For this section, I used the data on Total Sales of Appliance Units in the Philippines from Jan 2000 to Dec 2009 in PhilMonthlyData.csv. Split the data into training and test data set:

Training dataset = Jan 2000 – Dec 2007; and

Test dataset = Jan 2008 – Dec 2009.

R code

```
library(dplyr)
library(fpp2)
library(ggplot2)
library(nortest)
library(urca)
#read data
ph_monthly <-
read.csv("C:/Users/Administrator/Documents/PhilMonthlyData.csv")
sales_ts <- ts(na.omit(ph_monthly$sale_app), frequency = 12, start =
c(2000,1))
#train and test dataset
train1 <- subset(sales_ts, start = 1, end = 12*8)
test1 <- subset(sales_ts, start = length(sales_ts) - 12*2+1)
```

```
fit1 <- auto.arima(train1, stepwise = F, approximation = F)
summary(fit1)
```

The R code above gives the output:

```
Series: train1
ARIMA(1,1,2)(0,1,1)[12]

Coefficients:
          ar1     ma1      ma2     sma1
      -0.6250  0.3213  -0.4612  -0.6507
s.e.   0.1805  0.1738   0.0966   0.1526

sigma^2 estimated as 1.632e+09:  log likelihood=-999.7
AIC=2009.4   AICc=2010.18   BIC=2021.5

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE
Training set -3135.422 36649.57 25721.18 -0.833697 5.340406 0.3334437
                    ACF1
Training set -0.03085238
```

Using the auto.arima() function, the best performing model for the training dataset is the ARIMA(1, 1, 2)(0, 1, 1)[12] with equation (the solution before arriving at this equation can be found on the last page of this document):

$$\rightarrow y_t = (1 + \phi_1)y_{t-1} - \phi_1 y_{t-2} + y_{t-12} - (1 + \phi_1)y_{t-13} + \phi_1 y_{t-14} + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \theta_1\varepsilon_{t-12} + \theta_1\Theta_1\varepsilon_{t-13} + \Theta_1\theta_2\varepsilon_{t-14}$$

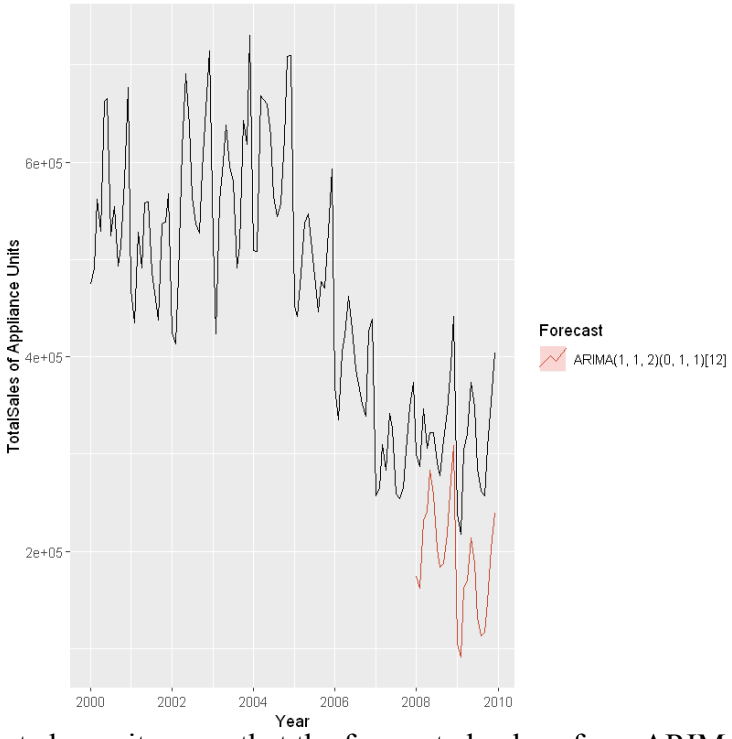
$$\rightarrow y_t = (1 + (-0.6250))y_{t-1} - (-0.6250)y_{t-2} + y_{t-12} - (1 + (-0.6250))y_{t-13} + (-0.6250)y_{t-14} + \varepsilon_t + 0.3213\varepsilon_{t-1} - 0.4612\varepsilon_{t-2} + 0.3213\varepsilon_{t-12} + 0.3213(-0.6507)\varepsilon_{t-13} + (-0.6507)(-0.4612)\varepsilon_{t-14}$$

$$\rightarrow y_t = 0.3750y_{t-1} + 0.6250y_{t-2} + y_{t-12} - 0.3750y_{t-13} - 0.6250y_{t-14} + \varepsilon_t + 0.3213\varepsilon_{t-1} - 0.4612\varepsilon_{t-2} + 0.3213\varepsilon_{t-12} - 0.2091\varepsilon_{t-13} + 0.3001\varepsilon_{t-14}$$

where the parameter estimates are $\widehat{\phi_1} = -0.6250$, $\widehat{\theta_1} = 0.3213$, $\widehat{\theta_2} = -0.4612$, and $\widehat{\Theta_1} = -0.6507$.

Based on this model, the plot of the forecasted value of sale_app for the test data added into the plot of the full dataset can be obtained by using the R code:

```
fcast1 <- forecast(fit1, h = 24)
autoplot(sales_ts) + autolayer(fcast1, series="ETS(M,A,M)", PI=FALSE)
+ xlab("Year") + ylab("Total Sales of Appliance Units") +
guides(colour=guide_legend(title="Forecast"))
```



From the plot above, it seems that the forecasted values from ARIMA(1, 1, 2)(0, 1, 1)[12] model always underestimate the actual values of the test dataset. Take for example, the sales of appliance units in December 2009 exceeds 40000, while the forecasted value for point in time is around 23000. Thus, the ARIMA(1, 1, 2)(0, 1, 1)[12] model might not be a good fit for the test dataset, but it seems to capture the seasonal pattern of actual data from Jan 2008 to Dec 2009.

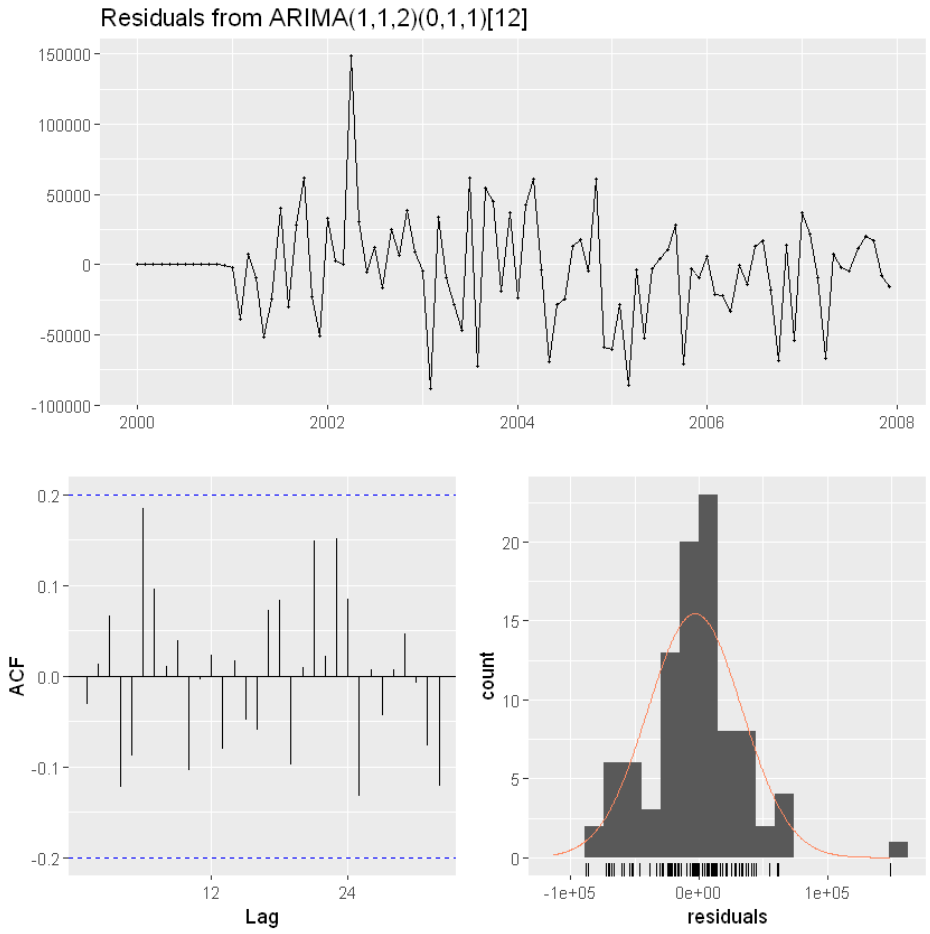
R code:

```
(acc1 <- data.frame(accuracy(fcast1, test1)) %>% select(RMSE,MAE,
MAPE, MASE) %>% slice(2))
```

RMSE	MAE	MAPE	MASE
129530.5	125294.1	40.37945	1.624285

Above is a table of accuracy measures of ARIMA(1, 1, 2)(0, 1, 1)[12] with respect to the testing dataset from Jan 2008 to Dec 2009. The MASE of the selected model is ~1.6243, which means that this model performs worse than the naïve forecast model on the same dataset. The mean absolute percent error (MAPE) between the sales predicted by the ARIMA(1, 1, 2)(0, 1, 1)[12] model and the actual sales is ~40.3795%, which seems to be very high. The model's mean absolute error (MAE) is 125294.1, which means that, on average, the sales forecast is expected to be 70063.98 sales units away from the actual sales of appliance units.

```
checkresiduals(fit1)
nortest::ad.test(residuals(fit1))
```



Test	Test Statistic		*p*-value
Ljung-Box test	Q*	12.949	0.6062
Anderson-Darling normality test	A	1.2979	0.002147

From the ARIMA(1, 1, 2)(0, 1, 1)[12] residual diagnostics histogram, the mean of the residuals seems to be centered at 0. However, the distribution of the histogram appears to have a long right tail which suggests that it is skewed to the right and not normal. To formally test for the normality of the residuals, an Anderson-Darling test was conducted. Since the *p*-value is less than .05, then there is enough evidence to conclude that there is a significant departure from normality. Although there seems to be a large variance in the first few months of 2002, the time plot shows that the variation of the residuals stays almost the same across the historical data, which means that it can be assumed that the variance of the residuals is constant.

There are no significant spikes in the ACF, but to formally test for autocorrelation, a Ljung-Box Test was conducted. The table above shows the computed test statistics and *p*-value of the Ljung-Box test. Since the test has a *p*-value greater than .05, it can be concluded that the residuals resemble white noise. A possible remedy is to consider other models or other SARIMA models which might pass all of these tests, but doing this might not give the best model in terms of the AICc value. Another remedy is to perform a Box-Cox transformation which might give a different SARIMA model.

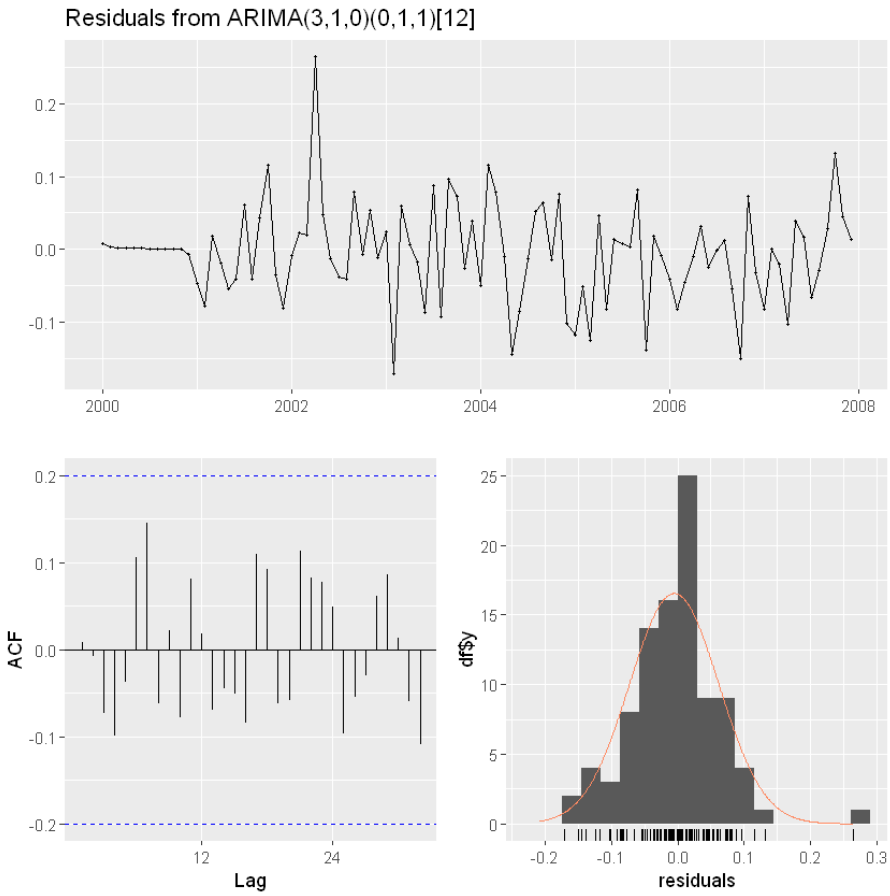
R code for recommendation:

```
fit1_1 <- auto.arima(train1, lambda = 0, stepwise = F, approximation
= F)
summary(fit1_1)
(acc1_1 <- data.frame(accuracy(forecast(fit1_1, h = 24), test1)) %>%
select(RMSE,MAE, MAPE, MASE) %>% slice(2))
checkresiduals(fit1_1)
nortest::ad.test(residuals(fit1_1))
```

```
Series: train1
ARIMA(3,1,0)(0,1,1)[12]
Box Cox transformation: lambda= 0

Coefficients:
          ar1      ar2     ar3     sma1
      -0.3647  -0.2819  0.1944  -0.7151
s.e.   0.1074   0.1109  0.1119   0.1670

sigma^2 = 0.005508:  log likelihood = 95.88
AIC=-181.76   AICc=-180.98   BIC=-169.66

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE
Training set -2587.405 34680.13 24435.92 -0.7774796 4.916868 0.3167819
                    ACF1
Training set 0.01023441
```

By performing a logarithmic transformation on the training dataset, the auto.arima() function identifies ARIMA(3,1,0)(0, 1, 1)[12] as the best model with better AICc and accuracy measures as shown below.

AICc	RMSE	MAE	MAPE	MASE
-180.98	65167.83	58113.65	17.58667	0.7533726



Test	Test Statistic		*p*-value
Ljung-Box test	Q*	11.85	0.6904
Anderson-Darling normality test	A	0.70626	0.0632

From the figures and table above, the ARIMA(3,1,0)(0, 1, 1)[12] model also complies with the properties that residuals should have for full extraction of patterns from the time series. The residual variance seems constant, and the Anderson-Darling test and Ljung-Box test *p*-values are greater than .05 which suggest that the residuals resemble white noise and do not depart from normality.

Using Volume of Palay Production, Q1 1994 – Q4 2008, in PhilQuarterData.csv. Split the data into training and test data set:

Training dataset = Q1 1994 – Q4 2005; and

Test dataset = Q1 2006 – Q4 2008.

R code

```
#read data
ph_quarterly <-
read.csv("C:/Users/Administrator/Documents/PhilQuarterData.csv")
volpal_ts <- ts(na.omit(ph_quarterly$volpal), frequency = 4, start =
c(1981,1))
#train and test dataset
train2 <- subset(volpal_ts, start = 1, end = 12*4)
test2 <- subset(volpal_ts, start = length(volpal_ts) - 3*4+1)
```

```
fit2 <- auto.arima(train2, stepwise = F, approximation = F)
summary(fit2)
```

The R code above gives the output:

```
Series: train2
ARIMA(1,0,0)(0,1,1)[4] with drift

Coefficients:
         ar1     sma1      drift
      0.3044  -0.5805  25158.209
s.e.  0.1433   0.1809   8825.583

sigma^2 estimated as 1.231e+11:  log likelihood=-623.54
AIC=1255.09   AICc=1256.11   BIC=1262.22

Training set error measures:
                    ME    RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set -8689.019 324266 233898.9 -2.433879 9.099651 0.7403385 0.006599536
```

Using the auto.arima() function, the best performing model for the training dataset is the ARIMA(1,0,0)(0,1,1)[4] (with drift) with equation (the solution before arriving at this equation can be found on the last page of this document):

$$\rightarrow y_t = c + \phi_1 y_{t-1} + y_{t-4} - \phi_1 y_{t-5} + \varepsilon_t + \Theta_1 \varepsilon_{t-4}$$

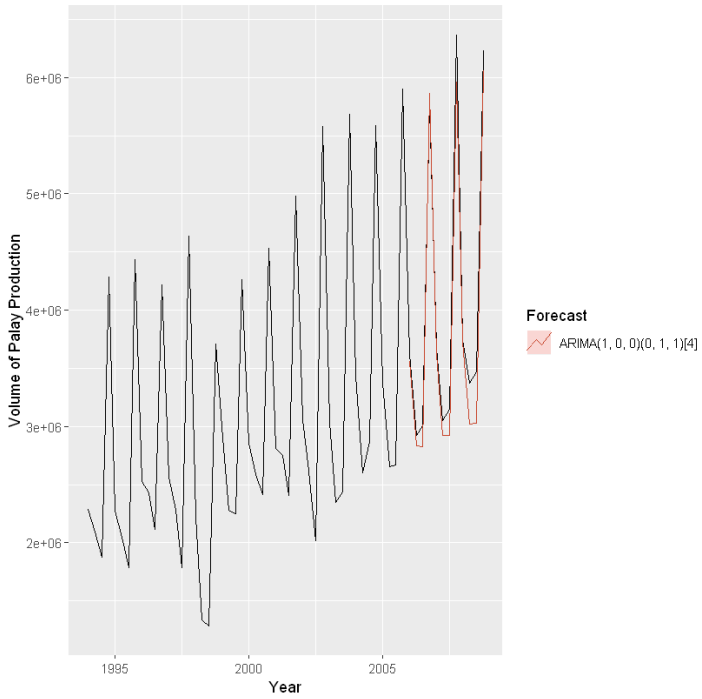
$$\rightarrow y_t = c + 0.3044 y_{t-1} + y_{t-4} - 0.3044 y_{t-5} + \varepsilon_t + (-0.5805)\varepsilon_{t-4}$$

$$\rightarrow y_t = c + 0.3044y_{t-1} + y_{t-4} - 0.3044y_{t-5} + \varepsilon_t - 0.5805\varepsilon_{t-4}$$
$$\rightarrow y_t = 17500.2105 + 0.3044y_{t-1} + y_{t-4} - 0.3044y_{t-5} + \varepsilon_t - 0.5805\varepsilon_{t-4}$$

where the parameter estimates are $\widehat{\phi_1} = 0.3044$, $\widehat{\Theta_1} = -0.5805$ and $c = 17500.2105$. The c was computed using the formula: $c = \mu(1 - \phi_1)$, where $\mu = 25158.209$. Thus, $c = \mu(1 - \phi_1) = 25158.209(1 - 0.3044) = 17500.2105$.

Based on this model, the plot of the forecasted value of pce for the test data added into the plot of the full dataset can be obtained by using the R code:

```
fcast2 <- forecast(fit2, h = 12)
autoplot(volpal_ts) +
  autolayer(fcast2, series="ETS(A,A,A)", PI=FALSE) +
  xlab("Year") + ylab("Personal Consumption Expenditure, in Million
Pesos") +
  guides(colour=guide_legend(title="Forecast"))
```



From the plot above, it seems that the forecast values of the ARIMA(1,0,0)(0,1,1)[4] (with drift) model are a close match to the test data, which contains the quarterly volume of palay production from Q1 2006 to Q4 2008. The model seems to capture the quarterly seasonal pattern and the slowly increasing trend at the end of the data.

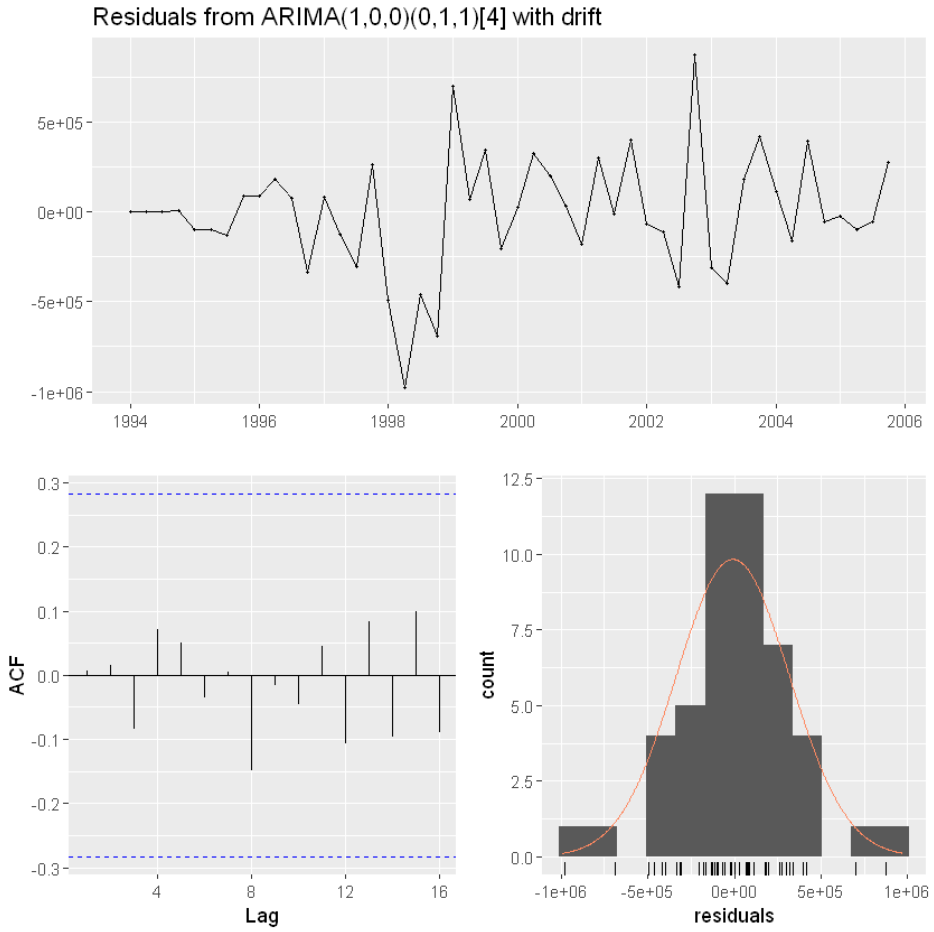
```
(acc2 <- data.frame(accuracy(fcast2, test2)) %>% select(RMSE,MAE,
MAPE, MASE) %>% slice(2))
```

The R code above gives:

RMSE	MAE	MAPE	MASE
228234.9	182744.5	4.780866	0.5784242

Above is a table of accuracy measures of ARIMA(1,0,0)(0,1,1)[4] (with drift) for the testing dataset from Q1 2006 to Q4 2008. The MASE of the selected model is ~0.5784242, which means this model performs better than the naïve forecast model on the same dataset. The mean absolute percent error (MAPE) between the palay production volume predicted by the ARIMA(1,0,0)(0,1,1)[4] (with drift) model and the actual expenditure is ~4.7809%, which seems to be small and good enough. The model's mean absolute error (MAE) is 182744.5, which means that, on average, the palay production volume forecast is expected to be 182744.5 units away from the actual quarterly palay production volume.

```
checkresiduals(fit2)
nortest::ad.test(residuals(fit2))
```



Test	Test Statistic		*p*-value
Ljung-Box test	Q*	2.2246	0.8979
Anderson-Darling normality test	A	0.52442	0.1728

From the ARIMA(1,0,0)(0,1,1)[4] (with drift) residual diagnostics histogram, the mean of the residuals seems to be centered at 0, and the distribution of the histogram does not seem to be skewed. To formally test for the normality of the residuals, an Anderson-Darling test was conducted. Since the *p*-value is greater than .05, then there is enough evidence to conclude that there is no significant departure from normality. The time plot shows that the variation of the residuals stays almost the same across the historical data, which means that it can be assumed that the variance of the residuals is constant.

There were no significant spikes in the ACF, but to formally test for autocorrelation, a Ljung-Box Test was conducted. The table above shows the computed test statistics and *p*-value of the Ljung-Box test. Since the test has a *p*-value greater .05, it can be concluded that the residuals are not distinguishable from a white noise. Since the ARIMA(1,0,0)(0,1,1)[4] (with drift) model has passed in both tests and the residual variation seems to be constant, then it can be stated that the model complied with the properties that residuals should have for full extraction of the patterns from the time series of palay production. Thus, this model can be used for forecasting.

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

$\rightarrow \phi_1(B)(1-B)^1(1-B^{12})^1 y_t = \Theta_1(B^{12})\theta_2(B)\varepsilon_t$

$\rightarrow (1-\phi_1 B)(1-B)\ (1-B^{12})\ y_t = (1+\ \Theta_1 B^{12})(1+\ \theta_1 B + \theta_2 B^2)\varepsilon_t$

$\rightarrow (1-B+\phi_1 B^2-\phi_1 B)(1-B^{12})\ y_t = (1+\ \theta_1 B^{12}+\ \theta_1 B + \theta_1\Theta_1 B^{13} + \theta_2 B^2+\ \Theta_1\theta_2 B^{14})\varepsilon_t$

$\rightarrow (1-B-\phi_1 B+\phi_1 B^2-B^{12}+B^{13}+\phi_1 B^{13}-\phi_1 B^{14})y_t =$
$(1+\ \theta_1 B^{12}+\ \theta_1 B + \theta_1\Theta_1 B^{13} + \theta_2 B^2+\ \Theta_1\theta_2 B^{14})\varepsilon_t$

$\rightarrow y_t - y_{t-1} - \phi_1 y_{t-1} + \phi_1 y_{t-2} - y_{t-12} + y_{t-13} + \phi_1 y_{t-13} - \phi_1 y_{t-14} =$
$\varepsilon_t+\ \theta_1\varepsilon_{t-12}+\ \theta_1\varepsilon_{t-1} + \theta_1\Theta_1\varepsilon_{t-13} + \theta_2\varepsilon_{t-2}+\ \Theta_1\theta_2\varepsilon_{t-14}$

$\rightarrow y_t - (1+\ \phi_1)y_{t-1} + \phi_1 y_{t-2} - y_{t-12} + (1+\phi_1)y_{t-13} - \phi_1 y_{t-14} =$
$\varepsilon_t+\ \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2}+\ \theta_1\varepsilon_{t-12} + \theta_1\Theta_1\varepsilon_{t-13}+\ \Theta_1\theta_2\varepsilon_{t-14}$

$\rightarrow y_t = (1+\ \phi_1)y_{t-1} - \phi_1 y_{t-2} + y_{t-12} - (1+\phi_1)y_{t-13} + \phi_1 y_{t-14} + \varepsilon_t+\ \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2}+\ \theta_1\varepsilon_{t-12}$
$+ \theta_1\Theta_1\varepsilon_{t-13} +\ \Theta_1\theta_2\varepsilon_{t-14}$

Δ

ARIMA(1,0,0)(0,1,1)[4] with drift

$\rightarrow \phi_1(B)(1-B)^0(1-B^4)^1 y_t = c + \Theta_1(B^4)\theta_0(B)\varepsilon_t$

$\rightarrow (1-\phi_1 B)(1-B^4)\ y_t = c + (1+\ \Theta_1 B^4)\varepsilon_t$

$\rightarrow (1-\phi_1 B - B^4 + \phi_1 B^5)y_t = c + (1+\ \Theta_1 B^4)\varepsilon_t$

$\rightarrow y_t - \phi_1 y_{t-1} - y_{t-4} + \phi_1 y_{t-5} = c + \varepsilon_t + \Theta_1\varepsilon_{t-4}$

$\rightarrow y_t = c + \phi_1 y_{t-1} + y_{t-4} - \phi_1 y_{t-5} + \varepsilon_t + \Theta_1\varepsilon_{t-4}$

Δ