David Liu
BAN 673

Case Study #3: Forecasting with AR and ARIMA Models

Consider the quarterly data on Walmart revenues (in $million) from the first quarter of 2005 through the second quarter of 2020 (673_case2.csv). The goal is to forecast Walmart's quarterly revenue for the next two quarters of 2020 and the first two quarters of 2021.

As you did in case study #2, start this case with the following:

- Create time series data set in R using the ts() function.
- Develop data partition with the validation partition of 16 periods and the rest for the training partition.

Questions

1. Identify time series predictability.
   a. 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.

```
walmart.ar1 <- Arima(walmart.ts, order = c(1,0,0))
summary(walmart.ar1)
```

Above, codes for the AR(1) model

```
> summary(walmart.ar1)
Series: walmart.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
```

Above, summary of the AR(1) model

The model has an autocorrelation coefficient of 0.8697 and an intercept of 110533.61. Altogether, it forms an equation as follows:

$$y_t = 0.1110533.61 + 0.8697y_{t-1}$$

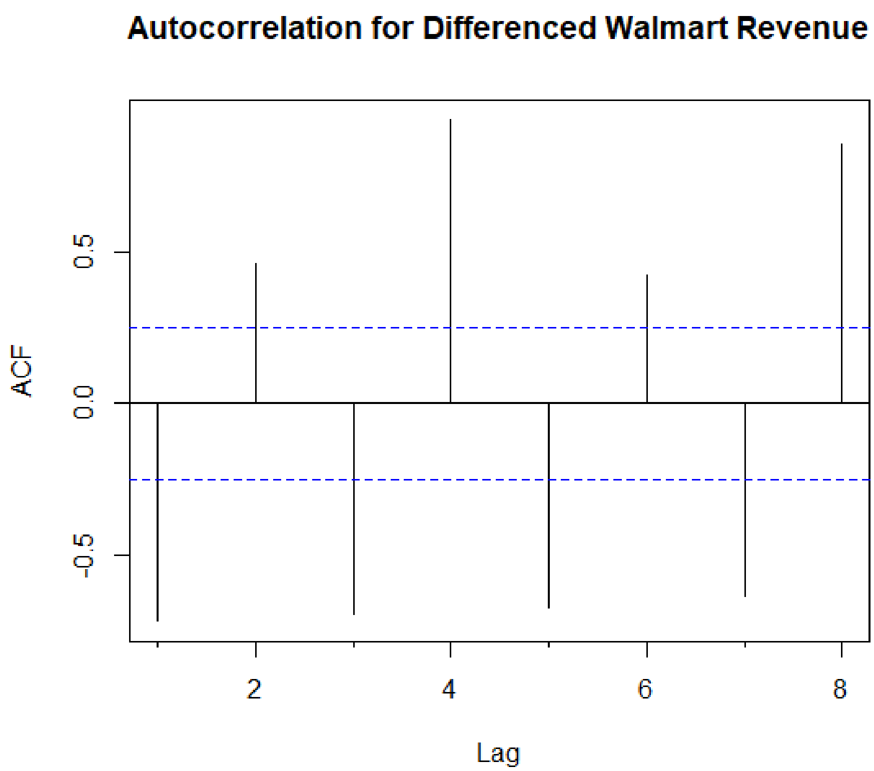
where y(t) is the response variable and y(t-1) is the predictor with period lag t-1. The autocorrelation coefficient of 0.8697 is somewhat large and close to 1, indicating that there is a possibility of random walk. In addition, the standard error of 0.0702 indicates that there is a 95% confidence that the autocorrelation coefficient may be anywhere as high as approximately 1 or as low as approximately 0.7, both of which are relatively high numbers. Overall, the order 1

ARIMA model suggests that the historical Walmart revenue data might not be strongly predictable.

b. 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.

```
diff.walmart <- diff(walmart.ts, lag = 1)
diff.walmart

diff.walmart.acf <- Acf(diff.walmart, lag.max = 8,
                        main = "Autocorrelation for Differenced Walmart Revenue")
diff.walmart.acf
```

Above, codes for differencing the historical Walmart revenue data with time lag 1, and then plotting the autocorrelation plot all the way to a maximum of 8 lags.



Above, the autocorrelation plot of differencing with a lag of 1 to 8.

Shown in the autocorrelation plot above, all autocorrelation values for time lags 1 to 8 are high enough to be consider somewhat significant. This suggests that the Walmart revenue data is very much predictable.

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

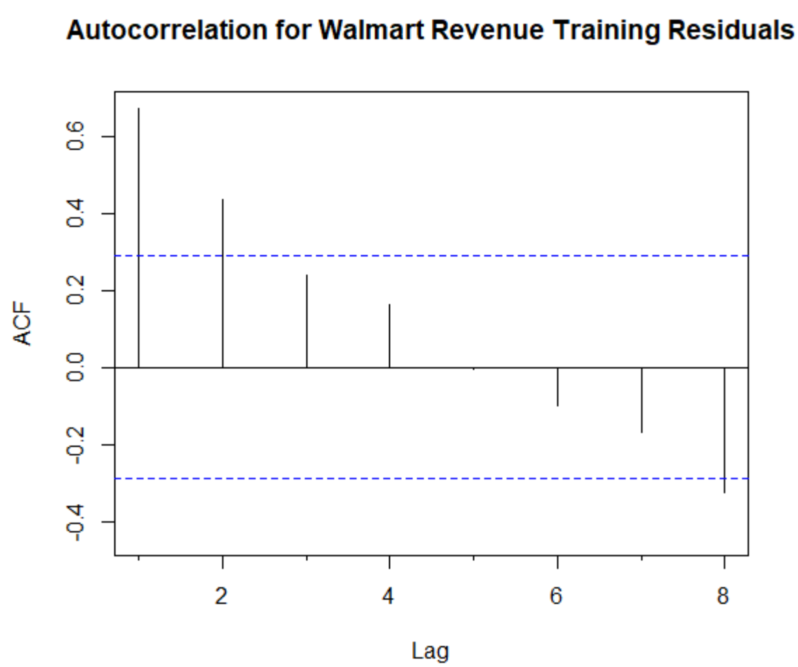
```
walmart.reg.quad.seas <- tslm(walmart.train.ts ~ trend + I(trend^2) + season)

walmart.reg.quad.seas.pred <- forecast(walmart.reg.quad.seas, h = 16, level = 0)
walmart.reg.quad.seas.pred
```

Above, codes for creating the tslm model with quadratic trend and seasonality, using the Walmart revenue training data, to forecast 16 quarters.

    b. 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.

```
walmart.train.residual <- walmart.reg.quad.seas.pred$residuals

walmart.train.residual.acf <- Acf(walmart.train.residual, lag.max = 8,
                                  main = "Autocorrelation for Walmart Revenue Training Residuals")
```

Above, codes for identifying the residuals in the training period, and using the Acf() function to identify the autocorrelation for the residuals with 8 time lags.



Above, plot for showing the autocorrelation for the Walmart revenue training residuals.

The above plot shows that there is significant autocorrelation at time lag 1, 2, and 8 for the training set residuals. Since the autocorrelations for the residuals are significant at certain time lags, it would be a good idea to use an AR model for forecasting residuals.

c. 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.

```
walmart.train.residual.ar1 <- Arima(walmart.train.residual, order = c(1,0,0))
summary(walmart.train.residual.ar1)
```

Above, codes for building the ar1 model using the residuals of the training data regression.

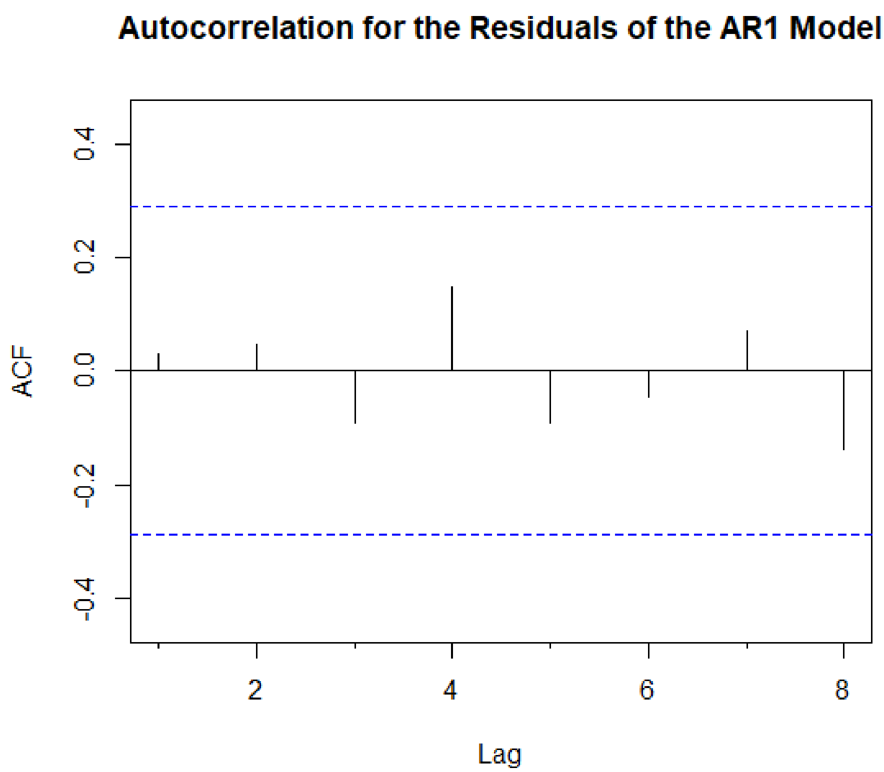
```
Series: walmart.train.residual
ARIMA(1,0,0) with non-zero mean

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
```

Above shows the summary of the ar1 model.

The model has an autocorrelation coefficient of 0.6564 and an intercept of -18.1858. Altogether, it forms an equation as follows:

$$y_t = -18.1858 + 0.6564y_{t-1}$$

where y(t) is the response variable and y(t-1) is a predictor with a lag value of -1. The autocorrelation coefficient of 0.6564 with an s.e. of 0.1067 is not very large, indicating that the training residual data is not quite considered random walk. The order 1 ARIMA model summary suggests that the training data residuals can be somewhat predictable.



Above shows the autocorrelation for the residuals of the AR1 model, which was built using the residuals of the previous training data regression model. Shown by the autocorrelation plot, none of the time lags from 1 to 8 shows any value of significance. This implies that the residuals of the AR1 model may be predominantly random walk in nature, and will not be well forecasted.

d. 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.

```
walmart.train.residual.ar1.pred <- forecast(walmart.train.residual.ar1, h = 16, level = 0)
walmart.train.residual.ar1.pred

walmart.2level <- walmart.reg.quad.seas.pred$mean + walmart.train.residual.ar1.pred$mean
walmart.2level
```

Above, using the residual AR(1) model to forecast the residuals for the 16 validation data, and then combining the residual forecasts with the TSLM model predictions to create a two-level forecasting model.

```
walmart.train.ar1 <- Arima(walmart.train.ts, order = c(1,0,0))
walmart.train.ar1.pred <- forecast(walmart.train.ar1, h=16, level = 0)
walmart.train.ar1.pred

val.table <- data.frame(walmart.valid.ts, walmart.reg.quad.seas.pred$mean, walmart.train.residual.ar1.pred$mean,
                        walmart.train.ar1.pred$mean, walmart.2level)
names(val.table) <- c("Walmart Val", "TSLM Forecast", "AR(1) Residual Forecast", "AR(1) Forecast", "Two Level")

val.table
```

Above codes for using an AR1 model to predict 16 quarters Walmart revenue using the Walmart revenue training data. All validation forecasts from the TSLM model, the AR(1) residual model, the AR(1) model, and the 2 level model, as well as the validation data are displayed in a table.

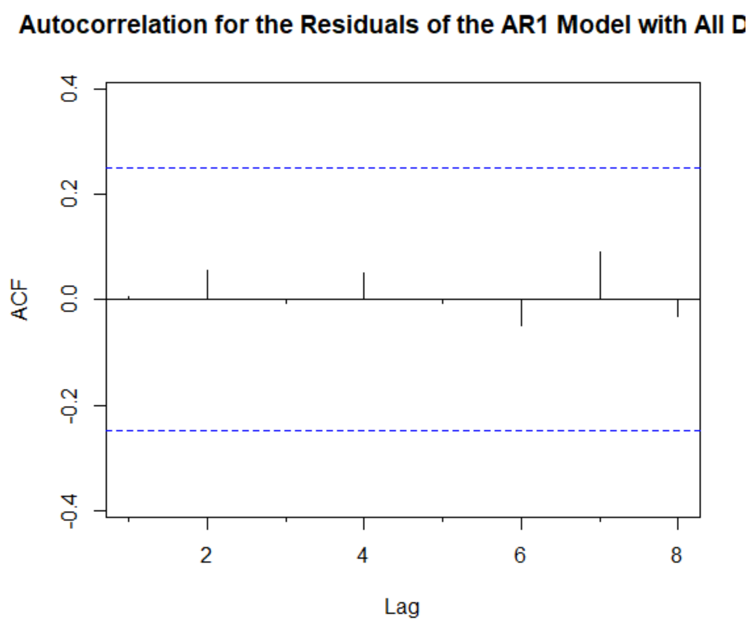
	Walmart Val	TSLM Forecast	AR(1) Residual Forecast	AR(1) Forecast	Two Level
1	118179	118979.2	-297.42701	117927.4	118681.8
2	130936	131214.6	-201.48930	115515.1	131013.1
3	117542	117466.4	-138.51251	113526.7	117327.9
4	123355	121429.0	-97.17238	111887.7	121331.8
5	123179	118950.5	-70.03531	110536.7	118880.5
6	136267	131025.8	-52.22160	109423.1	130973.6
7	122690	117117.5	-40.52808	108505.2	117077.0
8	128028	120920.0	-32.85205	107748.5	120887.2
9	124894	118281.4	-27.81324	107124.9	118253.6
10	138793	130196.6	-24.50559	106610.8	130172.1
11	123925	116128.3	-22.33434	106187.0	116106.0
12	130377	119770.7	-20.90905	105837.8	119749.8
13	127991	116972.0	-19.97345	105549.9	116952.1
14	141671	128727.1	-19.35929	105312.5	128707.8
15	134622	114498.7	-18.95613	105116.9	114479.8
16	137742	117981.0	-18.69148	104955.7	117962.3

Above, the table.

e. 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).

```
walmart.reg.quad.seas.alldata <- tslm(walmart.ts ~ trend + I(trend^2) + season)
walmart.reg.quad.seas.alldata.pred <- forecast(walmart.reg.quad.seas.alldata, h = 6, level = 0)
walmart.reg.quad.seas.alldata.pred

walmart.residual.alldata <- walmart.reg.quad.seas.alldata.pred$residuals
walmart.residual.alldata.ar1 <- Arima(walmart.residual.alldata, order=c(1,0,0))
walmart.residual.alldata.ar1.pred <- forecast(walmart.residual.alldata.ar1, h=6, level=0)
walmart.residual.alldata.ar1.acf <- Acf(walmart.residual.alldata.ar1.pred$residuals, lag.max = 8,
                                        main = "Autocorrelation for the Residuals of the AR1 Model with All Data")
walmart.residual.alldata.ar1.acf
```

Above, codes for the AR(1) model for forecasting residuals, and an autocorrelation chart of the AR(1) model.



Plot of the autocorrelation for the AR(1) residual model. As shown by the plot above, none of the 8 time lags has a significant autocorrelation value for the residuals of the AR(1) residual model. This implies that the residuals are predominantly random, and will not be good for forecasting.

```
walmart.reg.quad.seas.alldata <- tslm(walmart.ts ~ trend + I(trend^2) + season)
walmart.reg.quad.seas.alldata.pred <- forecast(walmart.reg.quad.seas.alldata, h = 6, level = 0)
walmart.reg.quad.seas.alldata.pred

walmart.residual.alldata <- walmart.reg.quad.seas.alldata.pred$residuals
walmart.residual.alldata.ar1 <- Arima(walmart.residual.alldata, order=c(1,0,0))
walmart.residual.alldata.ar1.pred <- forecast(walmart.residual.alldata.ar1, h=6, level=0)
walmart.residual.alldata.ar1.acf <- Acf(walmart.residual.alldata.ar1.pred$residuals, lag.max = 8,
                                        main = "Autocorrelation for the Residuals of the AR1 Model with All Data")
walmart.residual.alldata.ar1.acf

walmart.2level.alldata <- walmart.reg.quad.seas.alldata.pred$mean + walmart.residual.alldata.ar1.pred$mean
walmart.2level.alldata

val.table.alldata <- data.frame(walmart.reg.quad.seas.alldata.pred$mean,
                                walmart.residual.alldata.ar1.pred$mean, walmart.2level.alldata )
names(val.table.alldata) <- c("TSLM Forecast All", "AR(1) Residual Forecast All", "Two Level All")

val.table.alldata
```

Above, all codes relating to creating the LSTM, AR(1) residual model, and two level models using all Walmart revenue data.

	TSLM Forecast All	AR(1) Residual Forecast All	Two Level All
1	128087.3	5870.718	133958.0
2	140645.4	4981.396	145626.8
3	127141.8	4234.004	131375.8
4	131466.4	3605.889	135072.3
5	128655.5	3078.017	131733.5
6	141128.6	2634.390	143763.0

The table to display the forecasts from 2020-2021 using the TSLM, AR(1), and Two level models.

3. Use ARIMA Model and Compare Various Methods.
    a. 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.

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

walmart.ar.seas.pred <- forecast(walmart.ar.seas, h=16, level=0)
walmart.ar.seas.pred
```

Above, codes for creating the ARIMA(1,1,1)(1,1,1) model using the Walmart training data, and then using the model to predict 16 validation periods.

```
Coefficients:
          ar1     ma1    sar1     sma1
      -0.7186  0.6619  0.2714  -0.8280
s.e.   0.4351  0.4471  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 -378.6899 1647.138 1180.11 -0.3593133 1.101027 0.2703869 -0.02746983
```

Above, summary of the ARIMA(1,1,1)(1,1,1) model. The first set of (1,1,1) represents an order 1 ARIMA model with order 1 differencing to remove linear trend and an order 1 moving average for error lags. The second set of (1,1,1) handles seasonality for the model. It represents an order 1 ARIMA model for seasonality with an order 1 differencing to remove linear trend and an order 1 moving average for error lags. Altogether, it creates the following equation:

$$y_t - y_{t-1} = -0.7186(y_{t-1} - y_{t-2}) + 0.6619e_{t-1} + 0.2714(y_{t-1} - y_{t-5}) - 0.8280p_{t-1}$$

where the output is (y(t) – y(t-1)) . In the equation, y is the predictor values, t is the time periods, e is the error terms for the unexplained portions of the response variable, and p is a second set of error terms for the unexplained portions of the response variable that was calculated as a result of having seasonal terms in the model.

```
walmart.ar.seas.pred <- forecast(walmart.ar.seas, h=16, level=0)
walmart.ar.seas.pred
```

Codes above for using the ARIMA(1,1,1)(1,1,1) model to forecast the validation period.

David Liu
BAN 673

```
        Point Forecast
2016 Q3       118951.6
2016 Q4       131784.7
2017 Q1       118313.5
2017 Q2       123145.9
2017 Q3       121503.4
2017 Q4       134484.7
2018 Q1       121098.2
2018 Q2       125894.7
2018 Q3       124325.6
2018 Q4       137345.1
2019 Q1       123983.0
2019 Q2       128768.8
2019 Q3       127220.3
2019 Q4       140249.6
2020 Q1       126894.5
2020 Q2       131677.1
```

Above, shows the forecasted revenues for the validation period.

b. 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.

```
walmart.ar.seas.auto <- auto.arima(walmart.train.ts)
summary(walmart.ar.seas.auto)
```

Above, codes for creating the auto ARIMA model using the Walmart training data.

```
Series: walmart.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
```

Above is the summary for the auto ARIMA model. The auto.arima() function automatically created an ARIMA(0,1,0)(0,1,1) model. More specifically, the model is a 0 order model with an order 1 differencing to remove linear trend and a 0 order moving average for error lag. As for seasonality, it is an order 0 model for seasonality with an order 1 differencing to remove linear trend and an order 1 moving average for error lags. Altogether, it creates the following equation:

$$y_t – y_{t-1} = -0.6284p_{t-1}$$

where the output is (y(t) – y(t-1)) for having and order 1 differencing, and where p represents error terms for the unexplained portions of the response variable that was calculated as a result of having seasonal terms in the model.

```
walmart.ar.seas.auto.pred <- forecast(walmart.ar.seas.auto, h=16, level=0)
walmart.ar.seas.auto.pred
```

Above, codes for using the auto ARIMA model to predict for the validation period.

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

Above, predictions of the auto ARIMA model for the validation period.

c. 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.

```
walmart.ar.seas.acc <- accuracy(walmart.ar.seas.pred, walmart.valid.ts)
walmart.ar.seas.acc

walmart.ar.seas.auto.acc <- accuracy(walmart.ar.seas.auto.pred, walmart.valid.ts)
walmart.ar.seas.auto.acc

labels <- c("RMSE", "MAPE")
AR_Model <- c(walmart.ar.seas.acc[4], walmart.ar.seas.acc[10])
auto_AR_Model <- c(walmart.ar.seas.auto.acc[4], walmart.ar.seas.auto.acc[10])
walmart.acc.table <- data.frame(labels, AR_Model, auto_AR_Model )
walmart.acc.table
```

Above, codes for the accuracies of the ARIMA(1,1,1)(1,1,1) model and the auto ARIMA model.

```
  labels    AR_Model auto_AR_Model
1   RMSE 2728.859983   4692.662110
2   MAPE    1.392941      2.723536
```

Above, table of the RMSE and MAPE for both models. Using only RMSE and MAPE, the ARIMA(1,1,1)(1,1,1) model is the better model.

d. 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.

```
walmart.ar.seas.alldata <- Arima(walmart.ts , order = c(1,1,1), seasonal = c(1,1,1))
summary(walmart.ar.seas.alldata)

walmart.ar.seas.alldata.pred <- forecast(walmart.ar.seas.alldata, h=6, level=0)
walmart.ar.seas.alldata.pred


walmart.ar.seas.auto.alldata <- auto.arima(walmart.ts)
summary(walmart.ar.seas.auto.alldata)

walmart.ar.seas.auto.alldata.pred <- forecast(walmart.ar.seas.auto.alldata, h=6, level=0)
walmart.ar.seas.auto.alldata.pred
```

Above, codes for building the ARIMA(1,1,1)(1,1,1) and auto ARIMA model.

```
Series: walmart.ts
ARIMA(1,1,1)(1,1,1)[4]

Coefficients:
          ar1     ma1    sar1     sma1
      -0.5144  0.4076  0.2251  -1.0000
s.e.   0.6744  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 -0.03523737
```

Above, shows the summary of the ARIMA(1,1,1)(1,1,1) model. The equation for this model is the following:

$$y_t - y_{t-1} = -0.5144\,(y_{t-1} - y_{t-2}) + 0.4076e_{t-1} + 0.2251(y_{t-1} - y_{t-5}) - 1p_{t-1}$$

```
Series: walmart.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
```

Above, shows the summary of the auto ARIMA model. The model automatically created an ARIMA(0,1,0)(1,1,0) model. The equation for this model is the following:

$$y_t - y_{t-1} = -0.3858(y_{t-1} - y_{t-5})$$

	walmart.ar.seas.alldata.pred.mean	walmart.ar.seas.auto.alldata.pred.mean
1	136482.7	135067.4
2	149557.6	148831.9
3	138457.2	138766.4
4	142974.5	143171.9
5	141524.1	140608.6
6	154691.6	154340.5

Above, table of the 6 forecasts for both the ARIMA(1,1,1)(1,1,1) and auto ARIMA model from 2020 Q3 to 2021 Q4.

e. 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.

```
# TSLM model
walmart.alldata.tslm.pred <- forecast(walmart.reg.quad.seas.alldata, h = 4, level = 0)
walmart.alldata.tslm.pred

# Residual model
walmart.alldata.tslm.pred.res <- walmart.alldata.tslm.pred$residuals
walmart.alldata.res.ar1 <- Arima(walmart.alldata.tslm.pred.res, order=c(1,0,0))
walmart.alldata.res.ar1.pred <- forecast(walmart.alldata.res.ar1, h=4, level=0)
walmart.alldata.res.ar1.pred

# Two level model
walmart.alldata.2level.pred <- walmart.alldata.tslm.pred$mean + walmart.alldata.res.ar1.pred$mean
walmart.alldata.2level.pred
walmart.alldata.2level.fitted <- walmart.alldata.tslm.pred$fitted + walmart.alldata.res.ar1.pred$fitted


# ARIMA(1,1,1)(1,1,1) model
walmart.ar.seas.alldata.pred <- forecast(walmart.ar.seas.alldata, h=4, level=0)
walmart.ar.seas.alldata.pred

# Auto ARIMA model
walmart.ar.seas.auto.alldata.pred <- forecast(walmart.ar.seas.auto.alldata, h=4, level = 0)
walmart.ar.seas.auto.alldata.pred

# Seasonal Naive model
walmart.snaive.pred <- snaive(walmart.ts, h=4)
walmart.snaive.pred

#Accuracy
TSLM.acc <- accuracy(walmart.alldata.tslm.pred$fitted, walmart.ts)
twolevel.acc <- accuracy(walmart.alldata.2level.fitted, walmart.ts)
ARIMA.acc <- accuracy(walmart.ar.seas.alldata.pred$fitted, walmart.ts)
auto.ARIMA.acc <- accuracy(walmart.ar.seas.auto.alldata.pred$fitted, walmart.ts)
snaive.acc <- accuracy(walmart.snaive.pred$fitted, walmart.ts)
```

Above, codes for reassigning all models that uses all data to forecast only 4 quarters instead of 6. Common measures of accuracy for each model were then calculated.

```
TSLM <- c(TSLM.acc[2], TSLM.acc[5])
TwoLevel <- c(twolevel.acc[2], twolevel.acc[5])
ARIMA <- c(ARIMA.acc[2], ARIMA.acc[5])
auto.ARIMA <- c(auto.ARIMA.acc[2], auto.ARIMA.acc[5])
seasonal_naive <- c(snaive.acc[2], snaive.acc[5])

labels <- c("RMSE","MAPE")

alldata.table <- data.frame(labels, TSLM, TwoLevel, ARIMA, auto.ARIMA, seasonal_naive)
alldata.table
```

Above, codes for creating finding the RMSE and MAPE for each model, and then displaying them all on a table.

```
> alldata.table
  labels        TSLM    TwoLevel        ARIMA  auto.ARIMA seasonal_naive
1   RMSE 2729.215986 1619.310532 1641.8772260 1920.469220    5074.236428
2   MAPE    1.888066    1.082851    0.9832904    1.222555       3.872531
```

Shown above in the table, all models did better than the seasonal naïve model. The two level and ARIMA (1,1,1)(1,1,1) models were very close in terms of common accuracies, with the two level model having only a slightly higher MAPE and the ARIMA model having only a slightly higher RMSE. Both models probably have similar performances, and either can be used as the best model.