Stat 280 Exercise 4 - Veronica Bayani

Name: Veronica Bayani Student Number: 2009-00574

I. Use the sale_app data (Total Sales of Appliance Units in the Philippines, Jan 2000 – Dec 2009) in PhilMonthlyData.csv.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## Warning: package 'readr' was built under R version 4.2.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(fpp2)
```

```
## Warning: package 'fpp2' was built under R version 4.2.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
## -- Attaching packages ---------------------------------------------- fpp2 2.4 --
## v forecast  8.18     v expsmooth 2.3
## v fma       2.4
```

```
## Warning: package 'forecast' was built under R version 4.2.2
```

```
##
```

```
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.2.2
```

```
library(forecast)

philmon <- read.csv("PhilMonthlyData.csv", stringsAsFactors = FALSE, na.strings = c("NA"))
```

Getting the Sales of Appliances in Units Sold from 2000 to 2009

```
saleapp <- ts(na.omit(philmon$sale_app),start=c(2000,1),frequency=12)
```

Split the data into training and test data set:

Training dataset = Jan 2000 – Dec 2007; and Test dataset = Jan 2008 – Dec 2009.

```
#train dataset
saleapp_train <- window(saleapp,start=2000,end=c(2007,12))
saleapp_train
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2000 474500.0 491300.0 561800.0 528900.0 663200.0 665700.0 524800.0 555000.0
## 2001 468800.0 435100.0 527800.0 491700.0 558200.0 559400.0 487100.0 465800.0
## 2002 424400.0 413300.0 482700.0 613800.0 690600.0 637600.0 564300.0 536900.0
## 2003 518495.7 423300.0 560250.0 601000.0 638400.0 596400.0 580100.0 491400.0
## 2004 509400.0 508500.0 668500.0 664300.0 660100.0 627100.0 565300.0 543800.0
## 2005 452500.0 441300.0 484600.0 537300.0 547400.0 519400.0 483200.0 446600.0
## 2006 369300.0 335300.0 405200.0 422300.0 462300.0 428100.0 386700.0 372400.0
## 2007 257400.0 265300.0 310000.0 283100.0 342200.0 326350.0 258700.0 254200.0
##           Sep      Oct      Nov      Dec
## 2000 493000.0 513000.0 581300.0 676900.0
## 2001 438200.0 536700.0 538900.0 568100.0
## 2002 527400.0 592100.0 654800.0 714600.0
## 2003 515200.0 642800.0 618300.0 730500.0
## 2004 557200.0 625300.0 708600.0 710100.0
## 2005 477300.0 471000.0 529400.0 592744.1
## 2006 351600.0 338800.0 426300.0 438500.0
## 2007 265950.0 304000.0 348000.0 373800.0
```

```
#test dataset
saleapp_test <- window(saleapp,start=2008,end=c(2009,12))
saleapp_test
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2008 299503.0 287403.0 346505.0 306229.0 322493.0 322071.0 290698.0 278103.0
## 2009 238044.7 217870.3 303801.5 321495.3 373995.3 348701.5 284970.3 261957.8
##           Sep      Oct      Nov      Dec
## 2008 314734.0 344043.5 391781.0 441736.5
## 2009 256926.5 306657.8 354395.3 404350.8
```

1) [2pts] Using the auto.arima() function, show the best performing model for the training dataset and show the equations form of the model with estimated parameter values plugged in (for reference, see the ARIMA(1,0,3) equation form in https://otexts.com/fpp2/non-seasonal-arima.html)

```r
sales_arima <- auto.arima(saleapp_train, seasonal = F, stepwise = F, approximation = F)
summary(sales_arima)
```

```
## Series: saleapp_train
## ARIMA(5,1,0)
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5
##       -0.4403  -0.6626  -0.5144  -0.5921  -0.2795
## s.e.   0.0997   0.0918   0.0995   0.0903   0.1000
##
## sigma^2 = 3.738e+09:  log likelihood = -1180.28
## AIC=2372.56   AICc=2373.51   BIC=2387.88
##
## Training set error measures:
##                     ME    RMSE      MAE       MPE     MAPE      MASE
## Training set -7262.162 59199.4 45525.43 -3.065331 9.506404 0.5901816
##                     ACF1
## Training set -0.02575979
```

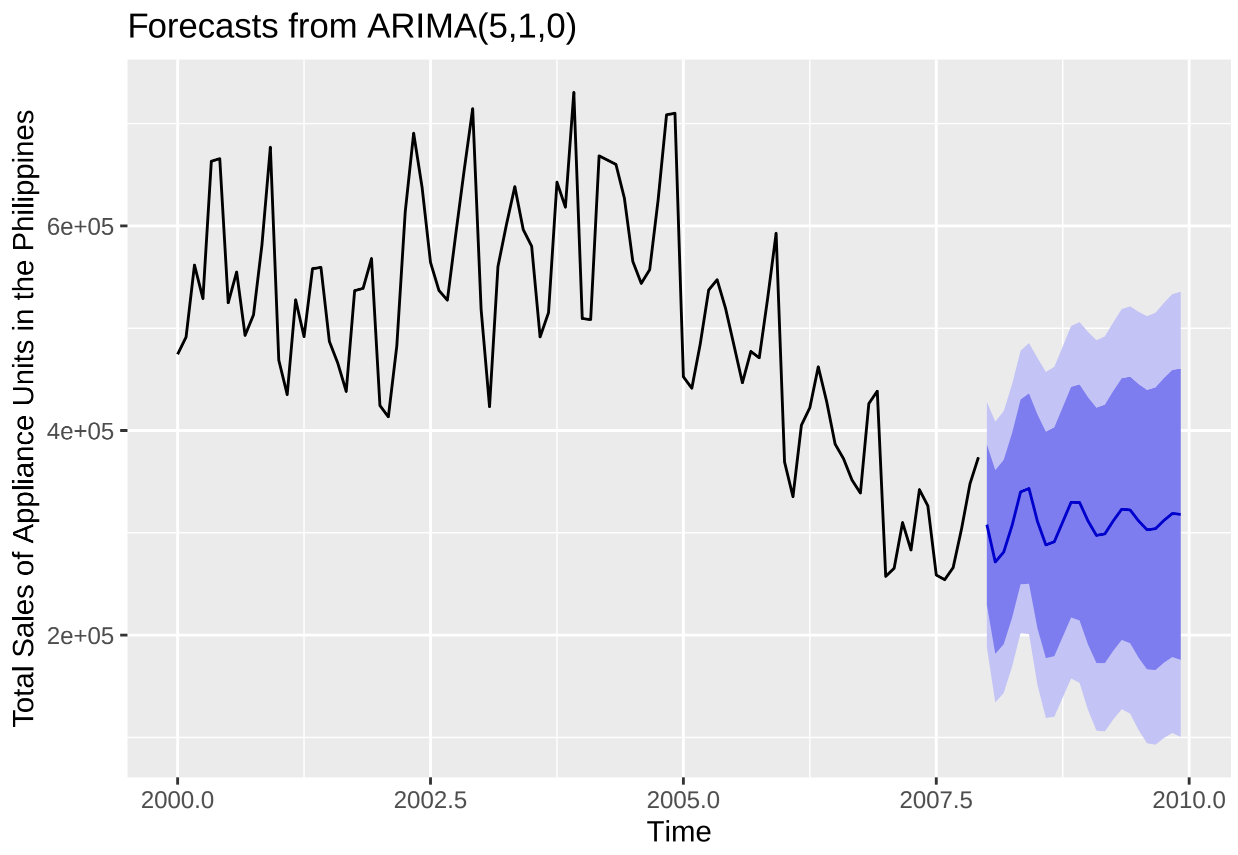
The best performing model is an ARIMA(5,1,0) model with a zero mean. The equation for this model is:

$$y_t = -0.4403y_{t-1} - 0.6626y_{t-2} - 0.5144y_{t-3} - 0.5921y_{t-4} - 0.2795y_{t-5} + \epsilon_t$$

2) [1pt] Based on the model in (1), show a plot of the forecasted value of sale_app for the test data added into the plot of the full dataset. Analyze the plot in terms of the forecasting performance of the selected model in (1)

Forecasting using the ARIMA(5,1,0) model

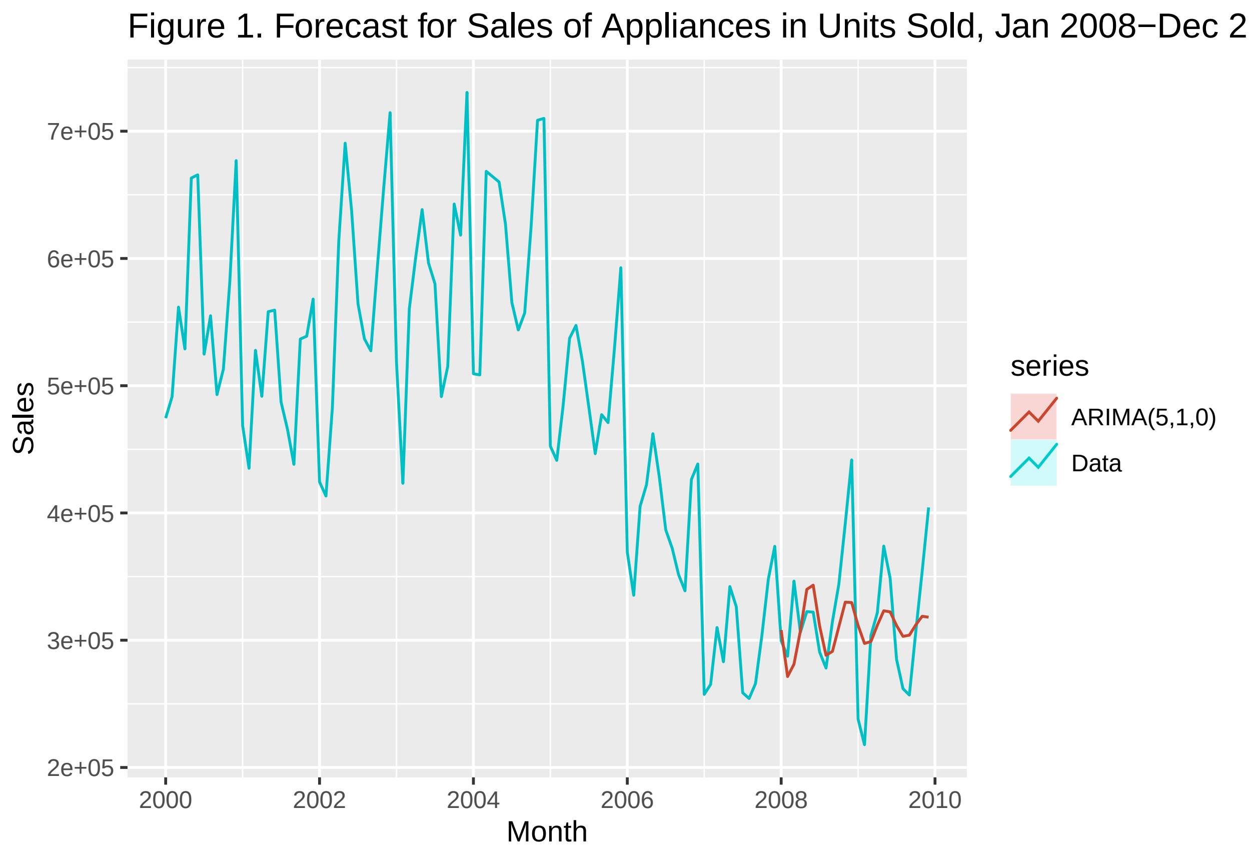
```r
sales_arima %>% forecast(h=24) %>%
  autoplot() +
  ylab("Total Sales of Appliance Units in the Philippines")
```



Plotting the forecasts and the actual values together,

```
autoplot(saleapp, series="Data") +
  autolayer(forecast(sales_arima, newdata = saleapp_test), series="ARIMA(5,1,0)", PI=FALSE) +
  xlab("Month") + ylab("Sales") +
  ggtitle("Figure 1. Forecast for Sales of Appliances in Units Sold, Jan 2008-Dec 2009")
```

```
## Warning in forecast.forecast_ARIMA(sales_arima, newdata = saleapp_test): The
## non-existent newdata arguments will be ignored.
```



Based on the plot, the ARIMA(5,1,0) does not do a good job in forecasting since it does not capture the seasonality well. The model under forecasts during the peak seasons and over forecasts during the off peak seasons.

3) [1pt] Generate the accuracy measures of the selected model in (1) with respect to the testing data set. Write a short analysis based on the accuracy measures.

```
accuracy(forecast(sales_arima,h=24), saleapp_test)
```

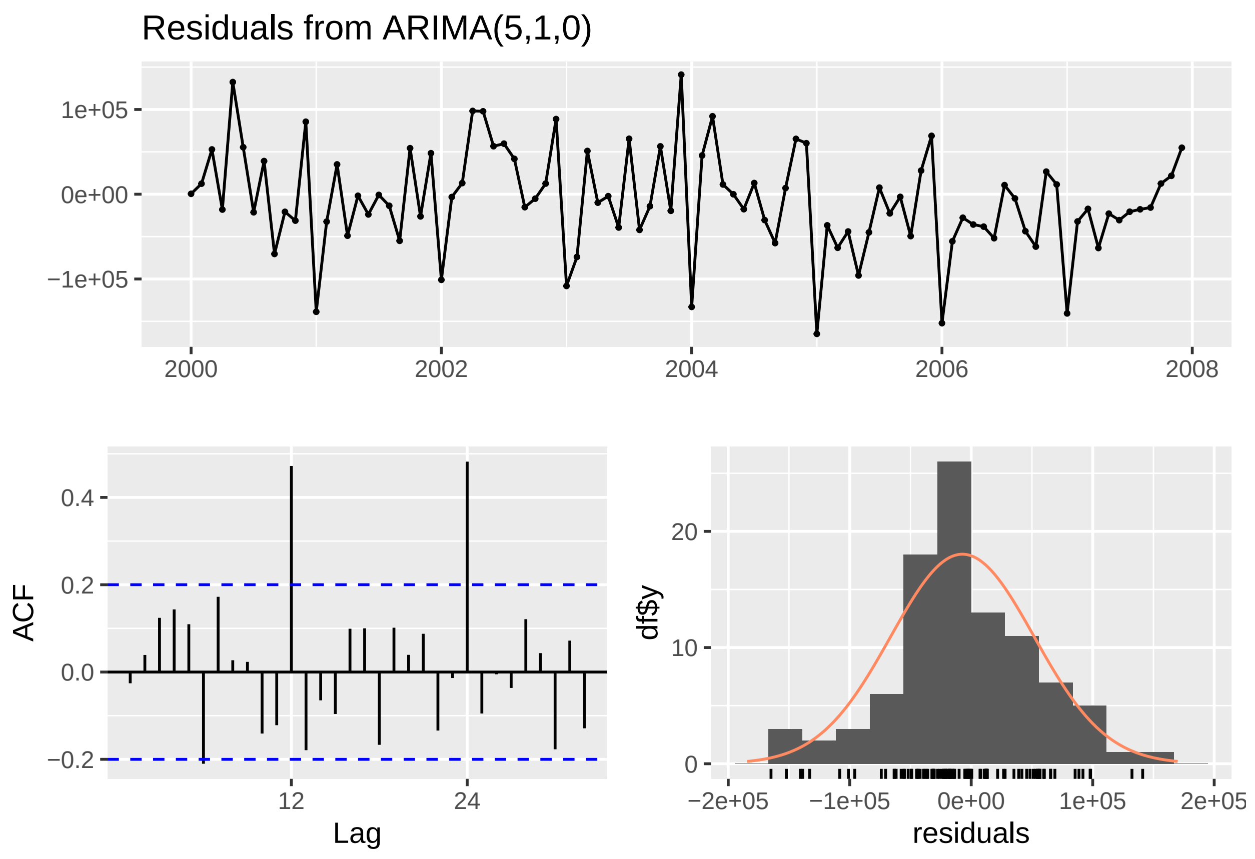
```
##                     ME     RMSE      MAE         MPE      MAPE      MASE
## Training set -7262.162 59199.40 45525.43 -3.06533142  9.506404 0.5901816
## Test set      7245.283 46740.82 36589.32  0.06571955 11.598453 0.4743359
##                     ACF1 Theil's U
## Training set -0.02575979        NA
## Test set      0.29757812 0.8697451
```

The RMSE, MAPE and MAE for the ARIMA(5,1,0) model are still quite high even if this is the best model with the lowest AICc as recommended by the auto.arima function. This is validated on the plot where the forecast values are either under or overforecasted as compared to the actual values so there is still room for improvement in the ARIMA(5,1,0) model.

4) [1pt] Check the residuals of the selected model in (1). Has the selected model in (1) complied with the properties that residuals should have for full extraction of the patterns from the time series? Any recommendations?

Testing for the presence of autocorrelation,

```
checkresiduals(sales_arima)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,1,0)
## Q* = 53.939, df = 14, p-value = 1.316e-06
##
## Model df: 5.   Total lags used: 19
```

Using a p value of 0.05, there is sufficient evidence to conclude that the residuals are not independently distributed and autocorrelation is present in the data.

Testing for the normality of the residuals,

```
shapiro.test(residuals(sales_arima))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(sales_arima)
## W = 0.97986, p-value = 0.1469
```

Based on the results of the Shapiro-Wilk normality test and using a p value of 0.05, the residuals are normally distributed.

The ARIMA(5,1,0) does not fully comply with the properties that residuals should have for full extraction of the patterns from the time series. There is sufficient evidence to conclude that autocorrelation is present based on the Ljung-Box test results even if the residuals are normally distributed.

To further improve the performance of the model, the autocorrelation issues must be addressed. This can be done by exploring the addition of other predictor variables in the model or via variable transformation.

II. Use the volpal data (Volume of Palay Production, Q1 1994 – Q4 2008) in PhilQuarterData.csv.

```
PhilQuarterlyData <- read.csv("PhilQuarterData.csv", stringsAsFactors = FALSE, na.strings = c("NA"))

#Quarterly Volume of Palay production data

palay_quarter <- ts(na.omit(PhilQuarterlyData$volpal),start=c(1994,1),frequency=4)
palay_quarter
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 1994 2288317 2090216 1876635 4282886
## 1995 2272045 2045286 1785510 4437808
## 1996 2523794 2427116 2116498 4216160
## 1997 2563757 2282704 1788141 4634361
## 1998 2220968 1338008 1284443 3711405
## 1999 2996188 2275865 2248702 4265870
## 2000 2856356 2586140 2412610 4534306
## 2001 2813930 2753901 2405187 4981852
## 2002 3058094 2614275 2020796 5577488
## 2003 3035561 2345717 2434696 5683910
## 2004 3434383 2604199 2871678 5586524
## 2005 3381885 2651140 2669137 5900843
## 2006 3615607 2923824 3006200 5781075
## 2007 3676704 3051176 3146959 6365355
## 2008 3748852 3371867 3467460 6227369
```

Split the data into training and test data set:

Training dataset = Q1 1994 – Q4 2005; and Test dataset = Q1 2006 – Q4 2008.

```
#train dataset
palay_train <- window(palay_quarter,start=1994,end=c(2005,4))
palay_train
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 1994 2288317 2090216 1876635 4282886
## 1995 2272045 2045286 1785510 4437808
## 1996 2523794 2427116 2116498 4216160
## 1997 2563757 2282704 1788141 4634361
## 1998 2220968 1338008 1284443 3711405
## 1999 2996188 2275865 2248702 4265870
## 2000 2856356 2586140 2412610 4534306
## 2001 2813930 2753901 2405187 4981852
## 2002 3058094 2614275 2020796 5577488
## 2003 3035561 2345717 2434696 5683910
## 2004 3434383 2604199 2871678 5586524
## 2005 3381885 2651140 2669137 5900843
```

```
#test dataset
palay_test <- window(palay_quarter,start=2006,end=c(2008,4))
palay_test
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 2006 3615607 2923824 3006200 5781075
## 2007 3676704 3051176 3146959 6365355
## 2008 3748852 3371867 3467460 6227369
```

1) [2pts] Using the auto.arima() function, show the best performing model for the training dataset and show the equations form of the model with estimated parameter values plugged in (for reference, see the ARIMA(1,0,3) equation form in https://otexts.com/fpp2/non-seasonal-arima.html)

```
palay_arima <- auto.arima(palay_train, seasonal = F, stepwise = F, approximation = F)
summary(palay_arima)
```

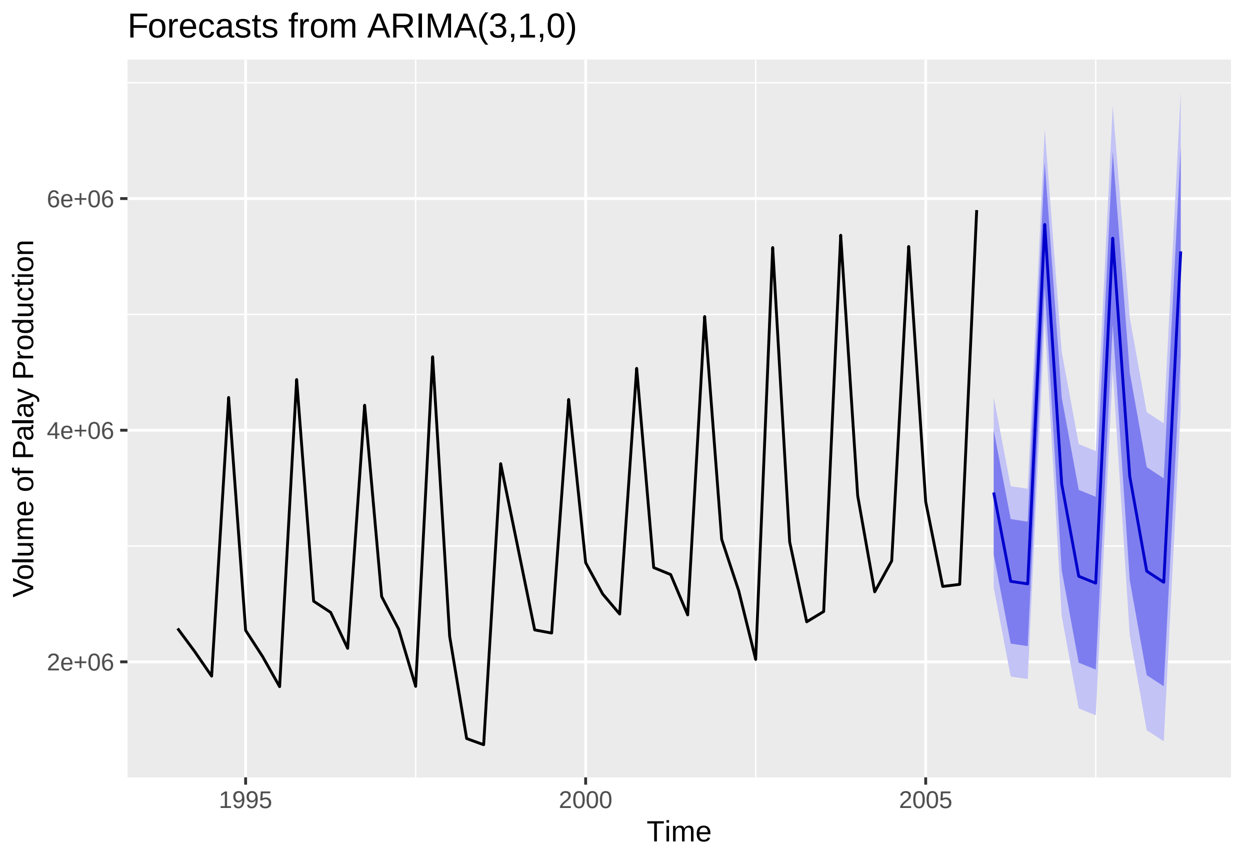
```
## Series: palay_train
## ARIMA(3,1,0)
##
## Coefficients:
##           ar1      ar2      ar3
##       -0.9665  -0.9614  -0.9616
## s.e.   0.0392   0.0389   0.0263
##
## sigma^2 = 1.755e+11:  log likelihood = -677.74
## AIC=1363.48   AICc=1364.44   BIC=1370.88
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 93002.33 401137.1 310472.6 0.9091582 11.74968 0.9827099 0.2504244
```

The best performing model is an ARIMA(3,1,0) model with a zero mean. The equation for this model is:

$$y_t = -0.9665y_{t-1} - 0.9614y_{t-2} - 0.9616y_{t-3} + \epsilon_t$$

2) [1pt] Based on the model in (1), show a plot of the forecasted value of volpal for the test data added into the plot of the full dataset. Analyze the plot in terms of the forecasting performance of the selected model in (1)

```
palay_arima %>% forecast(h=12) %>%
  autoplot() +
  ylab("Volume of Palay Production")
```

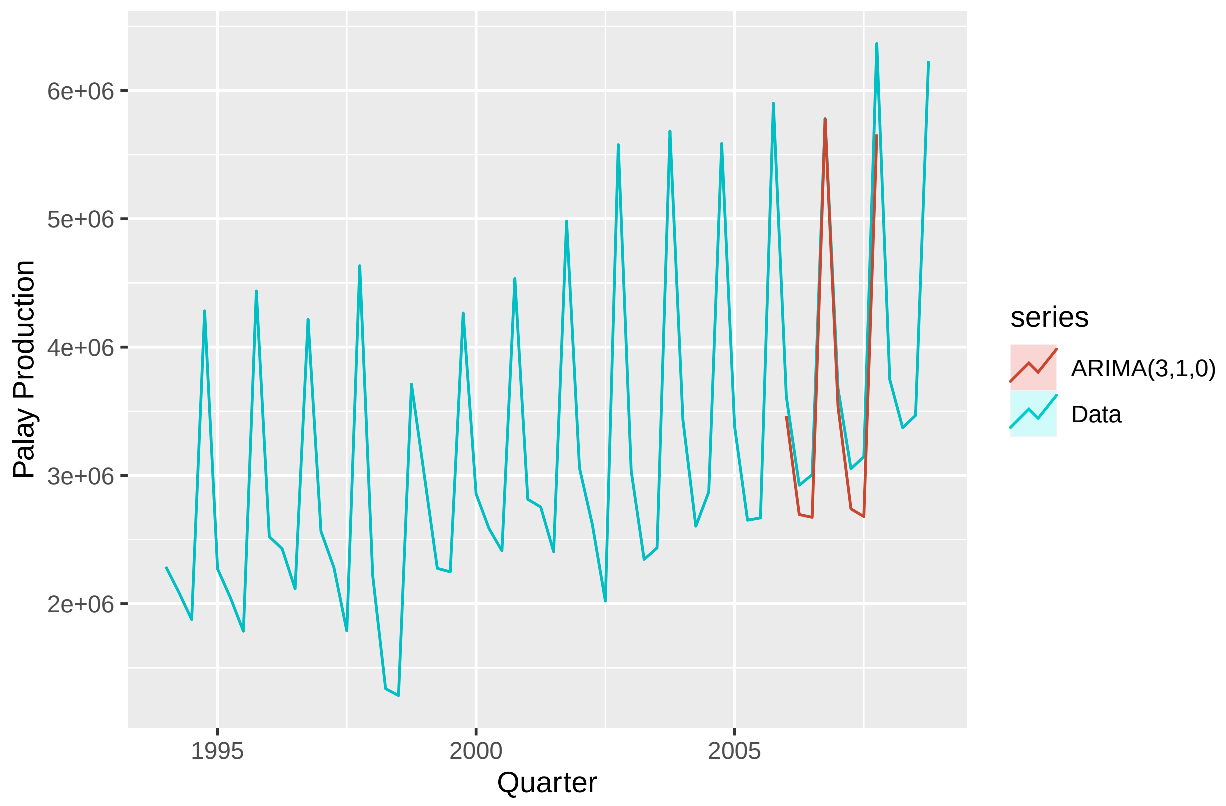


Plotting the forecasts and the actual values together,

```
autoplot(palay_quarter, series="Data") +
  autolayer(forecast(palay_arima, newdata = palay_test), series="ARIMA(3,1,0)", PI=FALSE) +
  xlab("Quarter") + ylab("Palay Production") +
  ggtitle("Figure 2. Forecast for Volume of Palay Production, Q1 2006 - Q4 2008")
```

```
## Warning in forecast.forecast_ARIMA(palay_arima, newdata = palay_test): The non-
## existent newdata arguments will be ignored.
```

Figure 2. Forecast for Volume of Palay Production, Q1 2006 – Q4 2008



Based on the plot of the actual and the forecast values, the ARIMA(3,1,0) model provides a good forecast for the Volume of Palay Production. The seasonality and trend are captured well in the forecast and, except for the slight underforecasting in the off-peak quarters, the actual and forecast values are close to each other.

3) [1pt] Generate the accuracy measures of the selected model in (1) with respect to the testing dataset. Write a short analysis based on the accuracy measures.

Checking the accuracy for the ARIMA(3,1,0) model

```
accuracy(forecast(palay_arima,h=12), palay_test)
```

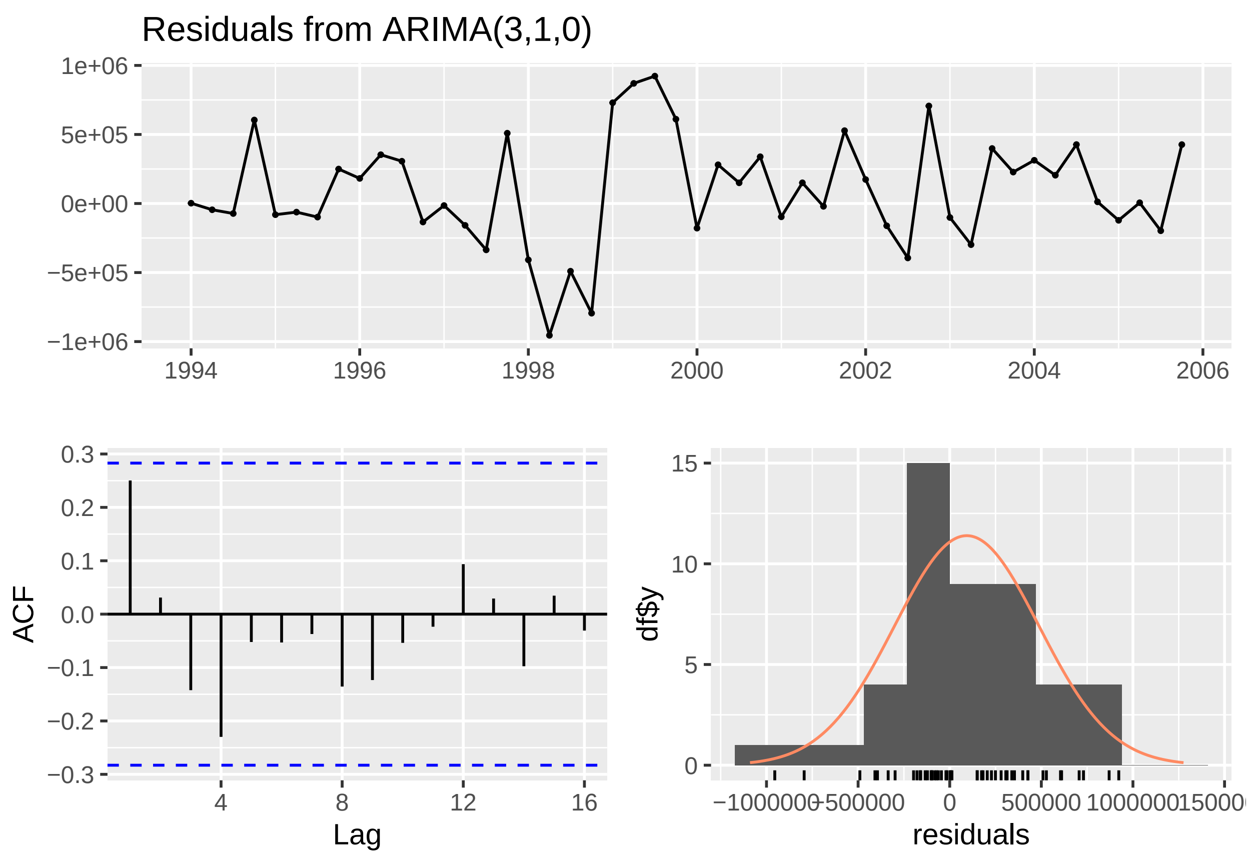
```
##                     ME     RMSE      MAE       MPE      MAPE      MASE
## Training set  93002.33 401137.1 310472.6 0.9091582 11.749681 0.9827099
## Test set     378719.62 453347.6 378719.6 9.8374455  9.837445 1.1987260
##                   ACF1 Theil's U
## Training set 0.2504244        NA
## Test set     0.3590336 0.2725992
```

The RMSE and MAE of the ARIMA(3,1,0) model slightly increased versus the RMSE and MAE of the train data set. The MAPE even improved as compared to the MAPE of the train data set. This is expected since, as seen on the plot, the forecast values are reasonably close to the actual values.

4) [1pt] Check the residuals of the selected model in (1). Has the selected model in (1) complied with the properties that residuals should have for full extraction of the patterns from the time series? Any recommendations?

Testing for the presence of autocorrelation,

```
checkresiduals(palay_arima)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,0)
## Q* = 8.7109, df = 5, p-value = 0.1212
##
## Model df: 3.   Total lags used: 8
```

Using a p value of 0.05, there is sufficient evidence to conclude that the residuals are independently distributed and there is no autocorrelation present in the data.

Testing for the normality of the residuals,

```
shapiro.test(residuals(palay_arima))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(palay_arima)
## W = 0.98238, p-value = 0.6802
```

Based on the results of the Shapiro-Wilk normality test, the residuals are normally distributed.

The ARIMA(3,1,0) fully complies with the properties that residuals should have for full extraction of the patterns from the time series. There is enough evidence to conclude that autocorrelation is not present and the residuals are normally distributed. This is validated by the plot of the forecast and the actual values where it is seen that the ARIMA(3,1,0) model does an excellent job in forecasting for the volume of palay production.