Forecast of the auto sales in the U.S

Xi Cheng

May 3, 2018

This project will provide a ARIMA model and residuals model to predict the auto sales data in the USA.

Data Source

U.S. Bureau of Economic Analysis, Motor Vehicle Retail Sales: Domestic Autos*, retrieved from:https://fred.stlouisfed.org/series/DAUTONSA

Data can also be obtained from : https://fred.stlouisfed.org/series/DAUTONSA?utm_source=series_page&utm_medium=related_content&utm_term=other_formats&utm_campaign=other_format

Release: Supplemental Estimates, Motor Vehicles Units: Thousands of Units, Not Seasonally Adjusted Frequency: Monthly

Autos are all passengers cars, including station wagons. Domestic sales are all United States sales of vehicles assembled in the U.S.

First of all , we read teh data from the csv file(this file is up to date, and for now, it is updated to March of 2018). And the data looks like:

```
##         DATE DAUTONSA
## 1 1967-01-01    564.1
## 2 1967-02-01    509.1
## 3 1967-03-01    670.4
## 4 1967-04-01    710.2
## 5 1967-05-01    744.8
## 6 1967-06-01    780.2
```

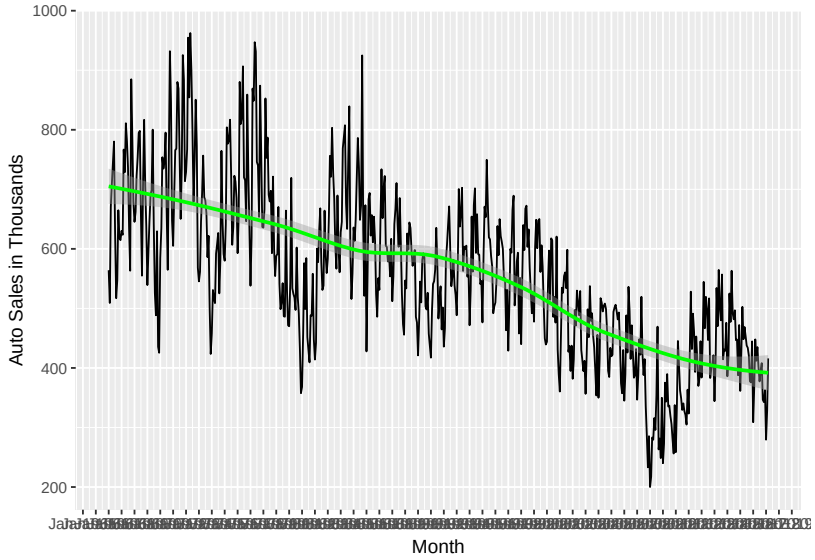
We will be using the monthly domestic auto sales(in thousands) for building our time series

Before we proceed, let's check if we have any missing data. Amelia library gives us a missmap function that shows the missing details in a visual map.

Since no missing data is found, I would just go ahead with visually plotting the daily count to see if I can identify any trends, seasonality, cycle or outlier from the data.

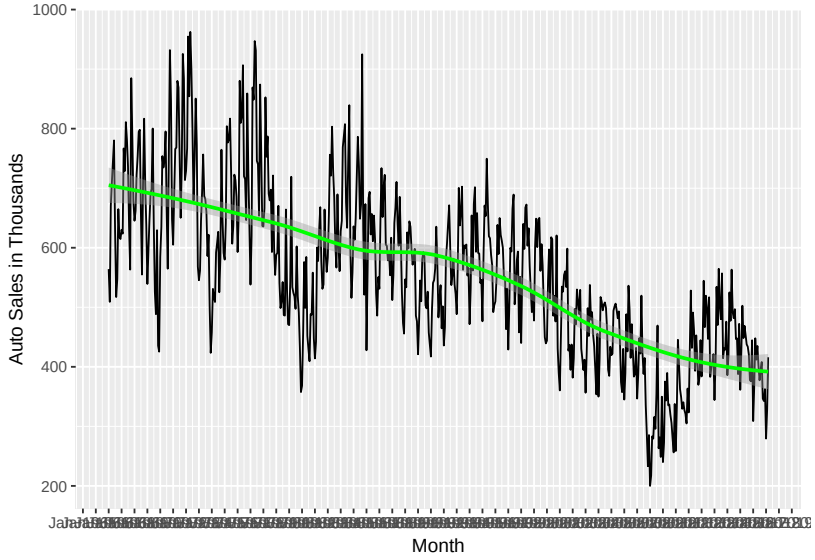
Before I plot the count data, I convert the dteday field from character to date type.

Plot the monthly sales data:

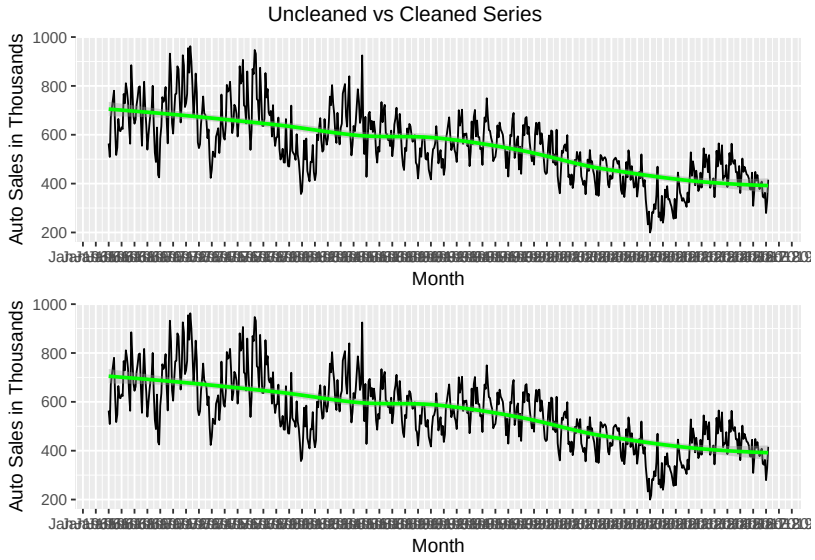


There seems to be an outlier that we could see from the plot. I need to remove the outlier before proceed with stationarizing the series.

Plot the cleaned monthly sales data:



Now, let's compare both cleaned and uncleaned plots:



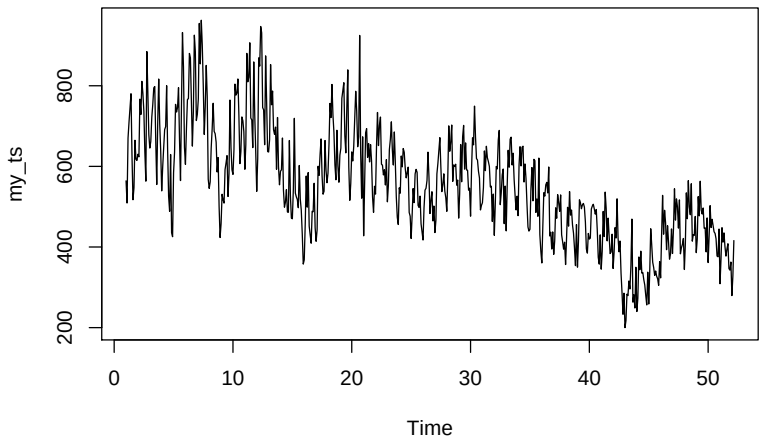
Smoothing the series

If data points are still volatile, then we can apply smoothing smoothing. By applying smoothing, we can have a better idea about the series and it's components. It also makes the series more predictable. In this case, I would use the quarterly/biannualy moving average. If the data points are on a daily basis,many level of seasonality(daily, weekly, monthly or yearly) can be incorporated.

However, looking at the graph, our data does not require any smoothing. Therefore, We go ahead with the cleaned data.

As data is monthly, we used frequency =12 in above command. We also ignore the NA values.

Next, we plot the cleaned series to infer visual cues from the graph.



Identify Level of Differencing Required

Now that the series is cleaned, we need to remove trend by using appropriate order of difference and make the series stationary. We do this by looking at acf, Dickey-Fuller Test and standard deviation.

Dickey Fuller test:

X(t) = Rho * X(t-1) + Er(t)

=> X(t) - X(t-1) = (Rho - 1) X(t - 1) + Er(t)

We have to test if Rho - 1 is significantly different than zero or not. If the null hypothesis gets rejected, we'll get a stationary time series.

Stationary testing and converting a series into a stationary series are the most critical processes in a time series modelling. We need to memorize each and every detail of this concept to move on to the next step of time series modelling.

To confirm that the series is not stationary, we perform the augmented Dickey-Fuller Test.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  my_ts
## Dickey-Fuller = -4.7461, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

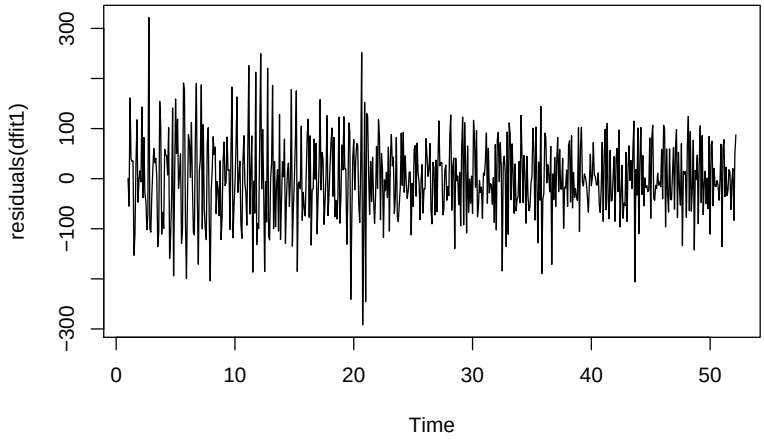
P value is 0.01 indicating the null hypothesis 'series is non-stationary' is true i.e the series is not stationary.

Plot the auto-correlation plot for the series to identify the order of differncing required.

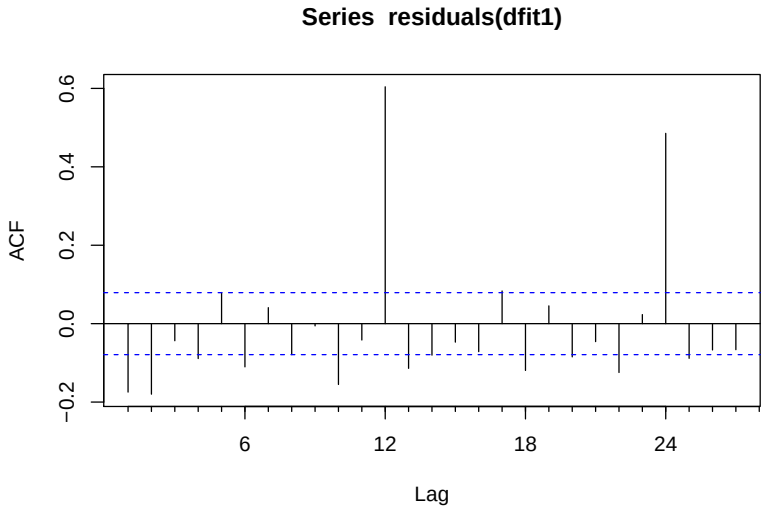
ACF plot shows positive correlation at higher lags. this indicates that we need differencing to make the series stationary.

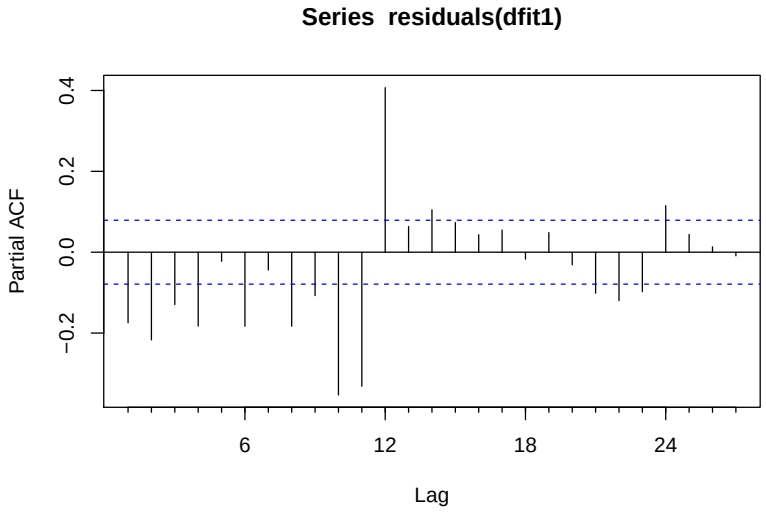
Let's try order difference We will fit ARIMA(0,d,0)(0,D,0)[12] models and verify acf residuals to find which 'd' or 'D' order of differencing is appropriate in our case.

Applying only one order of difference i.e ARIMA(0,1,0)(0,0,0)



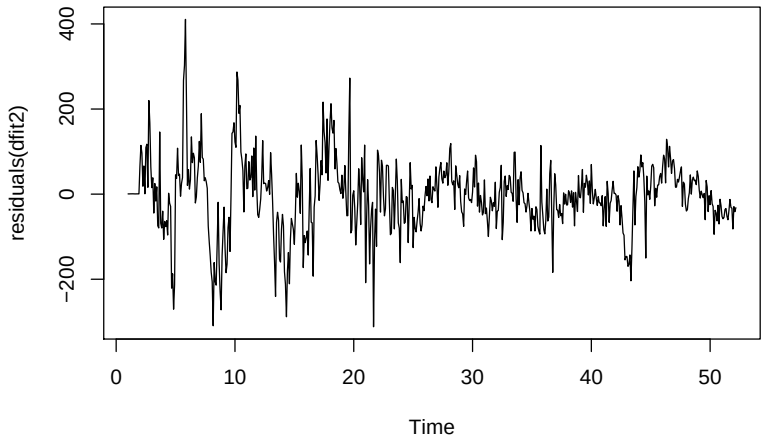
Below is the acf and Pacf plot of residuals.



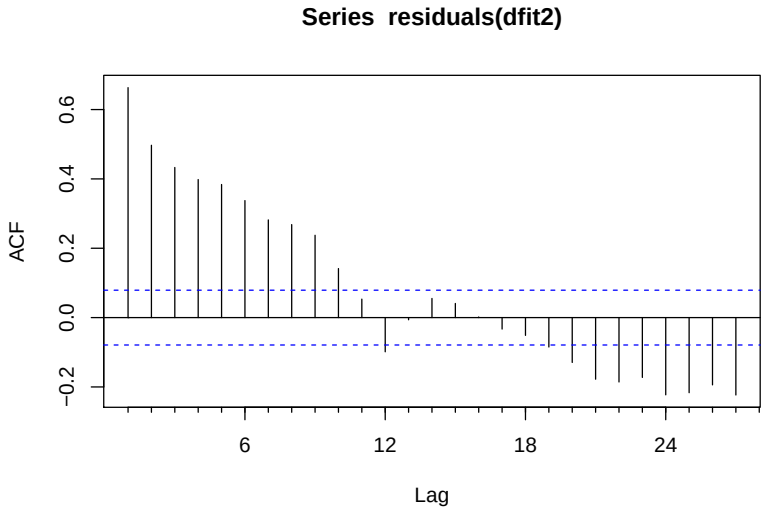


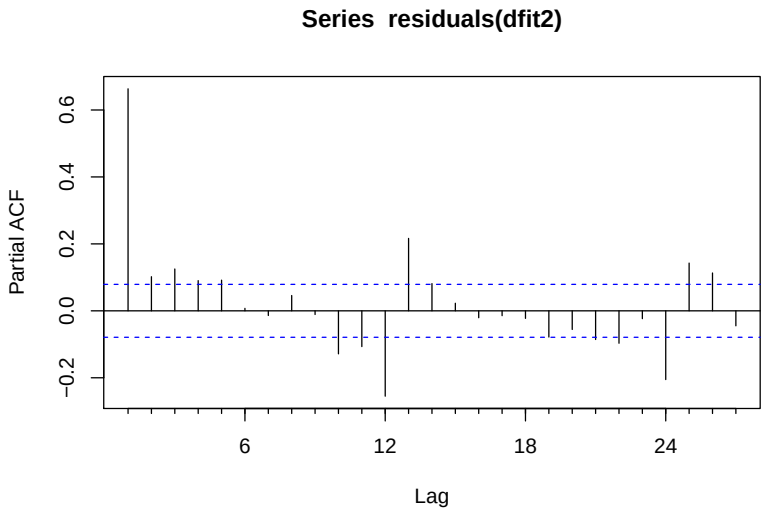
The differenced series still shows some strong autocorrelation at the seasonal period 12 and 24. Because the seasonal pattern is strong and stable, we know that we will want to use an order of seasonal differencing in the model.

Before that let's try only with one seasonal difference



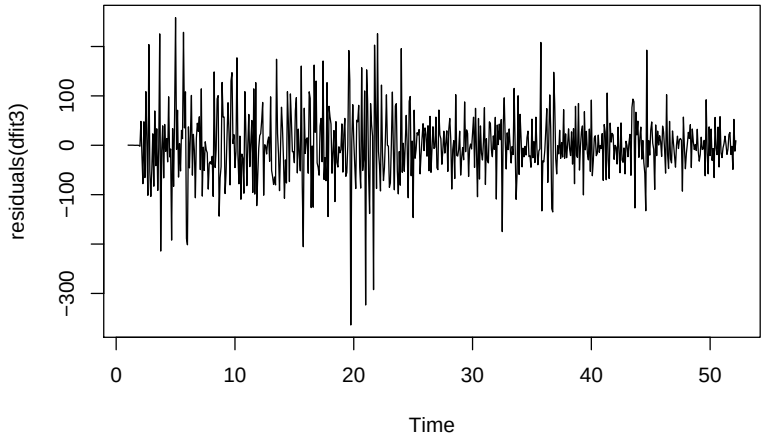
Looking a the residual plot, residuals does not look like white noise. We also need to check the acf and pacf plots of residuals.



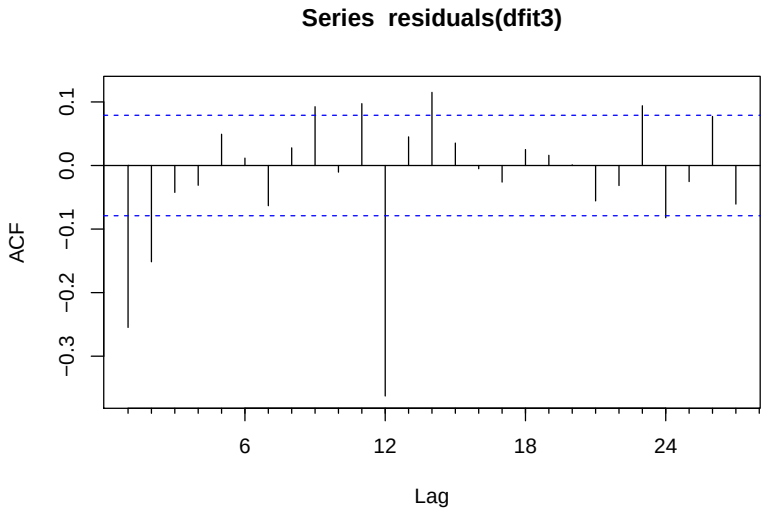


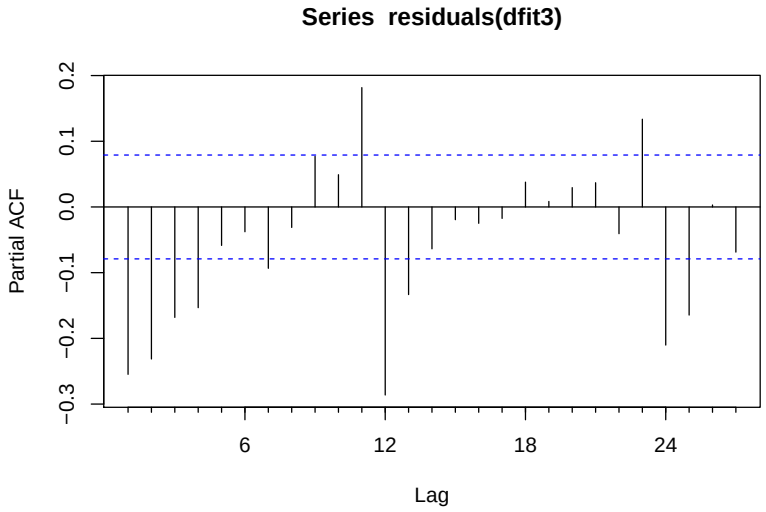
The seasonally differenced series shows a very strong pattern of positive autocorrelation and is similar to a seasonal random walk model. The correlation plots indicate an AR signature and/or incorporating another order of difference into the model.

Let's go ahead and apply both seasonal and non-seasonal differencing i,e ARIMA(0,1,0)(0,1,0)[12]



Residuals seems to return to the mean and we don't see any pattern in the residuals. Below is the acf and Pacf plot of residuals.





ACF at lag 1 is -ve and slightly smaller than -0.2. We know that if the lag 1 acf falls below -0.5, then the series is over differenced. Positive spikes in acf have become negative, another sign of possible over differencing. Therefore, this model might be suffering from slight over differencing. This overdifferencing can be compensated by adding a MA term.

To select the appropriate order of differencing, we have to consider the error statistics, the standard deviation in specific.

In below summary, SD is same as RMSE.

```
##
## Call:
## arima(x = my_ts, order = c(0, 1, 0))
##
##
## sigma^2 estimated as 6439:  log likelihood = -3563.67,  aic = 7129.33

##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.2398958 80.17913 63.03978 -1.093049 11.61302 0.9983885
##                    ACF1
## Training set -0.174578

##
## Call:
## arima(x = my_ts, order = c(0, 0, 0), seasonal = list(order = c(0, 1, 0), period = 12))
##
##
## sigma^2 estimated as 7473:  log likelihood = -3544.73,  aic = 7091.47

##                      ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -4.967695 85.60185 61.45344 -2.223507 11.52811 0.973265
##                    ACF1
## Training set 0.6632709

##
## Call:
## arima(x = my_ts, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 12))
##
## Coefficients:
```

```
##          ar1      ma1    sar1     sma1
##       0.3136  -0.7543  0.1753  -0.8545
## s.e.  0.0627   0.0417  0.0495   0.0264
##
## sigma^2 estimated as 2834:  log likelihood = -3253.29,  aic = 6516.59
```

```
##                       ME     RMSE      MAE        MPE    MAPE      MASE
## Training set -1.050691 52.66835 38.34043 -0.6363904 7.03846 0.6072141
##                      ACF1
## Training set 0.02134979
```

Selecting appropriate order of differencing:

The optimal order of differencing is often the order of differencing at which the standard deviation is lowest. (Not always, though. Slightly too much or slightly too little differencing can also be corrected with AR or MA terms.

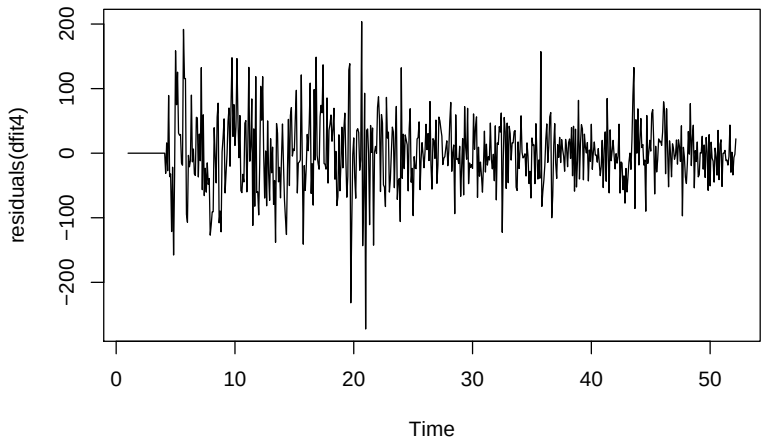
Out of the above, dfit3 model i.e ARIMA(1,1,1)(1,1,1)12 has the lowest standard deviation(RMSE) and AIC. Therefore, it seems like the correct order of differencing. But we will find that it isn't.

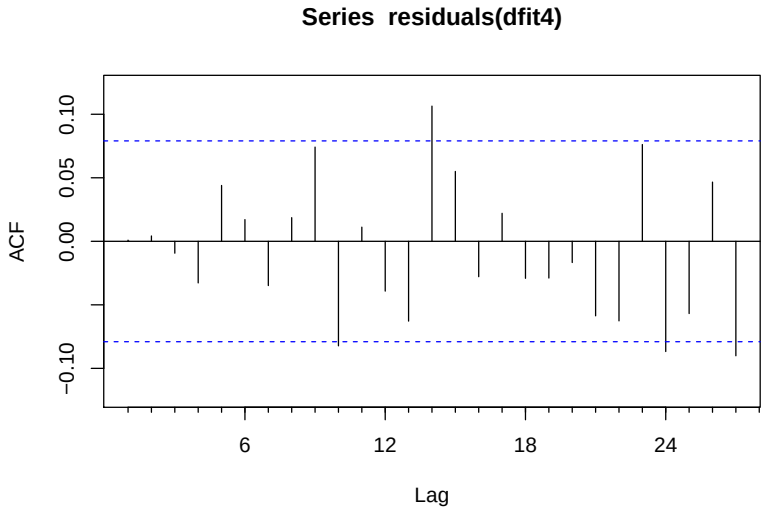
Therefore, the value of d=1 and D=1 is set. Now, we need to identify AR/MA and SAR/SMA values and fit the model.

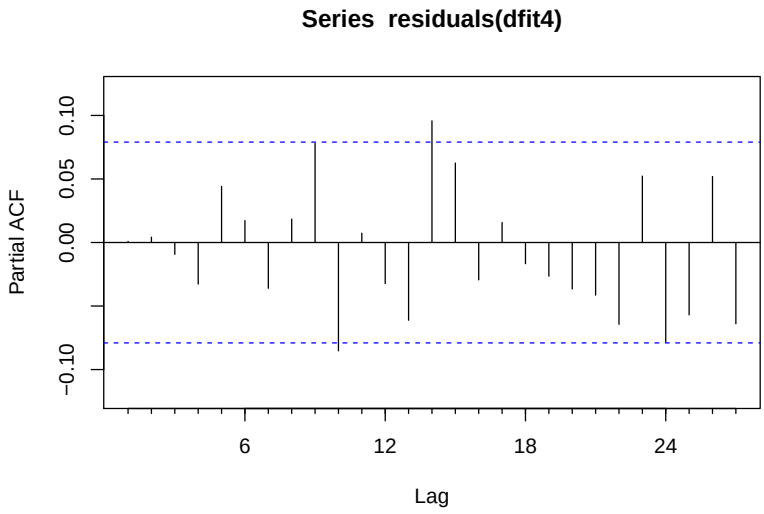
Identifying the AR/MA(p/q) and SAR/SMA(P/Q) components.

Looking back at the correlation plot of model dfit3, ACF is negative at lag 1 and shows sharp cut-off immediately after lag 1, we can add a MA to the model to compensate for the overdifferencing.

Since, we do not see any correlation at lag s,2s,3s etc i.e 12,24,36 etc, we do not need to add SAR/SMA to our model.







Looking at he residual plot for above model, slight amount of correlation remains at lag 10 and 22, but the overall plots seem good.

Thus, this model seems like a good fit pending statistically significant MA co-efficient and low AIC.

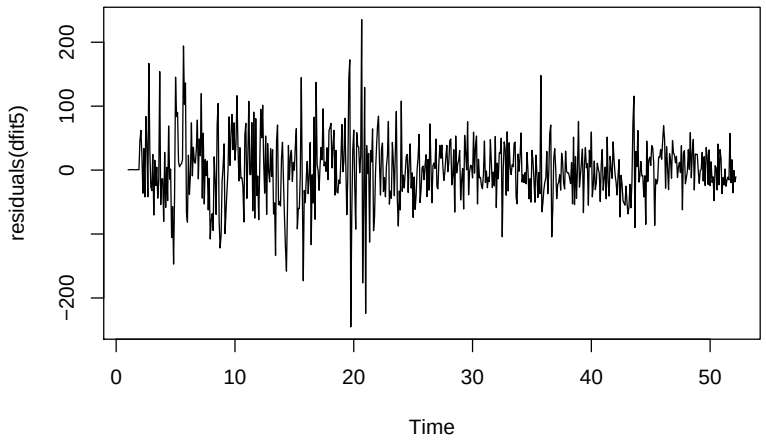
check for Statistical Significance

Let's check the model parameter's significance.

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.107440   0.187592  0.5727  0.566824
## ma1  -0.501720   0.184796 -2.7150  0.006628 **
## ma2  -0.158385   0.102514 -1.5450  0.122343
## sar1  0.448166   0.039489 11.3491 < 2.2e-16 ***
## sar2  0.122392   0.042863  2.8555  0.004298 **
```

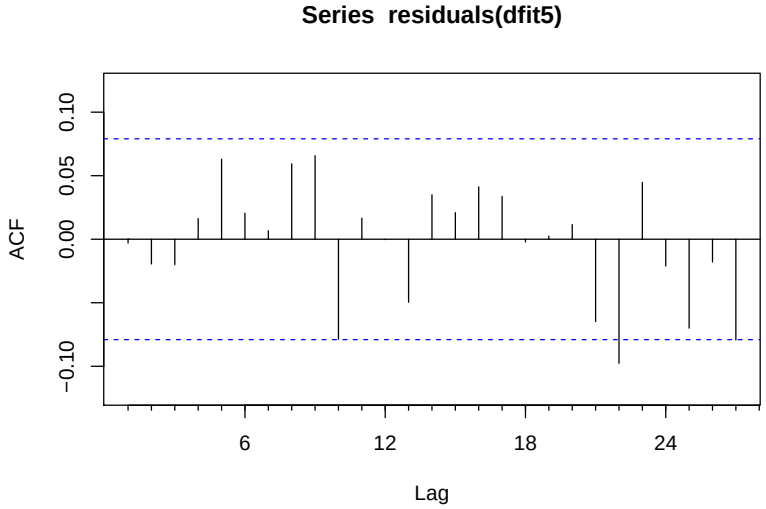
```
## sar3  0.173663   0.038547  4.5052 6.631e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

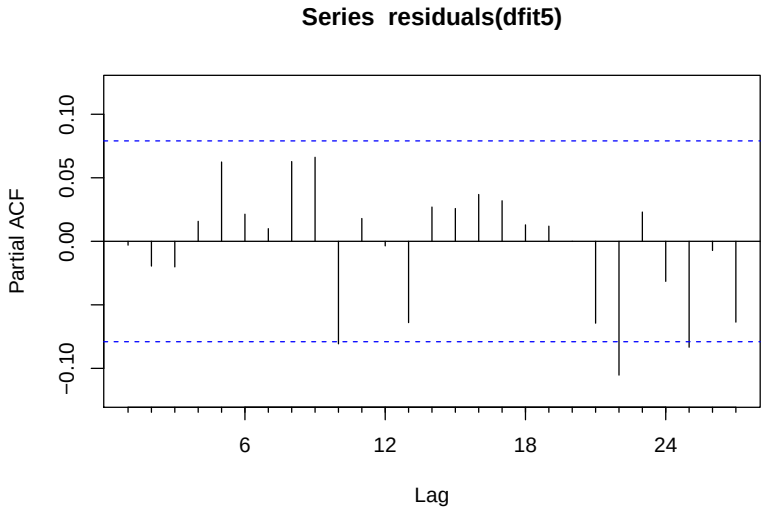
As we can see, P value is negligible and thus the test confirms that MA1 coefficient is statistically significant.



Residual plot is similar to that of the model we built above.

Next, we see the acf, pacf and summary of the auto built model.






```
##
## z test of coefficients:
##
##        Estimate Std. Error  z value  Pr(>|z|)
## ar1    0.946814   0.017459  54.2293 < 2.2e-16 ***
## ma1   -0.386043   0.044006  -8.7726 < 2.2e-16 ***
## ma2   -0.195999   0.040851  -4.7979 1.603e-06 ***
## sma1  -0.665899   0.040285 -16.5295 < 2.2e-16 ***
## sma2  -0.160354   0.039362  -4.0738 4.625e-05 ***
## drift -0.501368   0.266833  -1.8790   0.06025 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Auto arima gives us ARIMA(1,1,2)(1,2,0)[12] Auto arima gives us ARIMA(1,1,2)(1,2,0)[12]. All coefficients are significant.

Clearly this model performs worse than the model we built earlier as ARIMA(1,1,2)(3,0,0)[12]

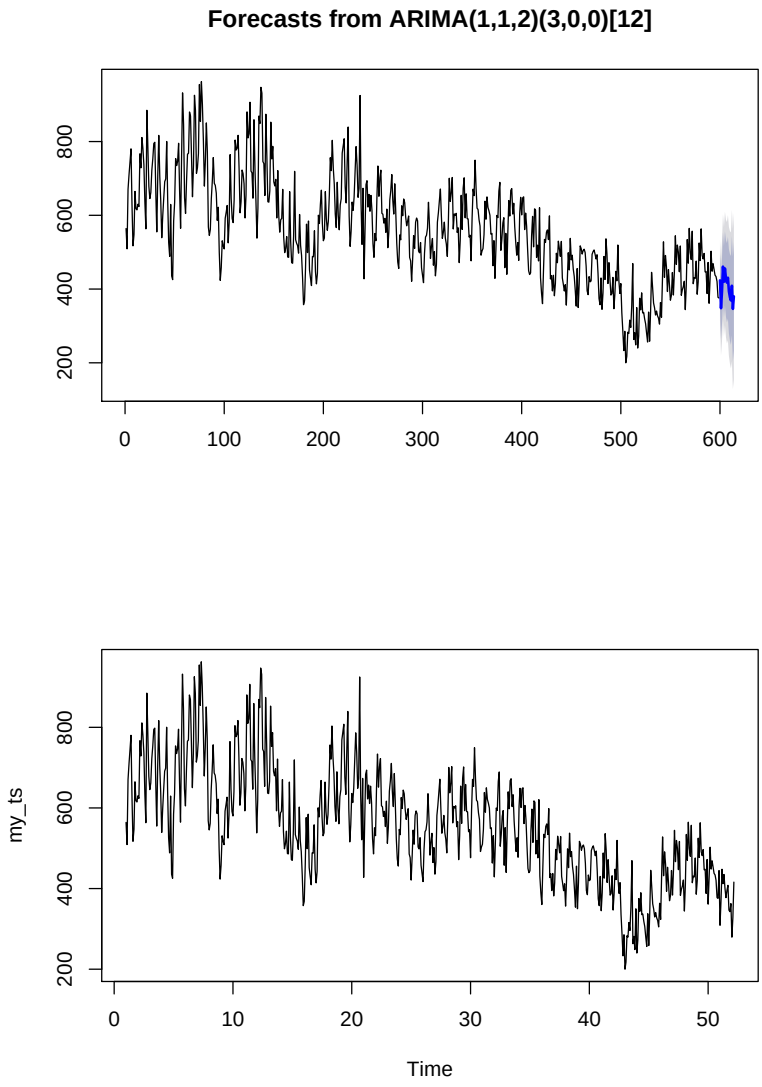
By rule of parsimony and/or minimum AIC, we can reject ARIMA(1,1,2)(1,2,0)[12] and accept ARIMA(1,1,2)(3,0,0)[12] as our model.

Model Validation

To see how our model will perform in future, we can use n-fold holdout method.

Fit the model to predict for observation(months) 72 through 83.

Use the above model to forecast values for last 10 months. Forecasting using a fitted model is straightforward in R. We can specify forecast horizon h periods ahead for predictions to be made, and use the fitted model to generate those predictions:

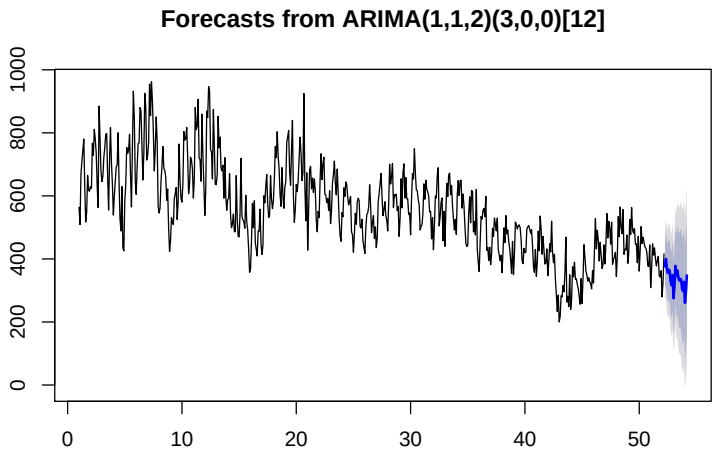


In the above graph, blue line is the predicted data and the confidence bands are in dark grey(80%) and light grey(95%).

Model's prediction is pretty good and we can see predicted sales closely follow the actual data. This is an indication of a good model.

Sales Prediction for 2018/19.

Next step in our model is to forecast values i.e the monthly sales data. We have specified h=24 to predict for next 24 observations(months) i.e next 2 years - 2018 and 2019.



So, from the forecasting model, we can see that the auto sales seems like is gonna keep going down in the next 2 years. But seems like it is gonna going up at the the beginning of the 2020.

Different car brands

After predict the value of the total sales in the US market, I want to do something more. I want to predict all the different brands of passenger automotives perform in the next month(we only predict MAY)

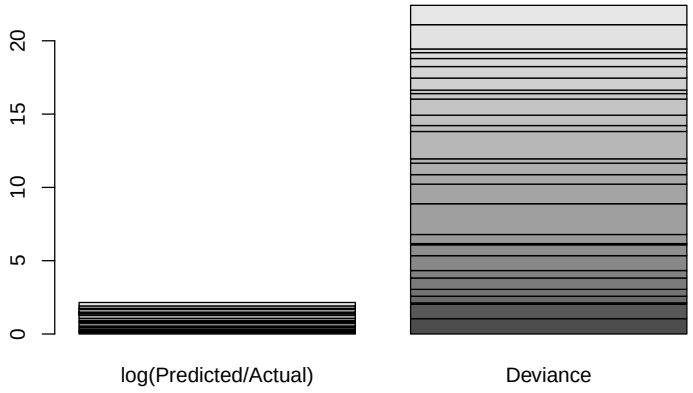
So, I'll produce predictions for US car sales by manufacture every month. Unlike above analysis, I'll try to focus on the residuals (the stuff I can't predict) to tell the story.

The Autoblog article https://www.autoblog.com/2014/10/01/september-2014-by-the-numbers/ highlights Mitsubishi for increasing sales. However, my prediction for Mitsubishi sales are pretty much exactly what the sales were. In essence, given this model, we didn't learn much. On the other hand, Land Rover and Jaguar had the largest residuals (in percent terms) and Land Rover and Acura had the largest deviance (Residual / Variance). I think these results are more telling because we didn't predict them correctly; something might have changed.

```
##
## <table style="text-align:center"><tr><td colspan="2" style="border-bottom: 1px solid black"></td></t
## <tr><td colspan="2" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">Audi</td><td>590.126</td></tr>
## <tr><td style="text-align:left">BMW</td><td>1,051.492</td></tr>
## <tr><td style="text-align:left">Buick</td><td>666.206</td></tr>
## <tr><td style="text-align:left">Cadillac</td><td>533.468</td></tr>
## <tr><td style="text-align:left">Chevrolet</td><td>5,691.774</td></tr>
## <tr><td style="text-align:left">Chrysler</td><td>1,097.289</td></tr>
## <tr><td style="text-align:left">Dodge</td><td>1,493.426</td></tr>
## <tr><td style="text-align:left">Ford</td><td>6,639.675</td></tr>
## <tr><td style="text-align:left">GMC</td><td>1,572.525</td></tr>
## <tr><td style="text-align:left">Honda</td><td>4,049.582</td></tr>
## <tr><td style="text-align:left">Hyundai</td><td>1,998.905</td></tr>
## <tr><td style="text-align:left">Infiniti</td><td>262.475</td></tr>
## <tr><td style="text-align:left">Jaguar</td><td>40.422</td></tr>
## <tr><td style="text-align:left">Jeep</td><td>2,132.637</td></tr>
## <tr><td style="text-align:left">Kia</td><td>1,650.236</td></tr>
```

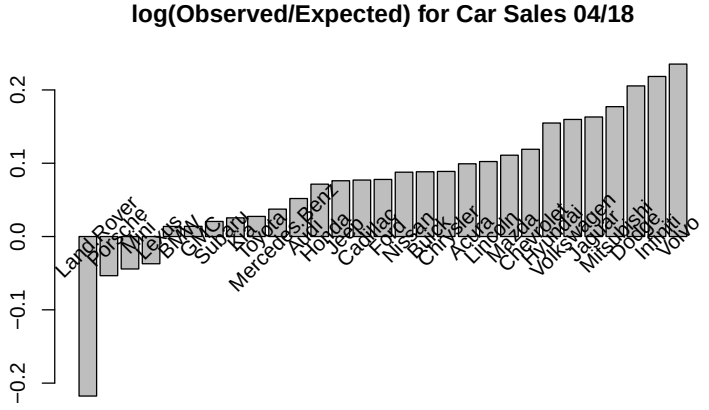
```
## <tr><td style="text-align:left">Land.Rover</td><td>160.884</td></tr>
## <tr><td style="text-align:left">Lexus</td><td>944.946</td></tr>
## <tr><td style="text-align:left">Lincoln</td><td>272.964</td></tr>
## <tr><td style="text-align:left">Mazda</td><td>894.234</td></tr>
## <tr><td style="text-align:left">Mercedes.Benz</td><td>1,184.982</td></tr>
## <tr><td style="text-align:left">Mini</td><td>183.743</td></tr>
## <tr><td style="text-align:left">Mitsubishi</td><td>193.982</td></tr>
## <tr><td style="text-align:left">Nissan</td><td>3,630.149</td></tr>
## <tr><td style="text-align:left">Porsche</td><td>158.540</td></tr>
## <tr><td style="text-align:left">Subaru</td><td>1,694.661</td></tr>
## <tr><td style="text-align:left">Toyota</td><td>5,895.316</td></tr>
## <tr><td style="text-align:left">Volkswagen</td><td>923.242</td></tr>
## <tr><td style="text-align:left">Volvo</td><td>153.691</td></tr>
## <tr><td colspan="2" style="border-bottom: 1px solid black"></td></tr></table>
```

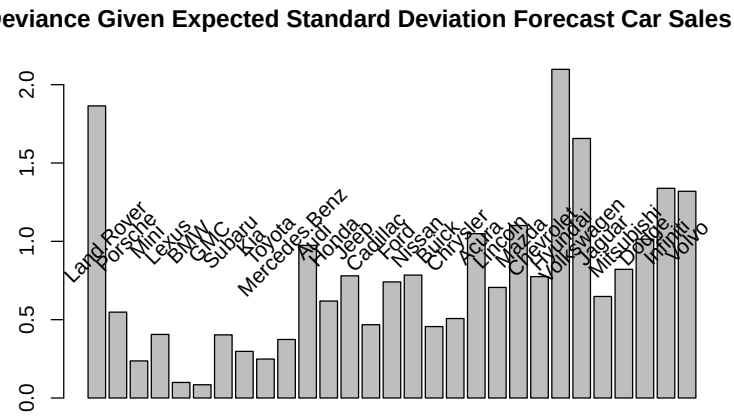
```
##
## <table style="text-align:center"><tr><td colspan="5" style="border-bottom: 1px solid black"></td></t
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">Audi</td><td>590.126</td><td>621.542</td><td>0.052</td><td>0.974</td
## <tr><td style="text-align:left">BMW</td><td>1,051.492</td><td>1,066.083</td><td>0.014</td><td>0.099<
## <tr><td style="text-align:left">Buick</td><td>666.206</td><td>727.750</td><td>0.088</td><td>0.456</td
## <tr><td style="text-align:left">Cadillac</td><td>533.468</td><td>576.208</td><td>0.077</td><td>0.468
## <tr><td style="text-align:left">Chevrolet</td><td>5,691.774</td><td>6,411.375</td><td>0.119</td><td>0
## <tr><td style="text-align:left">Chrysler</td><td>1,097.289</td><td>1,199.208</td><td>0.089</td><td>0
## <tr><td style="text-align:left">Dodge</td><td>1,493.426</td><td>1,834.167</td><td>0.206</td><td>1.014
## <tr><td style="text-align:left">Ford</td><td>6,639.675</td><td>7,177.542</td><td>0.078</td><td>0.741
## <tr><td style="text-align:left">GMC</td><td>1,572.525</td><td>1,594.542</td><td>0.014</td><td>0.085<
## <tr><td style="text-align:left">Honda</td><td>4,049.582</td><td>4,349.625</td><td>0.071</td><td>0.619
## <tr><td style="text-align:left">Hyundai</td><td>1,998.905</td><td>2,333.750</td><td>0.155</td><td>2.0
## <tr><td style="text-align:left">Infiniti</td><td>262.475</td><td>326.542</td><td>0.218</td><td>1.339
## <tr><td style="text-align:left">Jaguar</td><td>40.422</td><td>47.583</td><td>0.163</td><td>0.648</td
## <tr><td style="text-align:left">Jeep</td><td>2,132.637</td><td>2,301.292</td><td>0.076</td><td>0.780
## <tr><td style="text-align:left">Kia</td><td>1,650.236</td><td>1,692.833</td><td>0.025</td><td>0.298<
## <tr><td style="text-align:left">Land.Rover</td><td>160.884</td><td>129.417</td><td>-0.218</td><td>1.8
## <tr><td style="text-align:left">Lexus</td><td>944.946</td><td>910.500</td><td>-0.037</td><td>0.406</t
## <tr><td style="text-align:left">Lincoln</td><td>272.964</td><td>302.375</td><td>0.102</td><td>0.706<
## <tr><td style="text-align:left">Mazda</td><td>894.234</td><td>999.167</td><td>0.111</td><td>1.101</td
## <tr><td style="text-align:left">Mercedes.Benz</td><td>1,184.982</td><td>1,230.125</td><td>0.037</td>
## <tr><td style="text-align:left">Mini</td><td>183.743</td><td>175.792</td><td>-0.044</td><td>0.237</td
## <tr><td style="text-align:left">Mitsubishi</td><td>193.982</td><td>231.583</td><td>0.177</td><td>0.8
## <tr><td style="text-align:left">Nissan</td><td>3,630.149</td><td>3,963.250</td><td>0.088</td><td>0.78
## <tr><td style="text-align:left">Porsche</td><td>158.540</td><td>150.292</td><td>-0.053</td><td>0.548
## <tr><td style="text-align:left">Subaru</td><td>1,694.661</td><td>1,729.875</td><td>0.021</td><td>0.40
## <tr><td style="text-align:left">Toyota</td><td>5,895.316</td><td>6,059.458</td><td>0.027</td><td>0.24
## <tr><td style="text-align:left">Volkswagen</td><td>923.242</td><td>1,083.167</td><td>0.160</td><td>1
## <tr><td style="text-align:left">Volvo</td><td>153.691</td><td>194.458</td><td>0.235</td><td>1.320</td
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr></table>
```



I have made a shiny app to show the final result, and here is the shiny URL: https://chengxi.shinyapps.io/CarSalesShiny/ . From this shiny website , you can see all the prediction of the sales of different main brands in the US market.

And then, let us test the difference between actual data and the expected data for car sale in April:





From these two figures, we can see that the difference between the actual value and the predicted value is not that big. So we can see that it is a useful way to predict the auto sales of various brands.

Summary:

All this project is about prediction: The prediction of total value, and the prediction of various brands. From the result which is credible, as investors or the managers of Auto Industry, they can choose differnet ways to handle the situiation in the future.