013       742.8816 742.8816 742.8816
May 2013       749.5511 749.5511 749.5511
Jun 2013       764.4017 764.4017 764.4017
Jul 2013       752.2695 752.2695 752.2695
Aug 2013       790.8220 790.8220 790.8220
Sep 2013       822.9494 822.9494 822.9494
Oct 2013       951.0297 951.0297 951.0297
Nov 2013       969.0321 969.0321 969.0321
Dec 2013       965.9994 965.9994 965.9994
Jan 2014       854.7364 854.7364 854.7364
Feb 2014       823.3089 823.3089 823.3089
Mar 2014       774.1442 774.1442 774.1442
Apr 2014       747.0948 747.0948 747.0948
May 2014       753.7494 753.7494 753.7494
Jun 2014       768.5850 768.5850 768.5850
Jul 2014       756.4377 756.4377 756.4377
Aug 2014       794.9752 794.9752 794.9752
Sep 2014       827.0876 827.0876 827.0876
Oct 2014       955.1530 955.1530 955.1530
Nov 2014       973.1403 973.1403 973.1403
Dec 2014       970.0926 970.0926 970.0926
Jan 2015       858.8146 858.8146 858.8146
Feb 2015       827.3721 827.3721 827.3721
Mar 2015       778.1924 778.1924 778.1924
Apr 2015       751.1280 751.1280 751.1280
May 2015       757.7676 757.7676 757.7676
Jun 2015       772.5881 772.5881 772.5881
Jul 2015       760.4259 760.4259 760.4259
Aug 2015       798.9484 798.9484 798.9484
Sep 2015       831.0458 831.0458 831.0458
Oct 2015       959.0961 959.0961 959.0961
Nov 2015       977.0684 977.0684 977.0684
Dec 2015       974.0058 974.0058 974.0058
Jan 2016       862.7127 862.7127 862.7127
Feb 2016       831.2552 831.2552 831.2552
Mar 2016       782.0605 782.0605 782.0605
Apr 2016       754.9812 754.9812 754.9812
May 2016       761.6057 761.6057 761.6057
Jun 2016       776.4113 776.4113 776.4113
Jul 2016       764.2340 764.2340 764.2340
Aug 2016       802.7415 802.7415 802.7415
Sep 2016       834.8239 834.8239 834.8239
Oct 2016       962.8592 962.8592 962.8592
Nov 2016       980.8165 980.8165 980.8165
Dec 2016       977.7389 977.7389 977.7389
```

Further, we apply Acf() function to identify the autocorrelation of regression residual to test this model whether it fits well with AR model for residuals. Looking at the regression model's residuals autocorrelation result below, we can find out that there has a very clear relationship. Typically, for residuals in lag 1 and in several lags gradually drops to closer to zero, which means that autocorrelations between residuals are not fully incorporated

into the regression model, specifically in lag 1. This residuals model is considered as autocorrelated then this means that there are systematic movements in your time series which the forecast of regression model will be failed to capture. Thus, we will model this residual autocorrelation with an AR model and developing a two-level model, may overall improve the forecast.



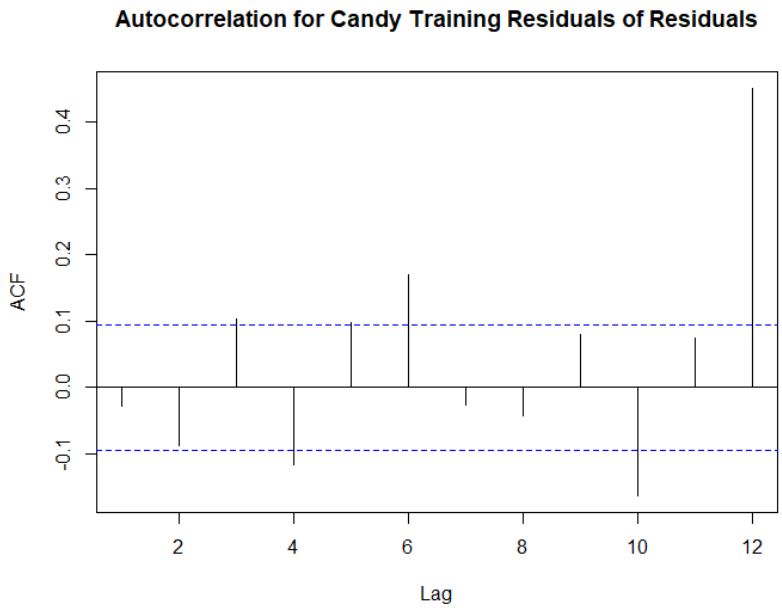
We developed regression' residuals of residuals of training data after AR(1) model fitted by Arima() function The Arima model of with order = c(1,0,0) with order 1, no differencing, and no moving average model. The summary result below shows that the correlation coefficient for the autoregressive model = 0.7651, and intercept is -0.5783.

The *AR(1)* model's equation is:

e_t = -0.5783+ 0.7651 e_{t-1}

```
> summary(res.ar1)
Series: train.trend.season$residuals
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1     mean
      0.7651  -0.5783
s.e.  0.0317   5.6166

sigma^2 estimated as 766.7:  log likelihood=-2047.12
AIC=4100.24   AICc=4100.29   BIC=4112.44

Training set error measures:
                     ME     RMSE      MAE      MPE     MAPE      MASE         ACF1
Training set -0.1367561 27.62561 21.45567 84.57916 188.3313 0.6776573 -0.02863605
```

To deeper analyze the result, we produced the Acf() autocorrelation function and autocorrelation chart for the AR(1) model's residuals (residuals of residuals) is presented below.



As can be seen from the chart, the AR(1) model for residuals absorbed autocorrelation relations between residuals in lags 1 and 11, but there is still autocorrelation left in lag 12, 10, 6,4 and 3. Typically, when autocorrelation at lag-1 exist and high, it is sufficient to fit and AR (1) model. Despite the latter point, overall, the AR(1) model for residuals may be combined with the original regression model to improve the time series forecast.

The table below describes U.S candy production in the validation period of 108 months from 2008 to 2016, regression model's forecast in the validation period (Reg.Forecast), AR(1) model's forecast of the regression residuals in the validation period (AR(1)Forecast), and combined forecast (Combined.Forecast) as a sum of the regression and AR(1) models' forecasts which is the two-level modeling results, regression + AR(1) for validation period.

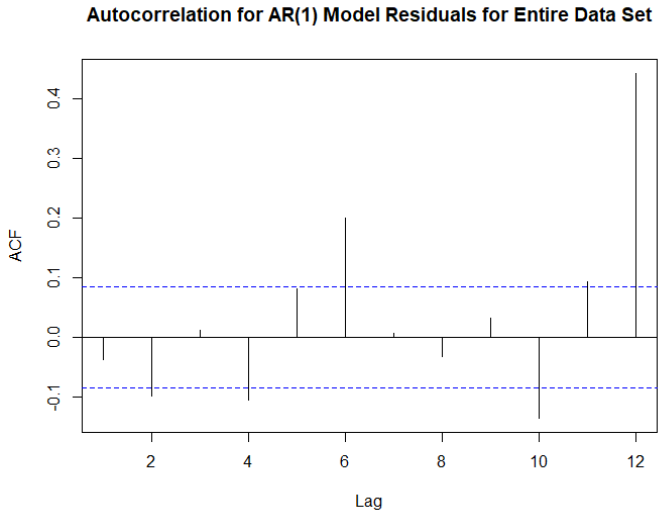
```
> valid.df
    Valid.Sales Reg.Forecast AR(1)Forecast Combined.Forecast
1      727.8765     826.4860   -99.3635777          727.1224
2      681.2608     795.1485   -76.1572198          718.9913
3      650.6321     746.0738   -58.4024321          687.6714
4      614.8743     719.1145   -44.8185475          674.2960
5      595.2073     725.8591   -34.4257496          691.4333
6      595.7629     740.7847   -26.4743976          714.3103
7      569.7271     728.7275   -20.3909541          708.3366
8      592.7763     767.3550   -15.7366155          751.6184
9      691.2446     799.5574   -12.1756605          787.3818
10     763.4422     927.7128    -9.4512348          918.2616
11     775.5853     945.7902    -7.3668232          938.4233
12     681.1618     942.8326    -5.7720756          937.0605
13     601.7333     831.6445    -4.5519614          827.0925
14     595.5969     800.2921    -3.6184729          796.6736
15     572.6836     751.2024    -2.9042768          748.2981
16     533.5269     724.2281    -2.3578576          721.8702
17     537.1500     730.9576    -1.9398016          729.0178
18     532.5557     745.8682    -1.6199541          744.2483
19     550.9456     733.7960    -1.3752442          732.4208
20     596.0373     772.4085    -1.1880209          771.2205
21     677.0434     804.5959    -1.0447795          803.5512
22     827.7832     932.7363    -0.9351880          931.8011
23     783.6012     950.7987    -0.8513414          949.9473
24     780.0645     947.8260    -0.7871918          947.0388
25     671.8748     836.6230    -0.7381120          835.8849
26     662.7437     805.2556    -0.7005619          804.5550
27     615.5550     756.1508    -0.6718330          755.4790
28     598.9957     729.1615    -0.6498530          728.5117
29     602.2079     735.8761    -0.6330364          735.2430
30     599.0848     750.7717    -0.6201704          750.1515
31     645.6643     738.6845    -0.6103268          738.0742
32     714.6398     777.2820    -0.6027957          776.6792
33     775.4507     809.4544    -0.5970337          808.8573
34     845.2160     937.5798    -0.5926253          936.9871
35     787.9358     955.6271    -0.5892525          955.0378
36     794.8460     952.6395    -0.5866721          952.0528
37     689.8383     841.4214    -0.5846978          840.8367
38     686.4334     810.0390    -0.5831873          809.4558
39     662.5289     760.9193    -0.5820317          760.3372
40     652.7834     733.9150    -0.5811476          733.3338
41     611.5223     740.6145    -0.5804711          740.0340
42     600.3244     755.4951    -0.5799536          754.9152
43     599.9020     743.3929    -0.5795576          742.8133
44     615.0497     781.9754    -0.5792546          781.3961
45     628.5455     814.1328    -0.5790229          813.5537
46     781.5363     942.2431    -0.5788455          941.6643
47     781.9540     960.2755    -0.5787099          959.6968
48     769.4408     957.2728    -0.5786061          956.6942
49     669.1071     846.0398    -0.5785267          845.4613
50     662.9191     814.6423    -0.5784659          814.0639
51     630.1666     765.5076    -0.5784194          764.9292
52     586.9718     738.4883    -0.5783838          737.9099
53     571.2827     745.1728    -0.5783566          744.5945
54     579.5188     760.0385    -0.5783358          759.4601
55     599.1985     747.9212    -0.5783199          747.3429
56     657.9325     786.4887    -0.5783077          785.9104
57     751.4565     818.6311    -0.5782984          818.0528
58     769.1249     946.7265    -0.5782912          946.1482
59     776.9561     964.7438    -0.5782858          964.1655
60     777.3644     961.7262    -0.5782816          961.1479
61     716.6773     850.4781    -0.5782784          849.8998
62     682.8960     819.0657    -0.5782760          818.4874
63     686.9495     769.9159    -0.5782741          769.3377
64     638.0125     742.8816    -0.5782727          742.3033
65     614.1695     749.5511    -0.5782716          748.9729
66     614.5316     764.4017    -0.5782707          763.8235
67     618.6641     752.2695    -0.5782701          751.6912
68     671.7382     790.8220    -0.5782696          790.2437
69     706.2584     822.9494    -0.5782692          822.3711
70     785.4345     951.0297    -0.5782689          950.4515
71     814.0291     969.0321    -0.5782687          968.4538
72     824.8944     965.9994    -0.5782686          965.4212
73     699.8984     854.7364    -0.5782684          854.1581
74     695.7780     823.3089    -0.5782683          822.7306
75     676.5005     774.1442    -0.5782683          773.5659
76     622.5094     747.0948    -0.5782682          746.5166
77     591.7395     753.7494    -0.5782682          753.1711
78     598.1571     768.5850    -0.5782681          768.0067
79     589.6485     756.4377    -0.5782681          755.8595
80     656.4814     794.9752    -0.5782681          794.3969
81     715.2998     827.0876    -0.5782681          826.5093
82     801.6371     955.1530    -0.5782680          954.5747
83     863.8543     973.1403    -0.5782680          972.5620
84     860.4467     970.0926    -0.5782680          969.5144
85     735.9487     858.8146    -0.5782680          858.2363
86     728.9528     827.3721    -0.5782680          826.7938
87     713.0147     778.1924    -0.5782680          777.6141
88     676.4490     751.1280    -0.5782680          750.5498
89     647.7821     757.7676    -0.5782680          757.1893
90     674.9149     772.5881    -0.5782680          772.0099
91     688.2687     760.4259    -0.5782680          759.8476
92     775.7774     798.9484    -0.5782680          798.3701
93     775.7332     831.0458    -0.5782680          830.4675
94     848.3398     959.0961    -0.5782680          958.5178
95     833.4378     977.0684    -0.5782680          976.4902
96     804.7890     974.0058    -0.5782680          973.4275
97     726.2541     862.7127    -0.5782680          862.1344
98     723.7555     831.2552    -0.5782680          830.6770
99     722.4897     782.0605    -0.5782680          781.4822
100    693.5491     754.9812    -0.5782680          754.4029
101    683.2661     761.6057    -0.5782680          761.0274
102    685.4267     776.4113    -0.5782680          775.8330
103    689.0867     764.2340    -0.5782680          763.6558
104    700.7766     802.7415    -0.5782680          802.1632
105    731.7091     834.8239    -0.5782680          834.2456
106    796.8426     962.8592    -0.5782680          962.2809
107    782.0698     980.8165    -0.5782680          980.2383
108    779.4621     977.7389    -0.5782680          977.1606
```

```
> round(accuracy(valid.two.level.pred, candyvalid.ts), 3)
                ME    RMSE     MAE     MPE   MAPE  ACF1 Theil's U
Test set -131.47 139.338 131.483 -19.562 19.564 0.703     2.902
> round(accuracy(train.trend.season.pred$mean, candyvalid.ts), 3)
                 ME     RMSE     MAE     MPE    MAPE ACF1 Theil's U
Test set -135.941 142.456 135.941 -20.241 20.241  0.7     2.963
```

The accuracy above shows that the two-level model has substantially better MAPE and RMSE than the regression model with quadratic trend and seasonality. Further, we re-run the regression model with AR(1) residuals based on the entire production data (candy.ts), and the results below. We will compare this model with other 4 models to evaluate accuracy in the following steps.

```
> summary(trend.season)

Call:
tslm(formula = Candy.ts ~ trend + I(trend^2) + season)

Residuals:
     Min       1Q   Median       3Q      Max
-179.357  -33.491   -0.419   37.739  119.703

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.909e+02  9.942e+00  49.383  < 2e-16 ***
trend        1.422e+00  5.722e-02  24.858  < 2e-16 ***
I(trend^2)  -1.885e-03  1.024e-04 -18.404  < 2e-16 ***
season2     -2.791e+01  1.091e+01  -2.559   0.0108 *
season3     -7.158e+01  1.091e+01  -6.564 1.26e-10 ***
season4     -1.003e+02  1.091e+01  -9.196  < 2e-16 ***
season5     -9.872e+01  1.091e+01  -9.052  < 2e-16 ***
season6     -8.639e+01  1.091e+01  -7.921 1.41e-14 ***
season7     -9.464e+01  1.091e+01  -8.677  < 2e-16 ***
season8     -5.436e+01  1.091e+01  -4.984 8.47e-07 ***
season9     -1.829e+01  1.091e+01  -1.677   0.0942 .
season10     1.011e+02  1.091e+01   9.270  < 2e-16 ***
season11     1.150e+02  1.091e+01  10.540  < 2e-16 ***
season12     1.096e+02  1.091e+01  10.049  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 51.73 on 526 degrees of freedom
Multiple R-squared:  0.8232,    Adjusted R-squared:  0.8189
F-statistic: 188.5 on 13 and 526 DF,  p-value: < 2.2e-16
```

```
> summary(residual.ar1)
Series: trend.season$residuals
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1    mean
      0.8317  0.3746
s.e.  0.0239  7.2007

sigma^2 estimated as 810.2:  log likelihood=-2574.07
AIC=5154.14   AICc=5154.19   BIC=5167.02

Training set error measures:
                     ME     RMSE      MAE      MPE     MAPE      MASE         ACF1
Training set -0.1911171 28.41083 21.93024 8.995106 167.2934 0.6369708 -0.03758105
```



3. Auto ARIMA Model

Use auto.arima() function to fit ARIMA model for the training data.

The output from using the *auto.arima()* function for candytrain.ts is presented below:

```
> summary(train.auto.arima)
Series: candytrain.ts
ARIMA(4,0,0)(2,1,2)[12] with drift

Coefficients:
         ar1     ar2     ar3      ar4     sar1    sar2     sma1     sma2   drift
      0.6477  0.0806  0.2226  -0.1114  -0.5340  0.0397  -0.0367  -0.5031  0.5586
s.e.  0.0491  0.0575  0.0580   0.0496   0.2618  0.0956   0.2560   0.1994  0.1973

sigma^2 estimated as 567.8:  log likelihood=-1927.29
AIC=3874.59   AICc=3875.13   BIC=3914.99

Training set error measures:
                     ME     RMSE      MAE          MPE     MAPE      MASE        ACF1
Training set -0.03613415 23.24271 17.7641 -0.07624424 2.737507 0.5529084 0.01124804
```

This is a seasonal ARIMA model. The first three parameters of the model describe an *AR* component with order 4 (p=4) for trend, no differencing (d=0) and no moving average for trend. The next three parameters describe AR seasonal component with order 2 (P=2), first differencing (D=1) and MA with order *2* (Q=2) components. The model is also done for monthly seasonality (number 12 in the ARIMA model description). The drift parameter is an "intercept" of this model. The ARIMA model's equation is:

$y_t - y_{t-1} = 0.5586 + 0.6477\ y_{t-1} + 0.0806y_{t-2} + 0.2226y_{t-3} - 0.1114y_{t-4} - 0.5340(y_{t-1}-y_{t-12}) - 0.0367(y_{t-2}-y_{t-13}) - 0.0367\rho_{t-1} - 0.5031\ \rho_{t-2}$

The ARIMA model's forecast in the first 24 periods of the validation period is presented below:

```
> train.auto.arima.pred
         Point Forecast     Lo 0     Hi 0
Jan 2008       738.2134 738.2134 738.2134
Feb 2008       730.7320 730.7320 730.7320
Mar 2008       695.0724 695.0724 695.0724
Apr 2008       655.7679 655.7679 655.7679
May 2008       664.3280 664.3280 664.3280
Jun 2008       663.0049 663.0049 663.0049
Jul 2008       654.7323 654.7323 654.7323
Aug 2008       715.2174 715.2174 715.2174
Sep 2008       764.7135 764.7135 764.7135
Oct 2008       831.0187 831.0187 831.0187
Nov 2008       848.9319 848.9319 848.9319
Dec 2008       860.3913 860.3913 860.3913
Jan 2009       775.6357 775.6357 775.6357
Feb 2009       769.0173 769.0173 769.0173
Mar 2009       723.6153 723.6153 723.6153
Apr 2009       693.8735 693.8735 693.8735
May 2009       696.0051 696.0051 696.0051
Jun 2009       697.4408 697.4408 697.4408
Jul 2009       690.3201 690.3201 690.3201
Aug 2009       750.9777 750.9777 750.9777
Sep 2009       795.3217 795.3217 795.3217
Oct 2009       863.3218 863.3218 863.3218
Nov 2009       878.2995 878.2995 878.2995
Dec 2009       884.9936 884.9936 884.9936
```

Use auto.arima() function to fit ARIMA model for the entire data.

The output from using the *auto.arima()* function for candy.ts is presented below:

```
> auto.arima <- auto.arima(Candy.ts)
> summary(auto.arima)
Series: Candy.ts
ARIMA(1,0,4)(0,1,2)[12]

Coefficients:
         ar1      ma1      ma2     ma3      ma4     sma1     sma2
      0.9555  -0.2563  -0.1199  0.0422  -0.1098  -0.6436  -0.0974
s.e.  0.0189   0.0474   0.0469  0.0470   0.0437   0.0459   0.0438

sigma^2 estimated as 615.3:  log likelihood=-2445.75
AIC=4907.49   AICc=4907.77   BIC=4941.64

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE          ACF1
Training set 1.163503 24.36452 18.40181 0.1130713 2.789418 0.5297023 -0.0008543002
```

This is a seasonal ARIMA model. The first three parameters of the model describe an AR component with order 1 (p=1) for trend, no differencing (d=0) and order 4 moving average (q=4) components for trend. The next three parameters describe no AR seasonal component (P=0), first differencing (D=1) and MA with order 2 (Q=2) components. The model is also done for monthly seasonality (number 12 in the ARIMA model description).

The ARIMA model's equation is:

$$y_t - y_{t-1} = 0.9555\, y_{t-1} - 0.2563\varepsilon_{t-1} - 0.1199\varepsilon_{t-2} + 0.0422\varepsilon_{t-3} - 0.1098\varepsilon_{t-4} - 0.6436\rho_{t-1} - 0.0974\, \rho_{t-2}$$

The ARIMA's model forecast in 24 periods is presented below:

```
> HW.ZZZ.pred <- forecast(HW.ZZZ, h = 24 , level = 0)
> HW.ZZZ.pred
         Point Forecast     Lo 0     Hi 0
Jan 2017       680.5494 680.5494 680.5494
Feb 2017       669.8835 669.8835 669.8835
Mar 2017       652.8397 652.8397 652.8397
Apr 2017       610.0296 610.0296 610.0296
May 2017       587.7277 587.7277 587.7277
Jun 2017       595.7319 595.7319 595.7319
Jul 2017       602.5135 602.5135 602.5135
Aug 2017       653.2016 653.2016 653.2016
Sep 2017       694.7166 694.7166 694.7166
Oct 2017       777.5155 777.5155 777.5155
Nov 2017       786.2865 786.2865 786.2865
Dec 2017       780.1746 780.1746 780.1746
Jan 2018       680.5494 680.5494 680.5494
Feb 2018       669.8835 669.8835 669.8835
Mar 2018       652.8397 652.8397 652.8397
Apr 2018       610.0296 610.0296 610.0296
May 2018       587.7277 587.7277 587.7277
Jun 2018       595.7319 595.7319 595.7319
Jul 2018       602.5135 602.5135 602.5135
Aug 2018       653.2016 653.2016 653.2016
Sep 2018       694.7166 694.7166 694.7166
Oct 2018       777.5155 777.5155 777.5155
Nov 2018       786.2865 786.2865 786.2865
Dec 2018       780.1746 780.1746 780.1746
```

Evaluate & Compare Performance

```
> # (1) regression model with quadratic trend and seasonality
> # (2) Two-level model, regression  + AR(1)
> # (3) Seasonal naive forecas
> # (4) HW model, (z,z,z)
> # (5) Auto ARIMA model
> round(accuracy(trend.season$fitted, Candy.ts), 3)
         ME   RMSE    MAE    MPE  MAPE  ACF1 Theil's U
Test set  0 51.056 41.676 -0.646 6.514 0.828     0.865
> round(accuracy(trend.season$fitted + residual.ar1$fitted, Candy.ts), 3)
             ME   RMSE   MAE    MPE  MAPE   ACF1 Theil's U
Test set -0.191 28.411 21.93 -0.241 3.356 -0.038     0.477
> round(accuracy((snaive(Candy.ts))$fitted, Candy.ts), 3)
            ME  RMSE   MAE   MPE  MAPE ACF1 Theil's U
Test set 4.409 45.58 34.74 0.457 5.438 0.76     0.785
> round(accuracy(HW.ZZZ.pred$fitted, Candy.ts), 3)
            ME   RMSE   MAE    MPE MAPE  ACF1 Theil's U
Test set 0.409 26.223 20.12 -0.014 3.09 0.142     0.437
> round(accuracy(auto.arima.pred$fitted, Candy.ts), 3)
            ME   RMSE    MAE   MPE  MAPE   ACF1 Theil's U
Test set 1.164 24.365 18.402 0.113 2.789 -0.001     0.396
```

As above, based on MAPE, the most accurate (best) model for forecasting is the auto-ARIMA model with the lowest MAPE value of 2.79%. Based on RMSE measure, auto ARIMA is also the best way to do the forecast with the lowest RMSE value of 24.365. Furthermore, auto-ARIMA also has the lowest value of MAE of 18.4 which is also very good.

Other than auto-ARIMA, Holt-Winter's model is also a good way to do the forecast. Holt-Winter's model is the second best with the value of RMSE of 26.22 and MAPE of 3.09.

However, assuming the superiority of the MAPE measure for business time series forecasting, we conclude that the best model to forecast production in 2017 and 2018 is the ARIMA (auto-ARIMA) model.

Conclusion

As above, we have tried variety of different methods and combinations, auto-ARIMA is the best forecasting method. “Season” plays an important role in the U.S market, as we can see in the Time Series components “seasonality” and plot of production from 2010 to 2016 in page 11 and 12, the peak usually appears around the third and fourth quarter of every year, and gradually going down after the end of first quarter. There is a huge demand in western countries especially in the U.S due to those important festivals and holidays, such as Halloween, Easter as well as Christmas. Besides, Chinese New Year comes around the February which is in first quarter.

In the time series component “Trend”, there is a downward trend around 2007 to 2009 because of the financial crisis during that periods of time. After that, there is another upward trend showing up.

Inconclusion, as we have mentioned earlier, because of the increasing demand, it would not be too hard to forecast into the future. By far, we only did the forecast for the next 24 periods, but we can definitely do more than that, especially the value of MAPE by using auto-ARIMA is only 2.8%.

Bibliography

US Candy Production by Month

https://www.kaggle.com/rtatman/us-candy-production-by-month