CASE STUDY#3

FORECASTING WALMART REVENUE WITH AR & ARIMA MODELS

Alarmelu Pichu Mani – TJ6723

Q1-Identify time series predictability.

1a-Using the AR(1) model for the historical data, Provide and explain the AR(1) model summary in your report. Explain if the Walmart revenue is predictable.

The predictability of a time series tells us that how usable the historical data and the data patterns in the data in the process of forecasting. It can be identified using the summary of the model in R.

```
> summary(revenue.ar1)
Series: revenue.ts
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1        mean
      0.8697  110533.610
s.e.  0.0702    8319.407

sigma^2 estimated as 90976908:  log likelihood=-655.77
AIC=1317.55   AICc=1317.96   BIC=1323.93

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE     MASE       ACF1
Training set 856.1966 9383.079 7808.116 0.08675575 7.117372 1.875218 -0.6152061
>
```

Figure 1- Summary output in R

By fitting the Walmart's quarterly time series data into the AR (1) in R and examining the summary, we can infer if the time series is predictable
The AR (1) model we get is $Y_t = 110533.61 + 0.87 * Y_{t-1}$, where 0.87 is the β_1 which is the coefficient of regression.

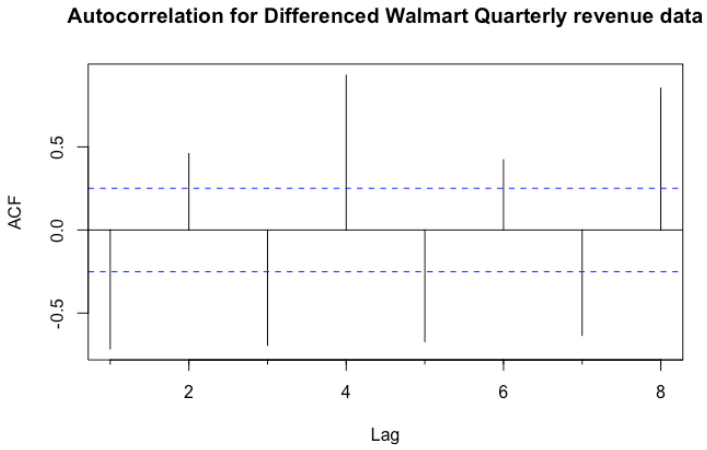
If β_1 is very close to 1 or equal to 1, then the model becomes a naïve forecast model. This means that the changes in the historical data cannot always be explained clearly by taking into consideration trend or seasonality. Hence, we can say that the time series is not predictable, or it is a 'random walk' based on the random walk hypothesis predominantly referred to in the finance domain to explain the behavior of stock price changes.

In our case, 0.87 though is skewed towards 1, it is not exactly very close to 1. Hence, we can say that the Walmart quarterly revenue time series data is predictable.

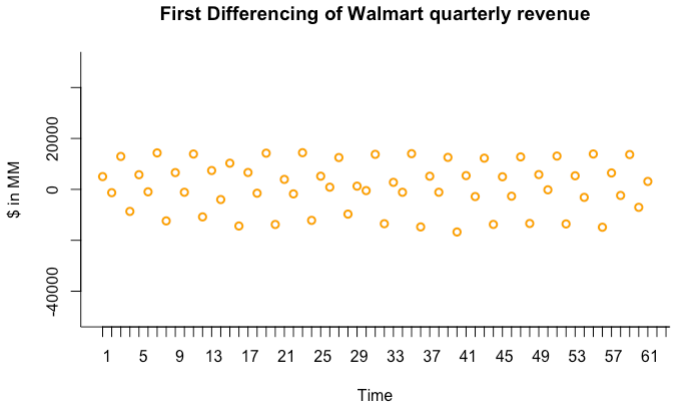
1b-Using the first differencing (lag-1) of the historical data and Acf() function Provide in the report the autocorrelation plot of the first differencing (lag-1) with the maximum of 8 lags and explain if Walmart revenue is predictable.

The first differencing (lag 1) is the mathematical equivalent of the approach discussed in 1a. If the original series is a random walk, then the differenced series also behaves like a random walk. This differenced data series is tested for autocorrelation using the Acf () in R. If the Acf () function shows that the autocorrelation coefficients at different lags are statistically significant (out of the horizontal threshold), then it can be inferred that the time series is predictable.

As for our Walmart revenue time series, we can see from the Autocorrelation for differenced data below, we can see that the autocorrelation coefficients are significant at every lag. Hence implying that the Walmart time series is in fact predictable.



Also,the scatterplot of the first differencing of the Walmart revenue shows that the data stays with in the ±2000 region roughly. There are no outliers on either ends of the y-limits. This also reinforces our earlier inference that the time series is predictable.



Q2- Apply the two-level forecast with regression model and AR model for residuals.

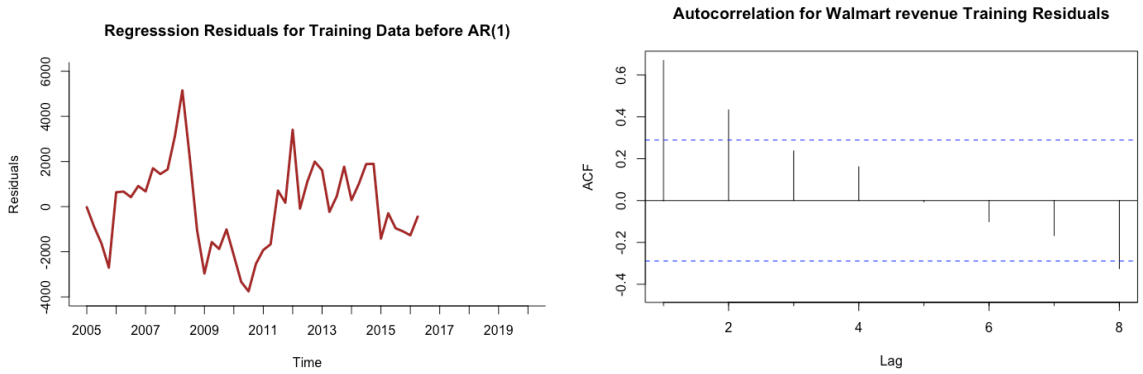
2-a. For the training data set, use the tslm () function to develop a regression model with quadratic trend and seasonality. Forecast Walmart's revenue with the forecast () function (use the associated R code from case #2). No explanation is required in your report.

Using the quadratic trend regression model from our earlier case study, the residuals have been identified and is reevaluated to see if a better forecast can be identified using the AR (1) model.

2b- Identify the regression model's residuals for the training period and use the Acf() function with the maximum of 8 lags to identify autocorrelation for these residuals. Provide the autocorrelation plot in your report and explain if it would be a good idea to add to your forecast an AR model for residuals.

The Acf() function in R helps us identify the autocorrelation coefficients for the training period.

Acf (train.Quadtrend.season.pred$residuals, lag.max = 8, main = "Autocorrelation for Walmart revenue Training Residuals")



The output of the Acf () gives us the above correlogram. In this we can see that lags 1,2 and 8 have significant non-random autocorrelation coefficients. This implies that the residuals do have a significant information that can be incorporated into our forecasting to make it more accurate and productive. AR (1) would be a good fit for this residual as it would be able to efficiently capture the meaningful variations in the residuals and forecast them.

2c- Develop an AR (1) model for the regression residuals, present and explain the model and its equation in your report. Use the Acf () function for the residuals of the AR (1) model (residuals of residuals), present the autocorrelation chart, and explain it in your report.

The AR(1) model incorporates the autocorrelation directly in the regression model, using the past observations as predictors. The AR(1) model is represented by the general equation $Y_t = \alpha+\beta_1 Y_{t-1}+\varepsilon_1$

In R, the Arima (1,0,0) function develops the AR (1) model as follows.

res.ar1 <- Arima (train. Quadtrend.season$residuals, order = c(1,0,0))

summary(res.ar1)

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

Coefficients:
         ar1      mean
      0.6564  -18.1858
s.e.  0.1067  569.9711

sigma^2 estimated as 1996918:  log likelihood=-398.19
AIC=802.39   AICc=802.96   BIC=807.87

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set 0.2132412 1382.062 1094.233 113.7081 151.5113 0.5948615 0.02935792
>
```

Figure 2 Summary output in R

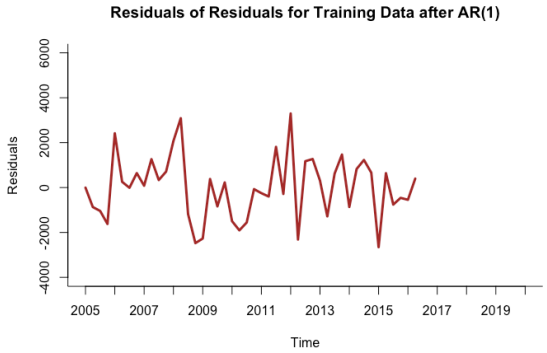
The summary () output of the AR (1) model on the regression residuals give us the β_1 and the α for the equation.

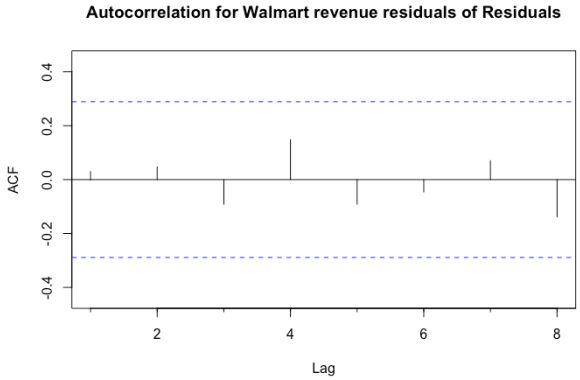
Hence the model can be now expressed using the following equation,

$$e_t = -18.1858+0.6564*e_{t-1}$$

where e_t is the error in the residual in the lag t and e_{t-1} is the error in the lag t-1, as we are applying the AR (1) on the residuals of the regression model.

When we apply the Acf () on the residuals of the AR (1) model (residuals of the residuals) we see the following correlogram.





When we analyze this correlogram, we see that there are no more significant autocorrelation coefficients in any of the lags. This implies that all the potential information that could be captured from the residuals is also already captured and incorporated as part of the AR (1) model in the second level of the forecast. A plot of the residuals before and after the application of the AR (1) model also shows a change in the peaks and dips. This shows that the AR (1) has efficiently captured the major variations into the forecast and whatever is left is statistically insignificant.

2d- Create a two-level forecasting model (regression model with quadratic trend and seasonality + AR (1) model for residuals) for the validation period. Show in your report a table with the validation data, regression forecast for the validation data, AR () forecast for the validation data, and combined forecast for the validation period.

To create a complete two-level forecast model with quadratic trend and seasonality with AR (1) model for the residuals and to present it in a table format, the following piece of code is helpful.

```
valid.two. level. pred <- train.Quadtrend.season. pred$mean +
res.ar1.pred$mean

valid.df <- data.frame(valid.ts,
train.Quadtrend.season.pred$mean,res.ar1.pred$mean,
valid.two.level.pred)

names(valid.df) <- c("Walmart revenue", "Reg.Forecast",
"AR(1)Forecast", "Combined.Forecast")
```

Walmart revenue	Reg.Forecast	AR(1)Forecast	Combined.Forecast
118179	118979.2	-297.42701	118681.8
130936	131214.6	-201.48930	131013.1
117542	117466.4	-138.51251	117327.9
123355	121429.0	-97.17238	121331.8
123179	118950.5	-70.03531	118880.5
136267	131025.8	-52.22160	130973.6
122690	117117.5	-40.52808	117077.0
128028	120920.0	-32.85205	120887.2
124894	118281.4	-27.81324	118253.6
138793	130196.6	-24.50559	130172.1
123925	116128.3	-22.33434	116106.0
130377	119770.7	-20.90905	119749.8
127991	116972.0	-19.97345	116952.1
141671	128727.1	-19.35929	128707.8
134622	114498.7	-18.95613	114479.8
137742	117981.0	-18.69148	117962.3

Figure 3 - Two-level forecast output

2e- Develop a two-level forecast (regression model with quadratic trend and seasonality and AR(1) model for residuals) for the entire data set. Provide in your report the autocorrelation chart for the AR(1) model's residuals and explain it. Also, provide a data table with the models' forecasts for Walmart revenue in 2020-2021 (regression model, AR(1) for residuals, and two-level combined forecast).

In order to develop the forecast for the entire data set- revenue.ts, using the quadratic trend and seasonality model, the following code is helpful

trend.season <- tslm(revenue.ts ~ trend + I(trend^2) + season)

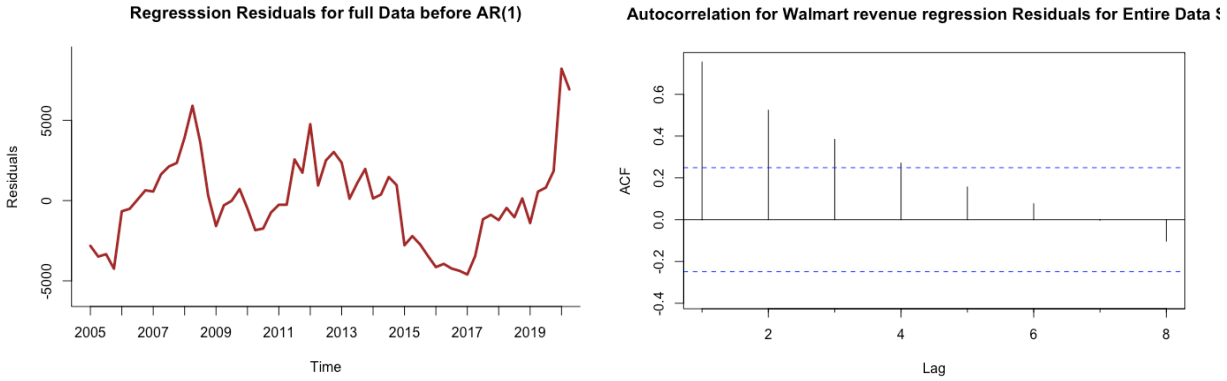
summary(trend.season)

```
> summary(trend.season)

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

Residuals:
    Min      1Q  Median      3Q     Max
-4605.0 -1701.3    25.3  1596.7  8218.6

Coefficients:
             Estimate Std. Error t value             Pr(>|t|)
(Intercept) 72987.016   1268.134  57.555 < 0.0000000000000002 ***
trend        1524.243     82.998  18.365 < 0.0000000000000002 ***
I(trend^2)    -10.632      1.277  -8.325      0.0000000000226 ***
season2      4193.148   1015.506   4.129             0.000122 ***
season3      1272.140   1033.433   1.231             0.223474
season4     13656.326   1033.634  13.212 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2872 on 56 degrees of freedom
Multiple R-squared:  0.9737,    Adjusted R-squared:  0.9713
F-statistic: 414.6 on 5 and 56 DF,  p-value: < 0.00000000000000022

>
```

Figure 4 - Summary output in R

The forecast for the period of 2020-2021 using the full data set is done using the forecast () function as follows.

trend.season.pred <- forecast(trend.season, h = 6, level = 0)

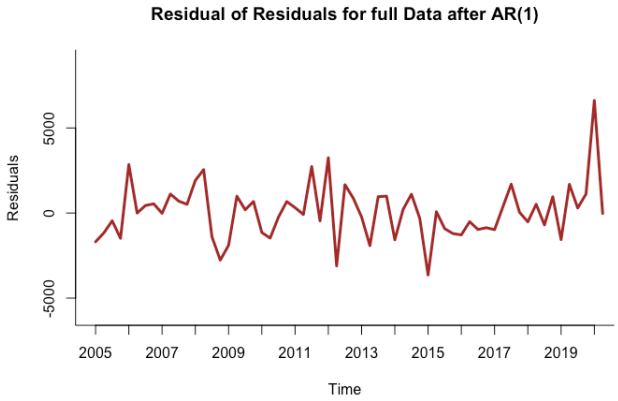


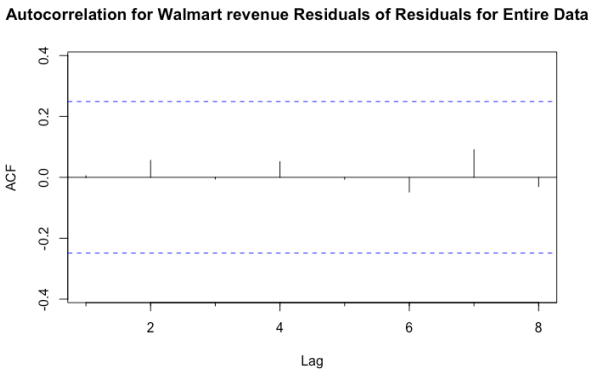
From the regression residual plot and the Acf () for the same data, we can see that lags 1 to 4 have statistically significant autocorrelation coefficients and hence the need for the second level forecast is evident.

Once the first level forecast using quadratic trend and seasonality are captured, we can get the residuals for the regression model and apply the AR (1) model to capture variations in the residuals and forecast the residuals. The code is as follows:

residual.ar1 <- Arima(trend.season$residuals, order = c(1,0,0))

residual.ar1.pred <- forecast(residual.ar1, h = 6, level = 0)





The above plots show the residuals of residuals for the full data after the application of AR(1) model. Though we can see some peaks and dips in the residual plot, the Acf () plot on the right gives a better idea of how much useful data is present in the residuals.

From the Autocorrelation for the residuals of residuals, we can see that there are no statistically significant autocorrelation coefficients present after the two-level forecasting.

After this step, we can develop a combined forecast using the quadratic trend predictions and the AR(1) predictions using the following piece of code and a table can be generated.

trend.season.ar1.pred <- trend.season.pred$mean + residual.ar1.pred$mean

trend.season.ar1.pred

table.df <- data.frame(trend.season.pred$mean, residual.ar1.pred$mean, trend.season.ar1.pred)

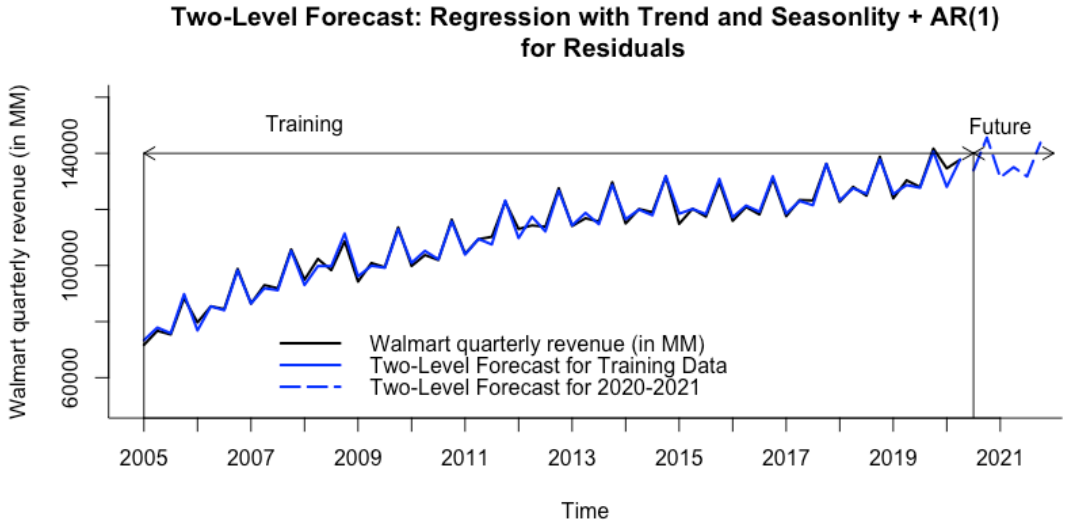
names(table.df) <- c("Reg.Forecast", "AR(1)Forecast","Combined.Forecast")

formattable(table.df)

Reg.Forecast	AR(1)Forecast	Combined.Forecast
128087.3	5870.718	133958.0
140645.4	4981.396	145626.8
127141.8	4234.004	131375.8
131466.4	3605.889	135072.3
128655.5	3078.017	131733.5
141128.6	2634.390	143763.0

Figure 5 - Regression with AR(1) forecast

The forecast can be seen in the following plot visually.



Q3-Use ARIMA Model and Compare Various Methods.

3a-Use Arima () function to fit ARIMA (1,1,1) (1,1,1) model for the training data set. Insert in your report the summary of this ARIMA model, present and briefly explain the ARIMA model and its equation in your report. Using this model, forecast revenue for the validation period and present it in your report.

The ARIMA (1,1,1) (1,1,1) model denotes a seasonal Arima model by including additional seasonal terms in the Arima (p, d, q) model.
The seasonal Arima model consists of the following parameters.

ARIMA (1, 1, 1) (1, 1, 1)$_{12}$ means the following:
p = 1, order 1 autoregressive model *AR (1)*
d = 1, order 1 differencing to remove linear trend
q = 1, order 1 moving average *MA (1)* for error lags
P = 1, order 1 autoregressive model *AR (1) for seasonality*
D = 1, order 1 differencing to remove linear trend
Q = 1, order 1 moving average *MA (1)* for error lags
m = 4, for quarterly seasonality

```
> summary(train.arima.seas)
Series: train.ts
ARIMA(1,1,1)(1,1,1)[4]

Coefficients:
          ar1     ma1    sar1     sma1
      -0.7185  0.6619  0.2714  -0.8280
s.e.   0.4351  0.4472  0.2707   0.2579

sigma^2 estimated as 3372999:  log likelihood=-365.59
AIC=741.18   AICc=742.89   BIC=749.75

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE        ACF1
Training set -378.6898 1647.138 1180.11 -0.3593132 1.101027 0.2703869 -0.02746972
>
```

Figure 6 - Summary output in R

Fitting the Arima model with our training data using the following R code.

```
train.arima.seas <- Arima(train.ts, order = c(1,1,1),seasonal = c(1,1,1))
summary(train.arima.seas)
```

The model equation based on the above parameters would be:

$Y_t - y_{t-1} = -0.7185(y_{t-1} - y_{t-2}) + 0.6619\varepsilon_{t-1} + 0.2714(y_{t-1} - y_{t-5}) - 0.8280P_{t-2}$

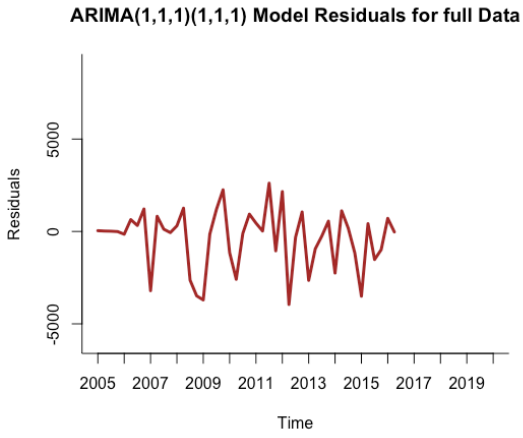
Forecasting the validation period based on the above model is as follows:

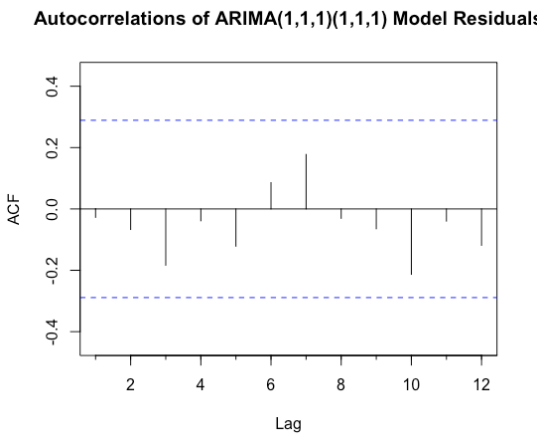
```
train.arima.seas.pred <- forecast(train.arima.seas, h = nValid,
                                  level = 0)
formattable(data.frame(train.arima.seas.pred))
```

	Point.Forecast	Lo.0	Hi.0
2016 Q3	118951.6	118951.6	118951.6
2016 Q4	131784.7	131784.7	131784.7
2017 Q1	118313.5	118313.5	118313.5
2017 Q2	123145.9	123145.9	123145.9
2017 Q3	121503.4	121503.4	121503.4
2017 Q4	134484.7	134484.7	134484.7
2018 Q1	121098.2	121098.2	121098.2
2018 Q2	125894.7	125894.7	125894.7
2018 Q3	124325.6	124325.6	124325.6
2018 Q4	137345.1	137345.1	137345.1
2019 Q1	123983.0	123983.0	123983.0
2019 Q2	128768.7	128768.7	128768.7
2019 Q3	127220.3	127220.3	127220.3
2019 Q4	140249.5	140249.5	140249.5
2020 Q1	126894.5	126894.5	126894.5
2020 Q2	131677.0	131677.0	131677.0

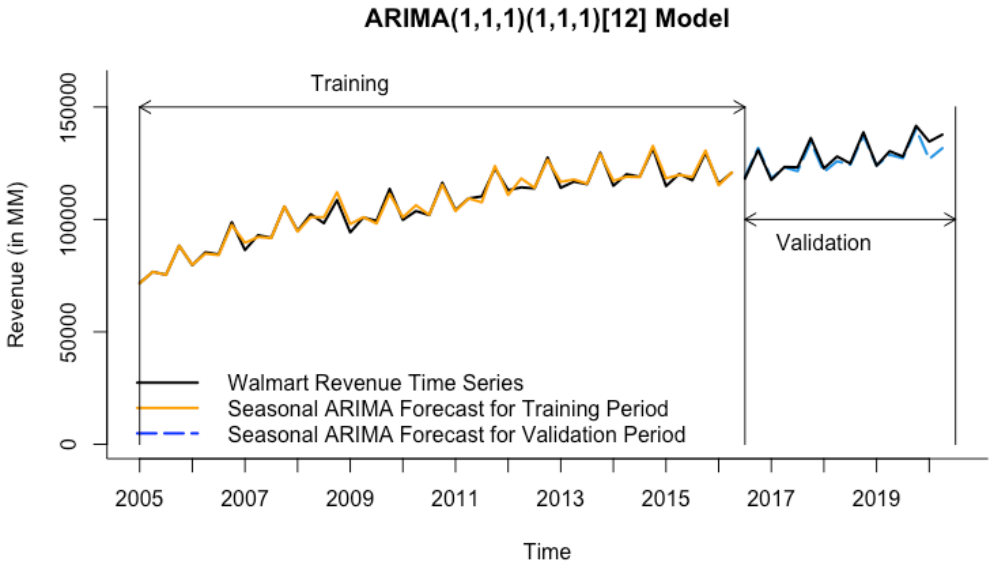
Figure 7 - Validation forecast Arima(1,1,1)(1,1,1)

The autocorrelation Acf() function and the residual plot for the residuals after applying the Arima(1,1,1)(1,1,1) model is as follows.





From the correlogram for the residuals of the Arima (1,1,1) (1,1,1) model, we can see that there are no statistically significant autocorrelation coefficients in any of the lag. This shows that the Arima (1,1,1) (1,1,1) model has efficiently captured all the data patterns and seasonality. Hence, we cannot see any significant data patterns left. We just see the random variations which cannot be incorporated in the forecasting.



The validation forecast can be visually seen in the above plot and is very close to the actual validation data. This shows that the model captures the data patterns efficiently.

3b- Use the auto.arima() function to develop an ARIMA model using the training data set. Insert in your report the summary of this ARIMA model, present and explain the ARIMA model and its equation in your report. Use this model to forecast revenue in the validation period and present this forecast in your report.

The auto.arima() model in R is used to identify optimal Arima model and the parameters. It does require the used to explicitly mention any parameter inputs.

Fitting the auto.arima() with the training data set using the following R code,

train.auto.arima <- auto.arima(train.ts)

summary(train.auto.arima)

```
> summary(train.auto.arima)
Series: train.ts
ARIMA(0,1,0)(0,1,1)[4]

Coefficients:
         sma1
      -0.6284
s.e.   0.2022

sigma^2 estimated as 3334870:  log likelihood=-366.58
AIC=737.16   AICc=737.48   BIC=740.59

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE       ACF1
Training set -343.5752 1702.905 1257.028 -0.3306204 1.167923 0.2880103 -0.1496364
>
```

Figure 8- Summary output in R

The model equation based on the above parameters would be:

$$Y_t - y_{t-1} = -0.6284\rho_{t-1}$$

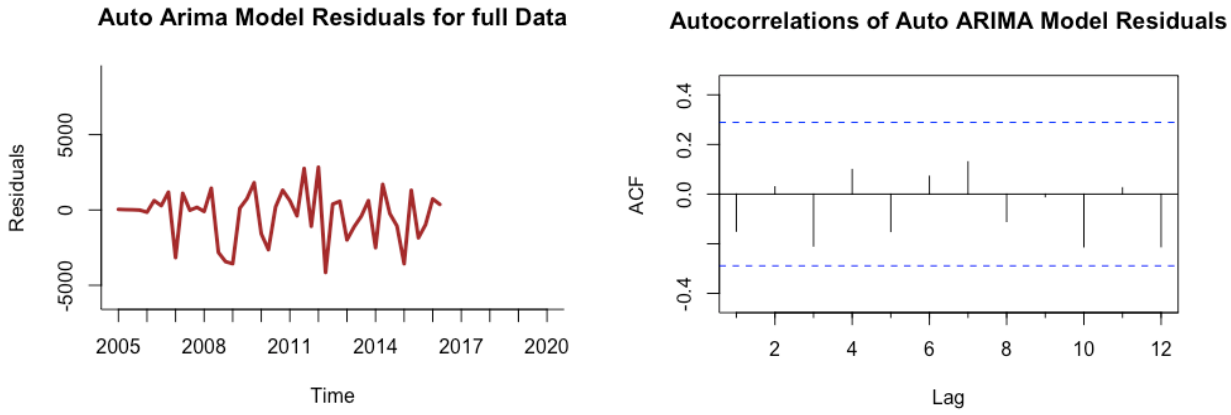
Forecasting the validation period based on the above model is as follows:

train.auto.arima.pred <- forecast(train.auto.arima, h = nValid, level = 0)
formattable(data.frame(train.auto.arima.pred))

	Point.Forecast	Lo.0	Hi.0
2016 Q3	119195.7	119195.7	119195.7
2016 Q4	132066.2	132066.2	132066.2
2017 Q1	117841.4	117841.4	117841.4
2017 Q2	122559.5	122559.5	122559.5
2017 Q3	120901.2	120901.2	120901.2
2017 Q4	133771.6	133771.6	133771.6
2018 Q1	119546.8	119546.8	119546.8
2018 Q2	124264.9	124264.9	124264.9
2018 Q3	122606.6	122606.6	122606.6
2018 Q4	135477.1	135477.1	135477.1
2019 Q1	121252.3	121252.3	121252.3
2019 Q2	125970.4	125970.4	125970.4
2019 Q3	124312.1	124312.1	124312.1
2019 Q4	137182.6	137182.6	137182.6
2020 Q1	122957.8	122957.8	122957.8
2020 Q2	127675.9	127675.9	127675.9

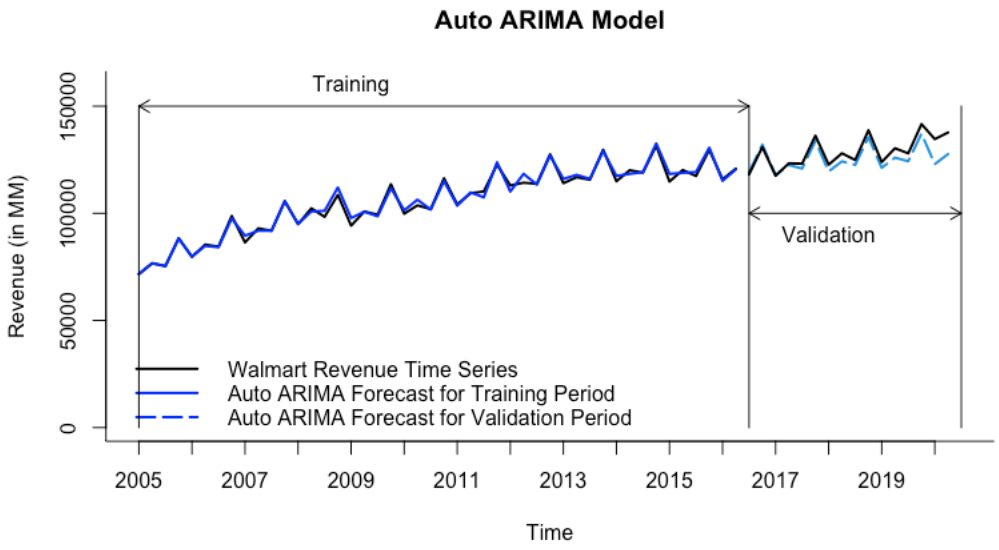
Figure 9 - Auto arima validation forecast

The autocorrelation Acf() function and the residual plot for the residuals after applying the Auto arima model is as follows.



From the correlogram for the residuals of the auto Arima model, we can see that there are no statistically significant autocorrelation coefficients in any of the lag. This shows that the auto Arima model has efficiently captured all the data patterns and seasonality. Hence, we cannot see any significant data patterns left. We just see the random variations which cannot be incorporated in the forecasting.

The validation forecast can be visually seen in the below plot and is very close to the actual validation data. This shows that the model captures the data patterns efficiently.



3c- Apply the accuracy() function to compare performance measures of the two ARIMA models in 3a and 3b. Present the accuracy measures in your report, compare them and identify, using MAPE and RMSE, the best ARIMA model to apply.

Though the forecasts of the Arima(1,1,1) (1,1,1) and auto Arima models look very similar, the accuracy() is the best tool to identify the best model.

The accuracy () is as follows for the 2 models from 3a and 3b:

```
> #accuracy for 3a - ARIMA(1,1,1)(1,1,1)
> round(accuracy(train.arima.seas.pred, valid.ts), 3)
                    ME     RMSE      MAE    MPE  MAPE  MASE   ACF1 Theil's U
Training set -378.690 1647.138 1180.11 -0.359 1.101 0.270 -0.027        NA
Test set     1534.403 2728.866 1840.75  1.142 1.393 0.422  0.501     0.267
> #accuracy for 3b - Auto ARIMA
> round(accuracy(train.auto.arima.pred, valid.ts), 3)
                    ME     RMSE      MAE    MPE  MAPE  MASE   ACF1 Theil's U
Training set -343.575 1702.905 1257.028 -0.331 1.168 0.288 -0.150        NA
Test set     3288.054 4692.662 3593.830  2.476 2.724 0.823  0.634     0.466
>
```

Figure 10 - Summary output in R

MAPE & RMSE comparison for the 2 models:

Model	MAPE	RMSE
Arima (1,1,1) (1,1,1)	Training: 1.101 Validation: 1.393	Training: 1647.14 Validation: 2728.86
Auto Arima	Training: 1.168 Validation: 2.724	Training: 1702.905 Validation: 4692.662

Based on the above results, we can see that the Arima (1,1,1) (1,1,1) model has better MAPE and RMSE profile. Though the parameters of the Auto Arima models are automatically chosen by R, it does not always guarantee the best results in terms of forecast error profiles. It may require some amount of effort to identify the best parameters for the model but following some best practices like visualizing the historical data, applying Acf () may help narrowing down the selection. Arima (1,1,1) (1,1,1) is the best model in terms of MAPE and RMSE

3d-Use two ARIMA models from 3a and 3b for the entire data set. Present models' summaries in your report. Use these ARIMA models to forecast Walmart revenue in 2020- 2021 and present these forecasts in your report.

Arima (1,1,1)(1,1,1) model using the entire data set:

Fitting the Arima model with our entire data using the following R code.

full.arima.seas <- Arima(revenue.ts, order = c(1,1,1),seasonal = c(1,1,1))

summary(full.arima.seas)

```
> full.arima.seas <- Arima(revenue.ts, order = c(1,1,1),seasonal = c(1,1,1))
> summary(full.arima.seas)
Series: revenue.ts
ARIMA(1,1,1)(1,1,1)[4]

Coefficients:
          ar1     ma1    sar1     sma1
      -0.5144  0.4076  0.2251  -1.0000
s.e.   0.6745  0.6937  0.1674   0.1504

sigma^2 estimated as 3153532:  log likelihood=-509.86
AIC=1029.72   AICc=1030.9   BIC=1039.94

Training set error measures:
                    ME     RMSE  MAE        MPE      MAPE      MASE        ACF1
Training set -283.7691 1641.877 1104 -0.2780365 0.9832904 0.2651396 -0.03523734
>
```

Figure 11- Summary output in R

The model equation based on the summary for the Arima (1,1,1)(1,1,1) model is as follows.

$$Y_t - y_{t-1} = -0.5144y_{t-1} + 0.4076\varepsilon_{t-1} + 0.2551P_{t-1} - 1.00\rho_{t-1}$$

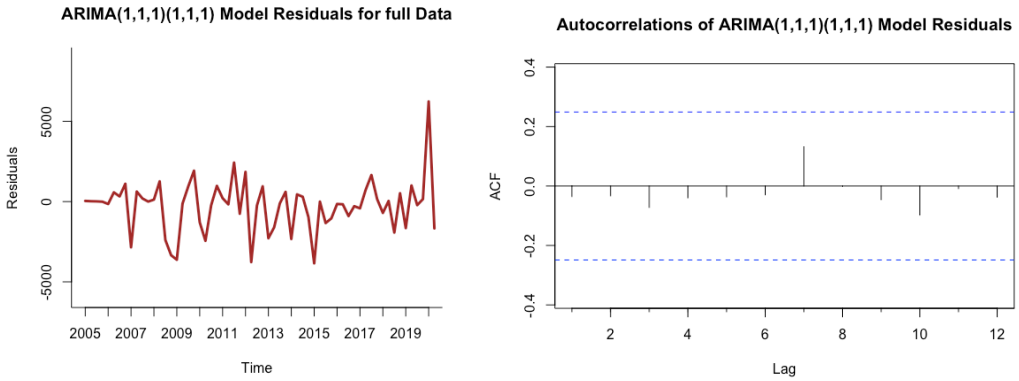
Forecasting the 2020-2021 period based on the above model is as follows:

full.arima.seas.pred <- forecast(full.arima.seas, h = 6, level = 0)
formattable(data.frame(full.arima.seas.pred))

	Point.Forecast	Lo.0	Hi.0
2020 Q3	136482.7	136482.7	136482.7
2020 Q4	149557.6	149557.6	149557.6
2021 Q1	138457.2	138457.2	138457.2
2021 Q2	142974.5	142974.5	142974.5
2021 Q3	141524.1	141524.1	141524.1
2021 Q4	154691.6	154691.6	154691.6

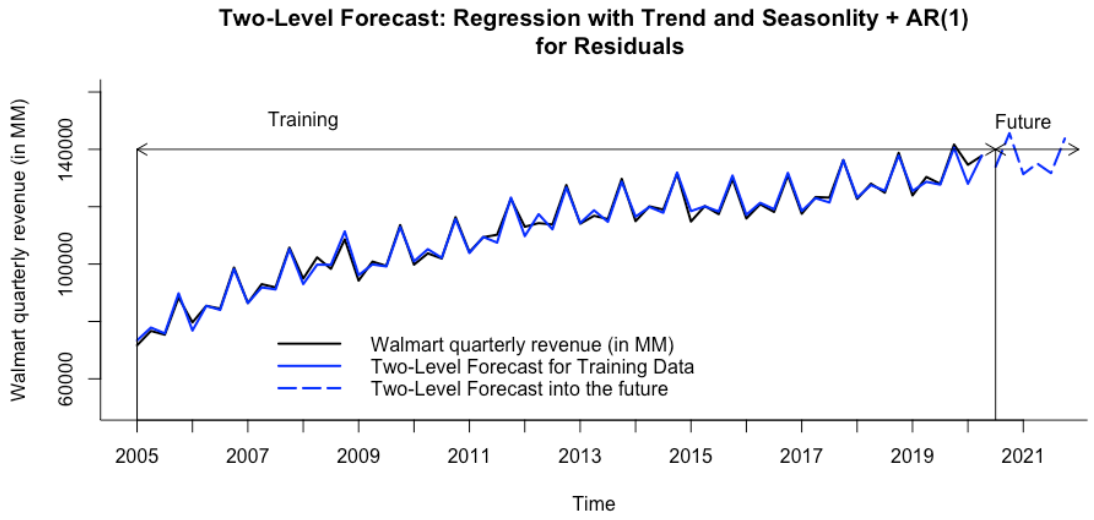
Figure 12 - Arima (1,1,1)(1,1,1) 2020-2021 forecast

The autocorrelation Acf() function and the residual plot for the residuals after applying the Arima(1,1,1)(1,1,1) model is as follows.



From the correlogram for the residuals of the Arima (1,1,1) (1,1,1) model, we can see that there are no statistically significant autocorrelation coefficients in any of the lag. This shows that the Arima (1,1,1) (1,1,1) model has efficiently captured all the data patterns and seasonality. Hence, we cannot see any significant data patterns left. We just see the random variations which cannot be incorporated in the forecasting.

The 2020-2021 forecast can be visually seen in the below plot and based on the Acf() on the residuals it is safe to say that the model captures the data patterns efficiently.



Auto Arima model using the entire data set:

Fitting the Arima model with our entire data using the following R code.

full.auto.arima <- auto.arima(revenue.ts)
summary(full.auto.arima)

```
> summary(full.auto.arima)
Series: revenue.ts
ARIMA(0,1,0)(1,1,0)[4]

Coefficients:
          sar1
       -0.3858
s.e.    0.1434

sigma^2 estimated as 4083367:  log likelihood=-514.53
AIC=1033.07   AICc=1033.29   BIC=1037.15

Training set error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE       ACF1
Training set -45.03607 1920.469 1380.755 -0.07042617 1.222555 0.3316057 -0.160586
>
```

Figure 13- Summary output in R

The model equation based on the above parameters would be:

$$Y_t - Y_{t-1} = -0.3858\rho_{t-1}$$

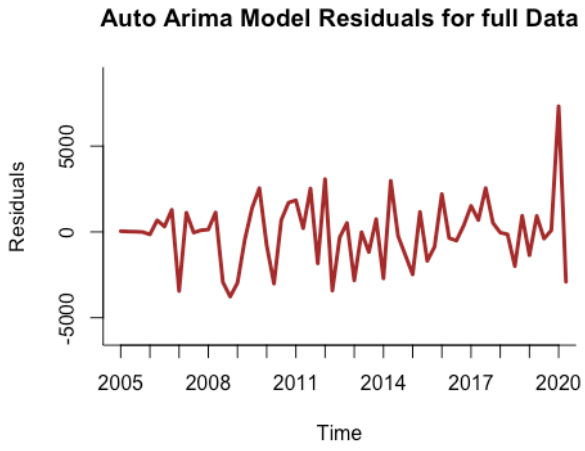
Forecasting the 2020-2021 period based on the above model is as follows:

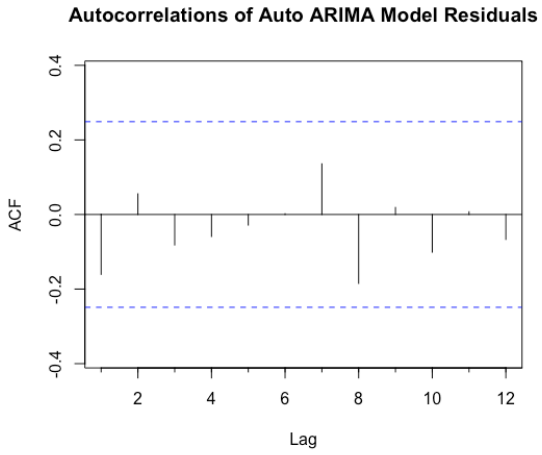
full.auto.arima.pred <- forecast(full.auto.arima, h = 6, level = 0)
full.auto.arima.pred

	Point.Forecast	Lo.0	Hi.0
2020 Q3	135067.4	135067.4	135067.4
2020 Q4	148831.9	148831.9	148831.9
2021 Q1	138766.4	138766.4	138766.4
2021 Q2	143171.9	143171.9	143171.9
2021 Q3	140608.6	140608.6	140608.6
2021 Q4	154340.5	154340.5	154340.5

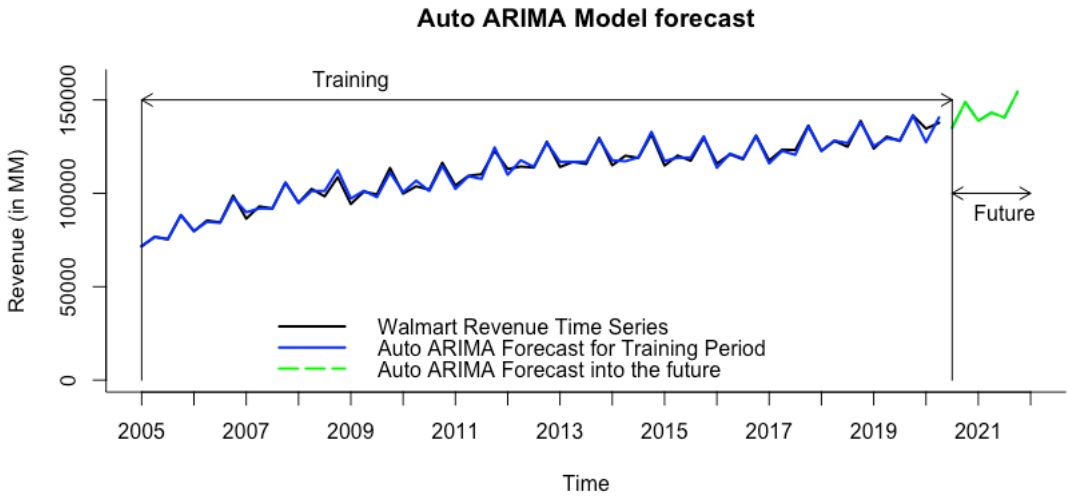
Figure 14 - Auto arima forecast for 2020-2021

The autocorrelation Acf() function and the residual plot for the residuals after applying the Auto Arima model is as follows





From the correlogram for the residuals of the auto Arima model, we can see that there are no statistically significant autocorrelation coefficients in any of the lag. This shows that the auto Arima model has efficiently captured all the data patterns and seasonality. Hence, we cannot see any significant data patterns left. We just see the random variations which cannot be incorporated in the forecasting.



The 2020-2021 forecast can be visually seen in the above plot and based on the Acf () on the residuals it is safe to say that the model captures the data patterns efficiently.

3e- Apply the accuracy() function to compare performance measures of the following forecasting models for the entire data set: (1) regression model with quadratic trend and seasonality; (2) two-level model (with AR(1) model for residuals); (3) ARIMA(1,1,1)(1,1,1) model; (4) auto ARIMA model; and (5) seasonal naïve forecast for the entire data set. Present the accuracy measures in your report, compare them, and identify, using MAPE and RMSE, the best model to use for forecasting Walmart's revenue in quarters 3 and 4 of 2020 and quarters 1 and 2 of 2021.

Comparing the error profiles using the accuracy().

```
> #(1) regression model with quadratic trend and seasonality;
> round(accuracy(full.Quadtrend.season.pred$fitted, revenue.ts), 3)
         ME     RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set  0 2729.216 2077.481 -0.076 1.888 0.755     0.282
>
> #(2) two-level model (with AR(1) model for residuals);
> round(accuracy( trend.season.pred$fitted + residual.ar1.pred$fitted, revenue.ts), 3)
             ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 65.319 1619.311 1181.396 0.032 1.083 0.005     0.161
>
> #(3) ARIMA(1,1,1)(1,1,1) model;
> round(accuracy(full.arima.seas.pred$fitted, revenue.ts), 3)
              ME     RMSE  MAE    MPE  MAPE   ACF1 Theil's U
Test set -283.769 1641.877 1104 -0.278 0.983 -0.035     0.162
>
> #(4) auto ARIMA model
> round(accuracy(full.auto.arima.pred$fitted, revenue.ts), 3)
              ME     RMSE      MAE   MPE  MAPE   ACF1 Theil's U
Test set -45.036 1920.469 1380.755 -0.07 1.223 -0.161      0.19
>
> #(5) seasonal naïve forecast for the entire data set
> round(accuracy((snaive(revenue.ts))$fitted, revenue.ts), 3)
              ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 3964.224 5074.236 4163.845 3.696 3.873 0.753     0.558
>
```

Model	MAPE	RMSE
Reg. Quadratic trend and seasonality	1.888	2729.216
Two-level model with AR (1) for residuals	1.083	1619.311
Arima (1,1,1) (1,1,1)	**0.983**	**1641.877**
Auto Arima	1.223	1920.469
Seasonal Naive	3.873	5074.236

The MAPE (Mean Absolute Percent Error) measures the size of the error in percentage terms. When it comes to forecasting, it is easier to understand the error given that MAPE gives the percentage of how much the actuals and the forecast were off from one another. While RMSE is a measure of how spread out the residuals are or how concentrated the data is around the line of best fit.

Comparing the RMSE and MAPE of the different models we have applied on the Walmart quarterly revenue time series data, we can see that Two-level model with AR (1) for residuals and Arima (1,1,1) (1,1,1) come very close to one another. If you are going by MAPE, Arima (1,1,1) (1,1,1) is the best model. While in terms of RMSE Two-level model with AR (1) for residuals have the lowest.

In my opinion, the Arima (1,1,1) (1,1,1) model is the best fit for the Walmart quarterly revenue forecasting because of the lowest MAPE and also a good RMSE value under overall comparison with the rest of the models. The second-best model is the Two-level model with AR (1) for residuals.