FORECASTING OF POTATO PRICE

TSA Project

Prithvirraj – 24PGAI0065
Satyam Chaurasiya – 24PGAI0049
Santanu Sen – 24PGAI0061

Contents

Introduction

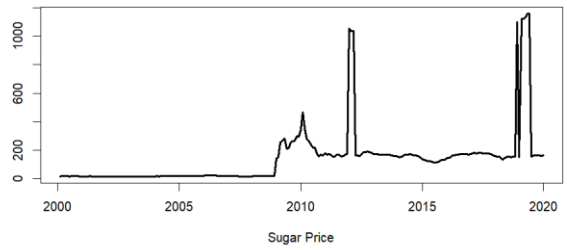
The data for this Time Series Forecasting project was obtained from Kaggle from the following URL:

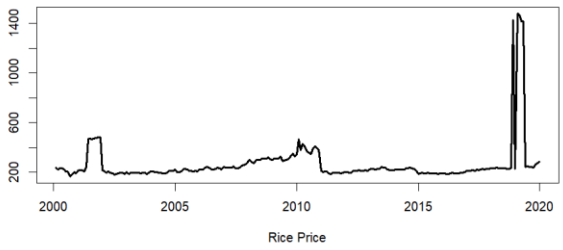
https://www.kaggle.com/datasets/kukuroo3/india-food-price-2000-2020

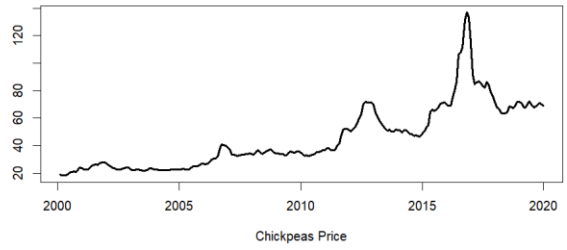
The dataset contains the monthly average prices of various food commodities namely Rice, Wheat, potato, sugarcane, Oil, onion and chickpeas from Feb, 2000 to Jan 2020. On analysing the dataset, it was observed that there were missing data for Oil and Onion. As a result, these two commodities were excluded from forecasting.

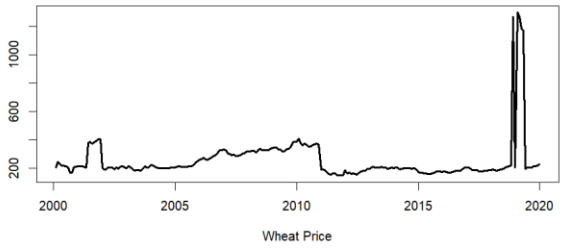
```
> summary(df)
      date         Chickpeas       Oil..mustard.      Potatoes          Rice
 Min.   :2000   Min.   : 18.16   Min.   : 36.10   Min.   : 4.105   Min.   : 166.4
 1st Qu.:2005   1st Qu.: 25.43   1st Qu.: 57.00   1st Qu.: 8.247   1st Qu.: 199.7
 Median :2010   Median : 36.13   Median : 73.07   Median :11.675   Median : 222.7
 Mean   :2010   Mean   : 45.27   Mean   : 77.67   Mean   :12.728   Mean   : 266.0
 3rd Qu.:2015   3rd Qu.: 64.08   3rd Qu.:102.30   3rd Qu.:16.371   3rd Qu.: 242.2
 Max.   :2020   Max.   :136.95   Max.   :114.88   Max.   :29.549   Max.   :1479.6
                                 NA's   :15
     Sugar             Wheat            Onions            year          month
 Min.   :  14.52   Min.   : 149.8   Min.   : 4.803   Min.   :2000   Min.   : 1.00
 1st Qu.:  17.09   1st Qu.: 188.2   1st Qu.: 7.641   1st Qu.:2005   1st Qu.: 3.75
 Median : 137.95   Median : 207.2   Median :12.089   Median :2010   Median : 6.50
 Mean   : 140.87   Mean   : 255.4   Mean   :15.247   Mean   :2010   Mean   : 6.50
 3rd Qu.: 169.38   3rd Qu.: 293.4   3rd Qu.:18.582   3rd Qu.:2015   3rd Qu.: 9.25
 Max.   :1161.59   Max.   :1297.0   Max.   :57.066   Max.   :2020   Max.   :12.00
                                    NA's   :24
```

Data of remaining food commodities were converted to time series and plotted. It was observed that except Potato, all other commodities are having too many outliers/errors in data. A simple time series plot of all commodities except Potaoto is shown below:

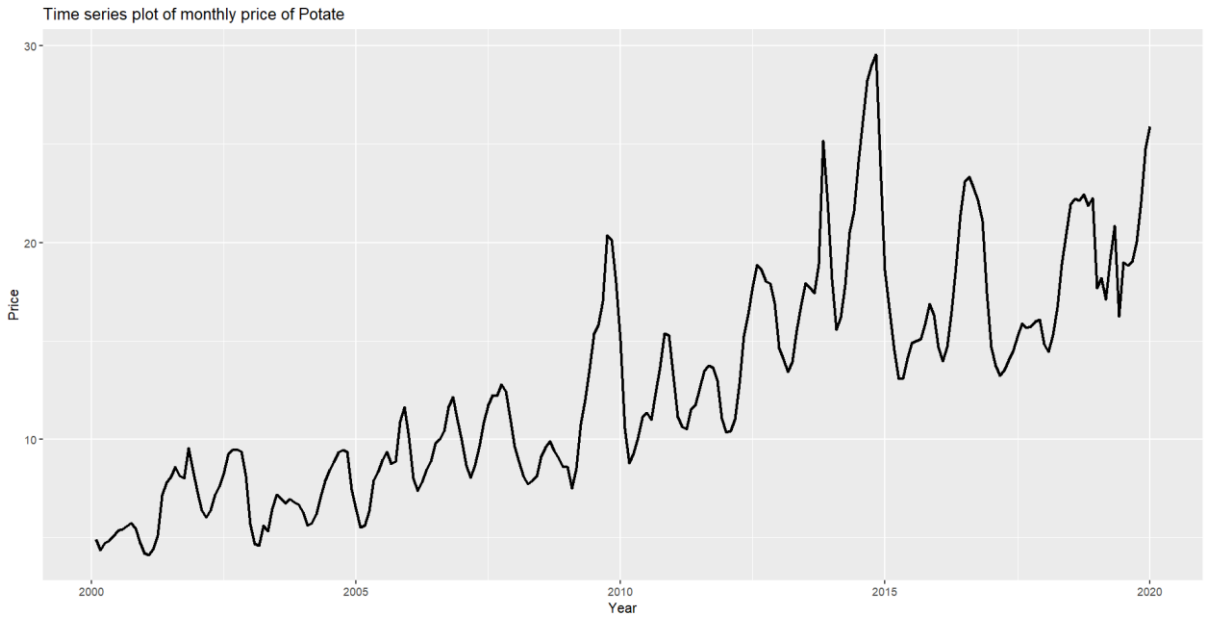








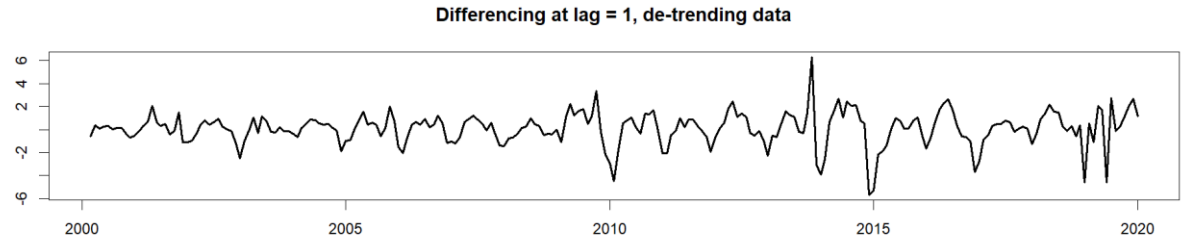
We extracted the potato data from the dataset, and converted it to time series and plotted to observe the trend and seasonality patter of the data. From the time series plot, it can be clearly observed that there is a linear trend in data, and there is clear seasonality.

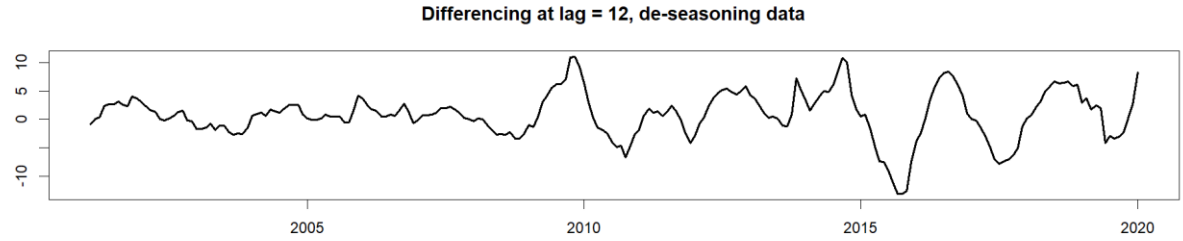


Analysis of Data

De-trending and de-seasoning using differencing

By plotting time series plot of potato prices, it is clear there is trend and seasonality in data. To de-trend, and de-season the data, we took differencing of data at lag = 1 (for de-trending) and lag = 12 (for de-seasoning).

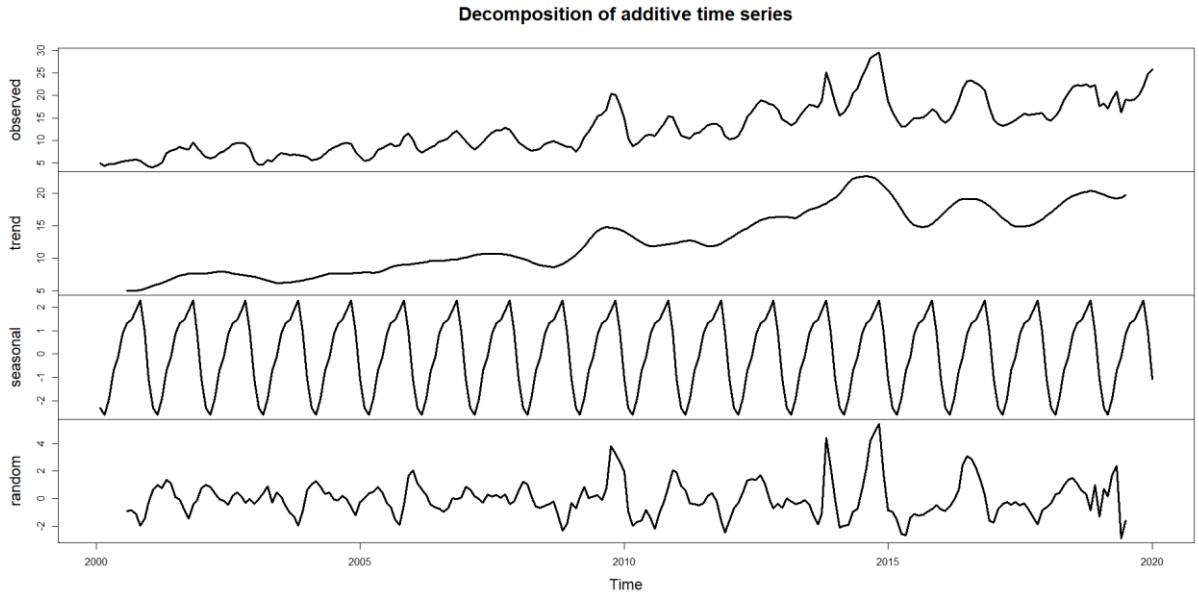




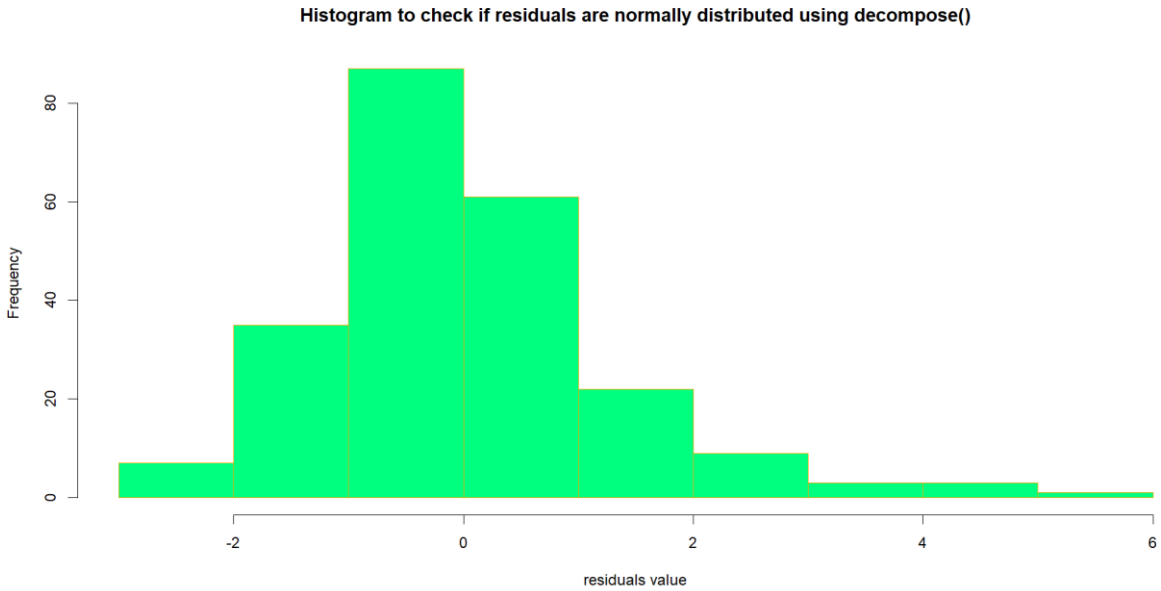
Decomposing data into trend, seasonality, and residual

Two separate method for decomposing data was used viz. decompose() and stl(). Analysis of residual and trend was done in both the scenarios, and it was observed that residual is not normally distributed in both the decomposition.

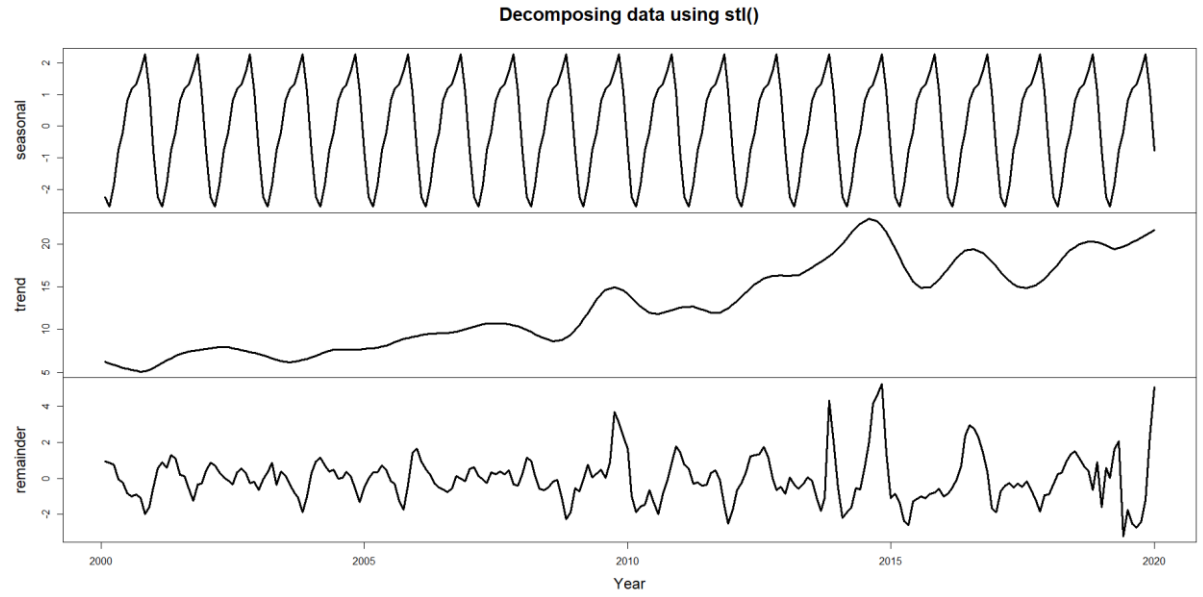
Decomposition using decompose() function



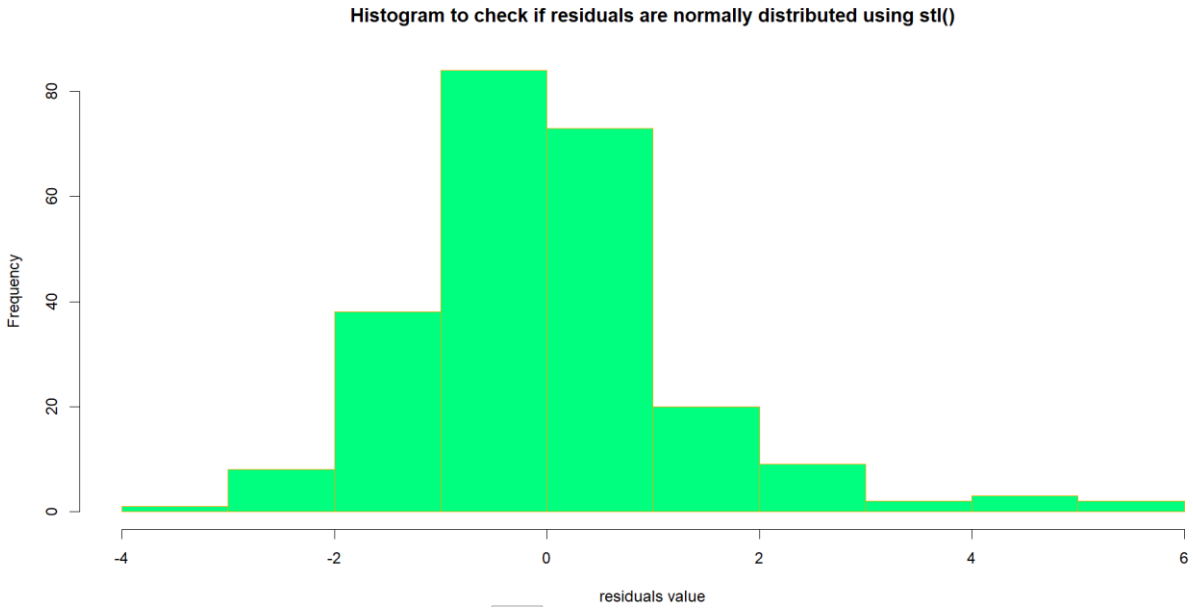
Plotting histogram of residuals



Decompose using stl() function



Plotting histogram of residuals



Analysis of Trend

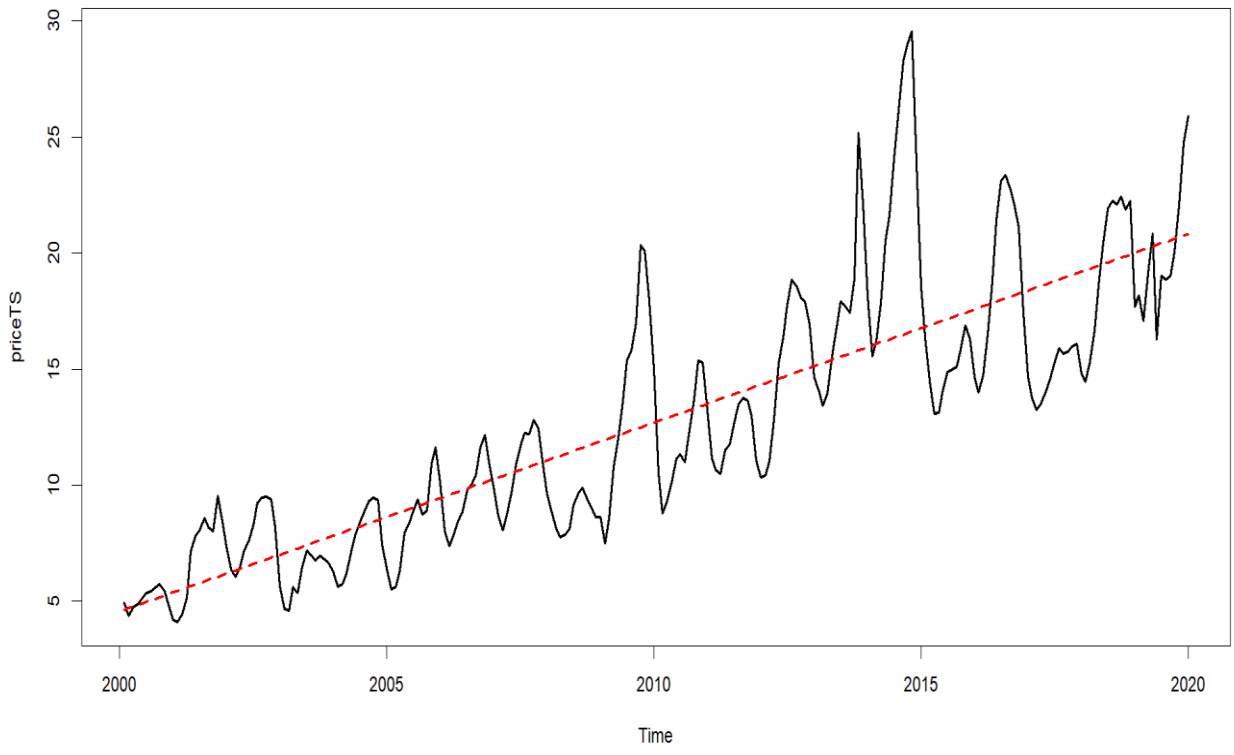
In order to analyse the trend line in time series data, we did following analysis.

1. Fitted a time series linear model with straight trend line – tslm (priceTS ~ trend). Summary of the model fit is shown below. It can be clearly observed that linear trend has very low p-value which compels us to reject the null hypothesis of no trend.

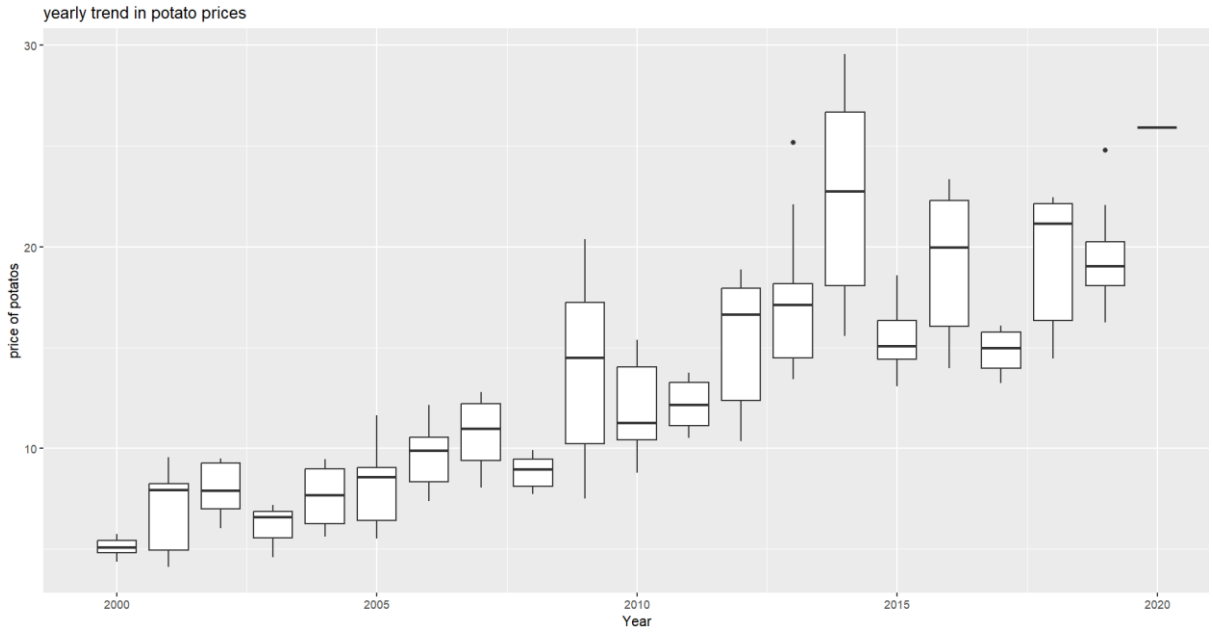
Additionally, we plotted the linear trend line given by this model along with Potato price time series data, and could observe the linear trend aligning with the long term time series data.

```
> summary(priceTS_FitT)

Call:
tslm(formula = priceTS ~ trend)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2787 -2.2073 -0.4601  1.6885 12.9261

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.565483   0.397240   11.49   <2e-16 ***
trend       0.067738   0.002858   23.70   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.067 on 238 degrees of freedom
Multiple R-squared:  0.7024,    Adjusted R-squared:  0.7012
F-statistic: 561.8 on 1 and 238 DF,  p-value: < 2.2e-16
```

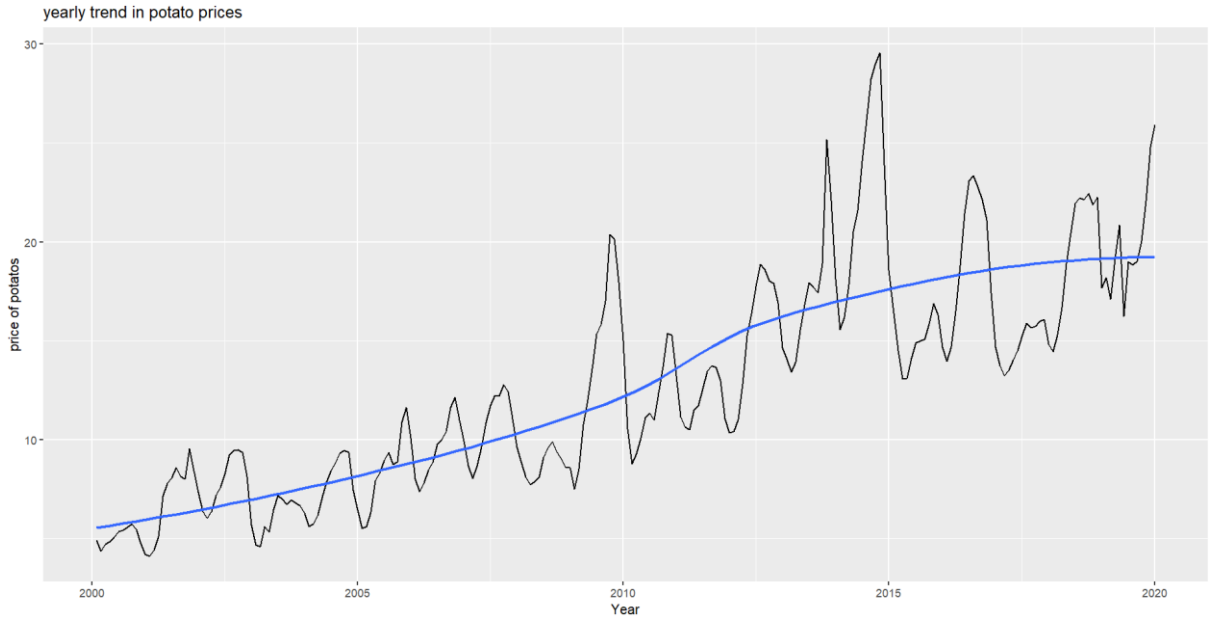
Plotting linear trend along with Potato price time series data



2. Plotting box plot of yearly potato prices, to observe the trend in prices



3. Plotting time series plot of potato prices along with smoothing (local regression) trend line

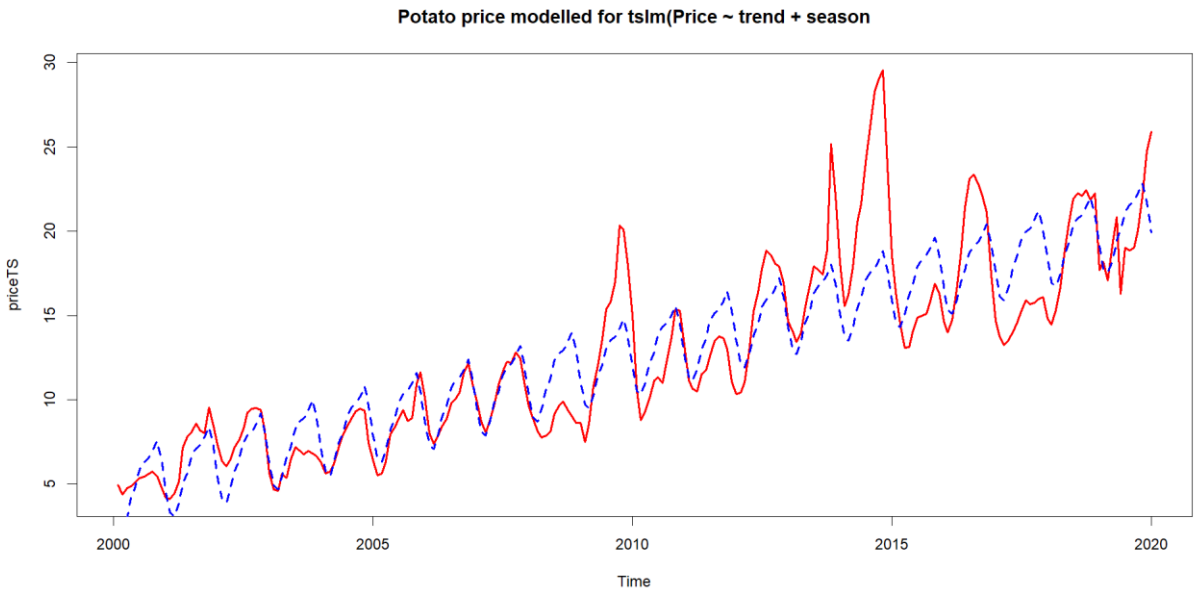


Analysis of Seasonality

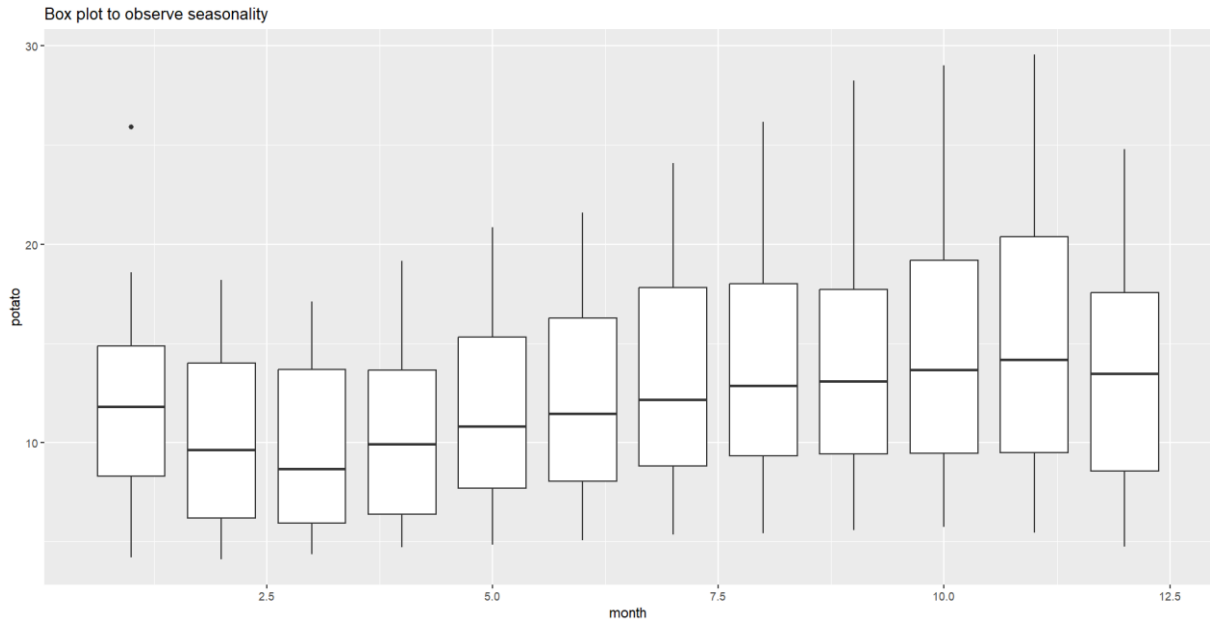
To check if there is any seasonality in the data, we used tslm function to model out data for seasonality. It was observed that there is a strong relationship in price of potatoes and month of they year. It was observed that price of potatoes tend to be higher during sowing month of October and November, and are lower during harvesting month of February and March. Below is the summary of the model fit using the formula tslm(priceTS ~ trend + season).

```
> summary(priceTS_FitTS)

Call:
tslm(formula = priceTS ~ trend + season)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2466 -1.7249 -0.1688  1.3146 10.7758

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.865452   0.683701   5.654 4.69e-08 ***
trend        0.066924   0.002525  26.507  < 2e-16 ***
season2     -1.416961   0.856309  -1.655 0.099361 .
season3     -1.736670   0.856231  -2.028 0.043700 *
season4     -1.051295   0.856160  -1.228 0.220749
season5      0.046915   0.856097   0.055 0.956346
season6      0.623029   0.856041   0.728 0.467484
season7      1.611350   0.855993   1.882 0.061056 .
season8      1.974326   0.855952   2.307 0.021980 *
season9      2.104306   0.855918   2.459 0.014699 *
season10     2.517158   0.855892   2.941 0.003611 **
season11     3.044573   0.855874   3.557 0.000456 ***
season12     1.860973   0.855862   2.174 0.030710 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.706 on 227 degrees of freedom
Multiple R-squared:  0.779,     Adjusted R-squared:  0.7674
F-statistic: 66.69 on 12 and 227 DF,  p-value: < 2.2e-16
```

We plotted the model fit line and potato price data to see how well model explain the price fluctuation over time, and below is our observation:



Additionally, we plotted boxplot of price data across all months from January to December, to observe the seasonality.



Data Modelling

Partitioning data in training and test(validation) set

Out of 20 years of whole data we use 17 years of data as training data and remaining 3 years are used as test data.

Prediction using Random walk

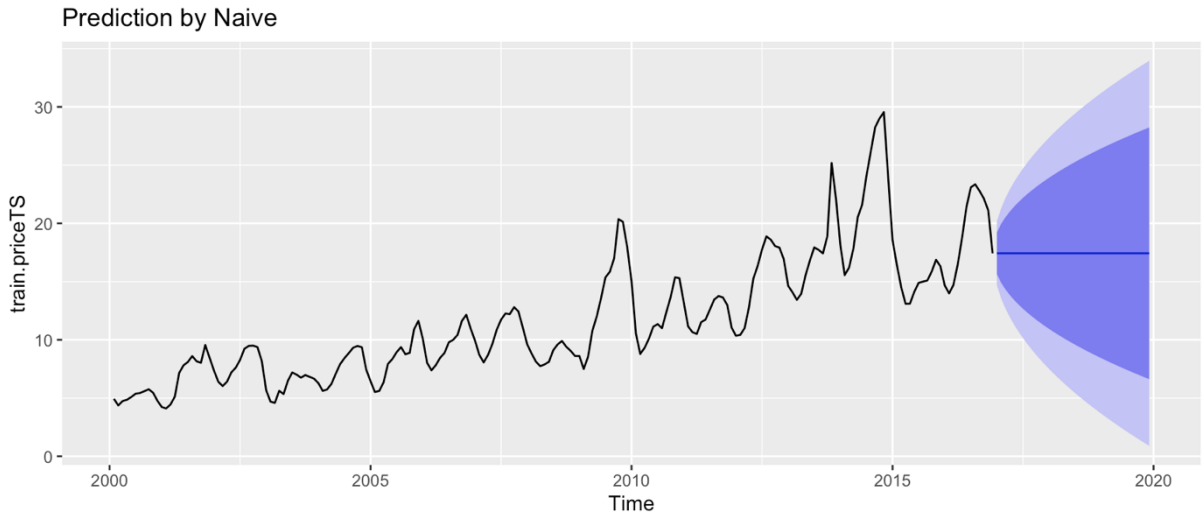
First we use Random walk to predict the model and below are the R_accuracy output for the same.

```
> accuracy_rwf
                    ME     RMSE      MAE          MPE      MAPE      MASE      ACF1 Theil's U
Training set 0.06184966 1.405424 1.000456  0.03397659  8.597305 0.3450148 0.5293222        NA
Test set     0.48596165 3.172219 2.743726 -0.25725561 15.215492 0.9461949 0.7947396  2.133184
```

Prediction by Naive

After Random Walk we use simple Naïve model for prediction. The R_ouptut for accuracy and prediction are as follows:

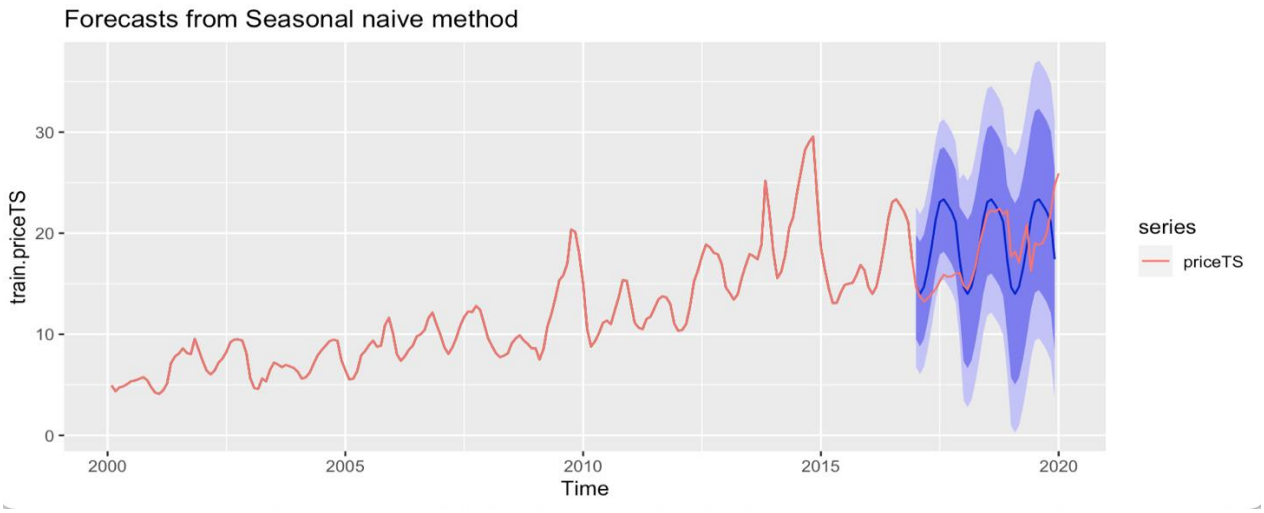
```
> accuracy_Naive
                    ME     RMSE      MAE          MPE      MAPE      MASE      ACF1 Theil's U
Training set 0.06184966 1.405424 1.000456  0.03397659  8.597305 0.3450148 0.5293222        NA
Test set     0.48596165 3.172219 2.743726 -0.25725561 15.215492 0.9461949 0.7947396  2.133184
>
```



Prediction with seasonal naïve

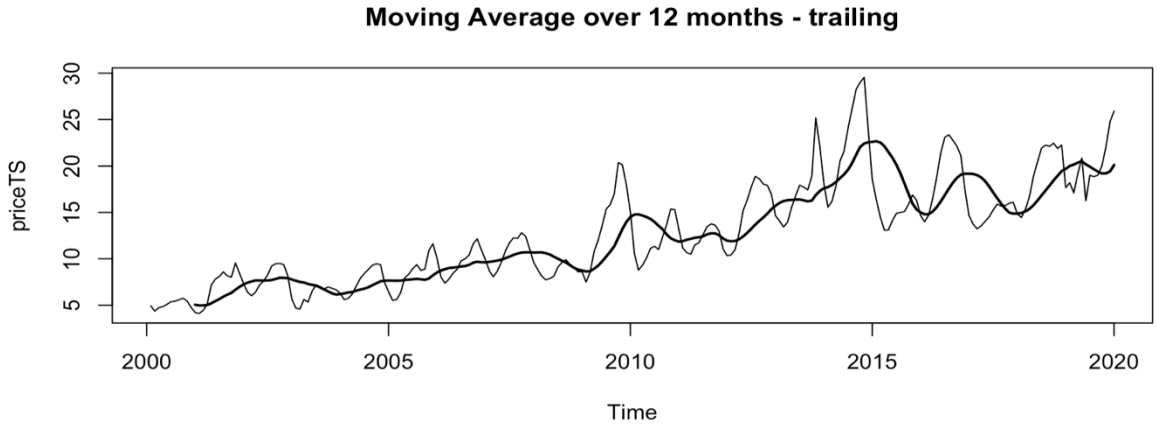
After using simple naïve model, we use seasonal naïve model and below are the output :

```
> accuracy_SNaive
                    ME     RMSE      MAE        MPE     MAPE     MASE      ACF1 Theil's U
Training set  0.8868346 4.042710 2.899747  4.370126 22.18580 1.00000 0.9271897        NA
Test set     -1.2519699 3.824953 2.919495 -8.831986 17.06481 1.00681 0.7435786  2.888436
>
```



Prediction by Moving average model

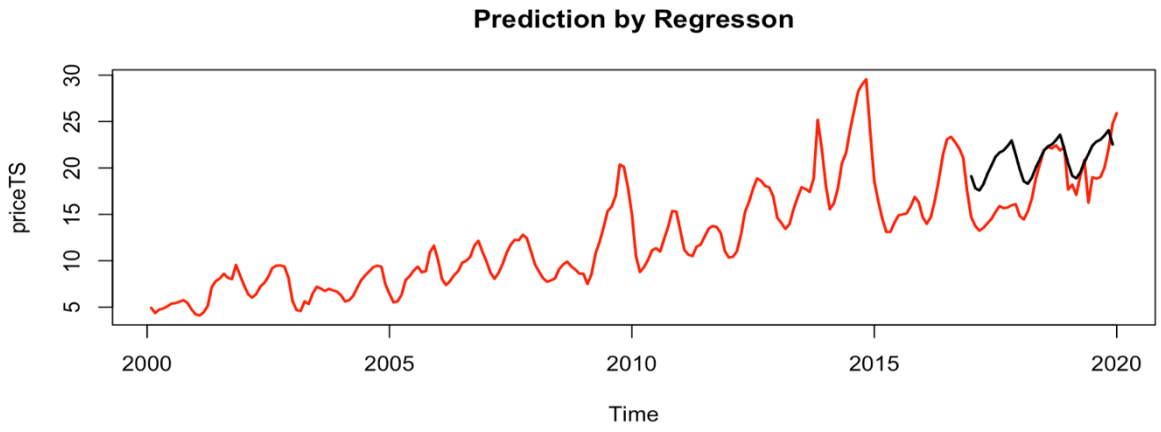
Here we took 12 month moving average because when we were taking moving average of 4 months or 6 months, it's becoming more overfitting type.



Prediction by using regression model

Below is the accuracy which we got from R output by predicting for 36 months on the training set of 17 years.

```
> accuracy_tslm
                     ME     RMSE      MAE        MPE     MAPE      MASE      ACF1 Theil's U
Training set  5.250316e-17 2.425733 1.720775  -2.492205 14.79827 0.5934224 0.9079839        NA
Test set     -3.045458e+00 3.844280 3.195134 -19.339841 19.95958 1.1018662 0.7154159  2.983586
```



Prediction by using exponential smoothing model

In exponential model , we tried and testes so many different MMM, NMN, MNM etc model and finally we choose the one which is suggested by R i.e. MNM model.

Below are accuracy and prediction plot for exponential smoothing model:

```
> accuracy_ets_auto
                     ME      RMSE       MAE       MPE      MAPE      MASE      ACF1 Theil's U
Training set 0.07879728 0.9105354 0.6091538 0.2596652  5.268671 0.2100713 0.3981323        NA
Test set     0.98831037 2.9239325 2.4579487 3.6593633 13.300765 0.8476424 0.7528196  1.964397
```

Below are the output of exponential smoothing.

```
  Smoothing parameters:
    alpha = 0.9856
    gamma = 0.0016

  Initial states:
    l = 5.9446
    s = 0.9001 1.0308 1.1507 1.1187 1.106 1.1102
           1.0904 1.0334 0.9673 0.8761 0.8106 0.8057

  sigma:  0.0758

     AIC     AICc      BIC
1004.472 1007.039 1054.171

Training set error measures:
                     ME      RMSE       MAE       MPE     MAPE      MASE      ACF1
Training set 0.07879728 0.9105354 0.6091538 0.2596652 5.268671 0.2100713 0.3981323
```

Prediction by using ARIMA model

Here in ARIMA Model, we use ARIMA(1,1,2) as p=1, d=1 and q=2 as per the best suggestion by R.

Below is the output of ARIMA Model :

```
ARIMA(1,1,2)

Coefficients:
          ar1     ma1     ma2
      -0.6906  1.3695  0.5956
s.e.   0.1698  0.1441  0.0721

sigma^2 = 1.311:  log likelihood = -312.79
AIC=633.57   AICc=633.78   BIC=646.81

Training set error measures:
                     ME      RMSE       MAE        MPE     MAPE     MASE        ACF1
Training set 0.02832275 1.133521 0.7397459 0.2264387 6.469925 0.255107 0.03245788
```

Please find below the accuracy model of ARIMA model over test data:

```
> accuracy_ARIMA112
                    ME     RMSE       MAE        MPE      MAPE     MASE        ACF1 Theil's U
Training set 0.02832275 1.133521 0.7397459  0.2264387  6.469925 0.255107 0.03245788        NA
Test set     2.93665755 4.298014 3.3173169 13.8384977 16.599427 1.144002 0.79336629  2.618581
```

Prediction by SARIMA model

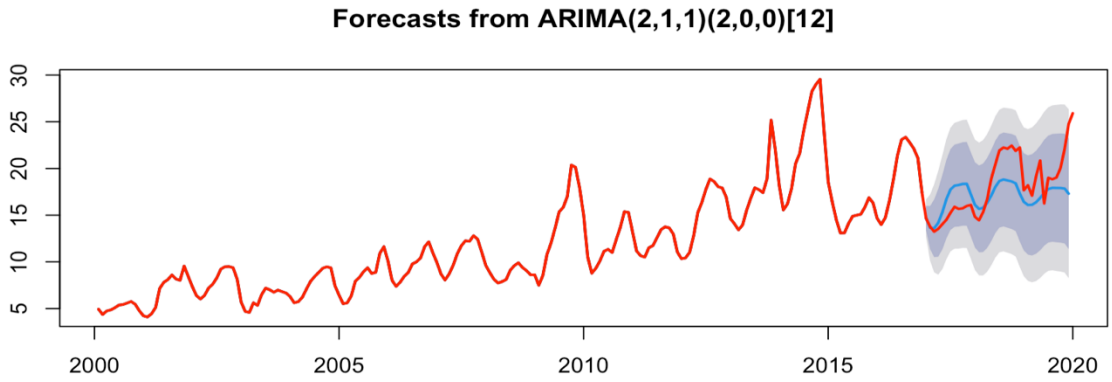
We use the SARIMA model as SARIMA(1,1,0)(2,0,0)[12] where seasonality is monthly and differencing is 1.

Please find below the output of R for this SARIMA model.

```
ARIMA(2,1,1)(2,0,0)[12]

Coefficients:
         ar1      ar2      ma1    sar1    sar2
      1.4216  -0.5484  -0.9623  0.2873  0.2071
s.e.  0.0615   0.0613   0.0206  0.0708  0.0718

sigma^2 = 1.039:  log likelihood = -289.67
AIC=591.34   AICc=591.77   BIC=611.19

Training set error measures:
                    ME    RMSE       MAE        MPE     MAPE      MASE        ACF1
Training set 0.1012645 1.00435 0.6841754 0.6694799 5.946111 0.2359431 0.01707184
>
```

Please find below the accuracy and prediction graph for SARIMA model:

```
> accuracy_autoarima
                     ME      RMSE       MAE        MPE      MAPE      MASE       ACF1 Theil's U
Training set 0.1012645 1.004350 0.6841754 0.6694799  5.946111 0.2359431 0.01707184        NA
Test set     0.9225060 2.635103 2.1445928 3.2247286 11.230880 0.7395792 0.72132272   1.67883
```



Model Selection and accuracy matrix

Finally we compare the error and accuracy among all model and then select SARIMA as optimal Model.

Please find below the accuracy matrix among top performing model:

```
> accuracy_matrix
                                ME      RMSE       MAE         MPE      MAPE      MASE       ACF1 Theil's U
SARIMA(1,1,0)(2,0,0)[12]  0.9225060 2.635103 2.144593   3.224729 11.23088 0.7395792 0.7213227  1.678830
ARIMA(112)                2.9366575 4.298014 3.317317  13.838498 16.59943 1.1440020 0.7933663  2.618581
Exp Smoothing(MNM)        0.9883104 2.923933 2.457949   3.659363 13.30077 0.8476424 0.7528196  1.964397
Time Series Regression   -3.0454576 3.844280 3.195134 -19.339841 19.95958 1.1018662 0.7154159  2.983586
Seasonal Naive           -1.2519699 3.824953 2.919495  -8.831986 17.06481 1.0068099 0.7435786  2.888436
```

As we can see here, RMSE and MAPE are minimum for SARIMA model among all models.

Forecast for next year (12 months) i.e. 2020-21

Finally after selecting SARIMA model, we forecast for next year on the same model and below are the values for the same:

```
> forecasted_values <- predict_autoarima_next_year$mean
> print(forecasted_values)
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2020          26.90827 27.21072 28.26405 29.37823 28.62342 29.77554 29.82659 29.84022 30.20357 30.57713 31.38782
2021 30.39040
```

Please find below the graph as well: