Assignment 5

Matthew Dunne

July 22, 2018

Question 1

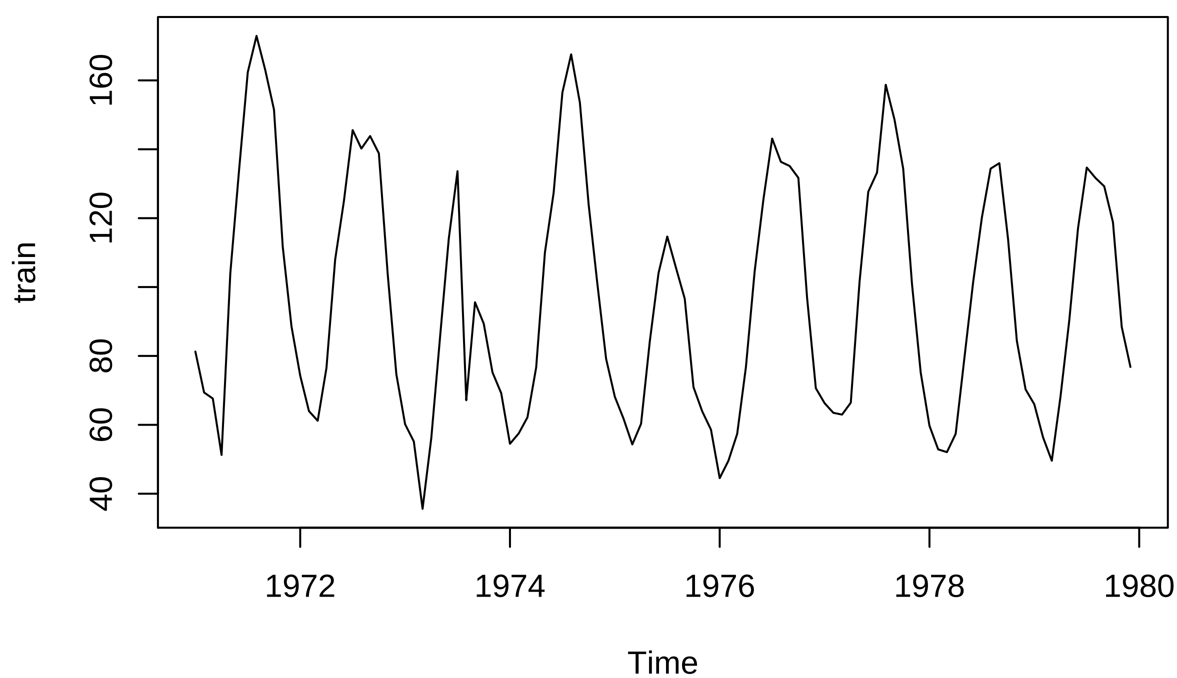
Load the condmilk.rda dataset and split it into a training dataset (1971/1 - 1979/12) and a test dataset (1980/1 - 1980/12).

```
suppressMessages(library(fpp))
suppressMessages(library(TSA))
train<-window(condmilk, c(1971, 1), c(1979,12))
test<-window(condmilk, c(1980, 1), c(1980,12))
```

Question 2

Plot the training dataset. Is Box-Cox transformation necessary for this data?

```
plot(train)
```



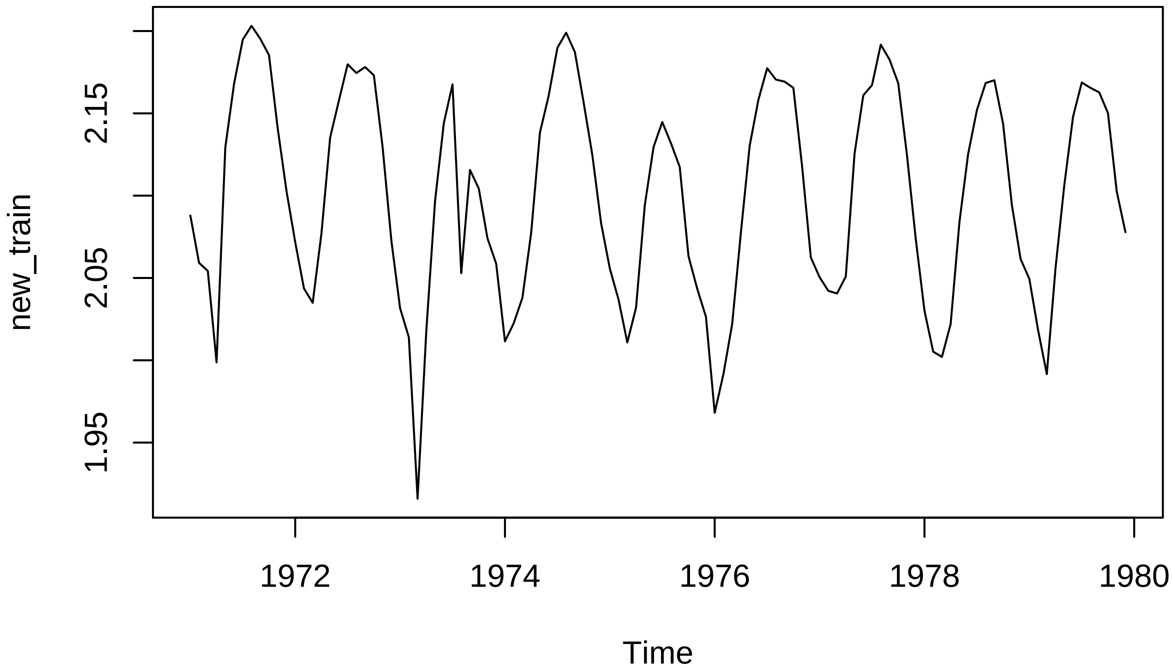
The variance clearly changes throughout the time series. So a Box-Cox Transformation is necessary. We will use the following lambda.

```
my_lambda<-BoxCox.lambda(train)
my_lambda
```

```
## [1] -0.3944237
```

After transforming the data data its plot is as follows:

```
new_train<-BoxCox(train, lambda = my_lambda)
plot(new_train)
```



The variance is now consistent over time.

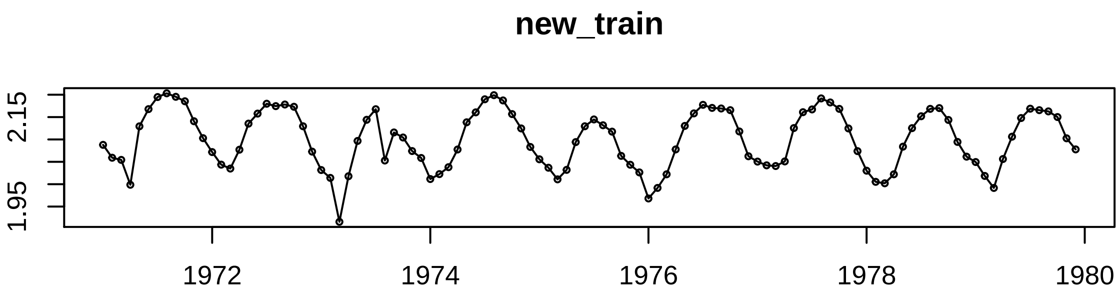
Question 3

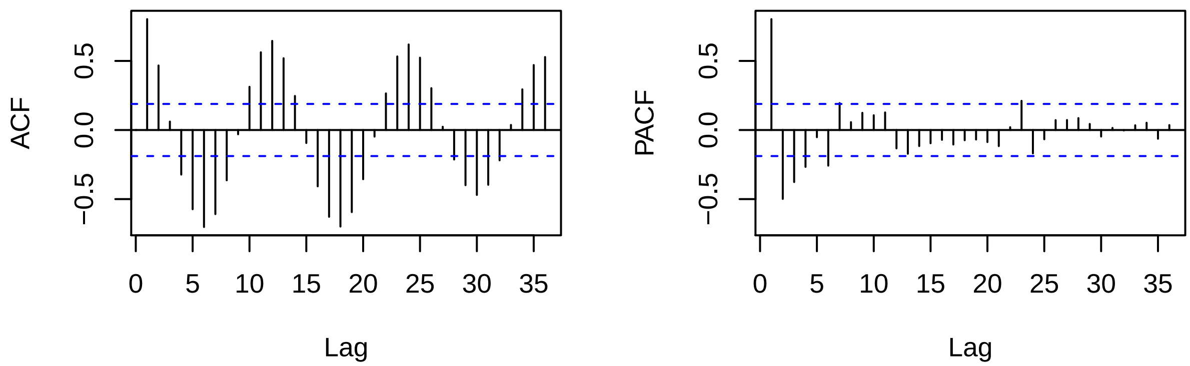
Is the training dataset stationary? If not, find an appropriate differencing which yields stationary data. Plot the ACF and PACF to determine the appropriate seasonal differencing which yields stationary data.

The mean of our transformed training data appears basically stable to the naked eye. Let us take a more rigorous look.

The ACF and PACF of our transformed data look as follows:

```
tsdisplay(new_train)
```





We see that the ACF drops quite quickly, though there clearly a seasonal component with spikes every 12 months. The ACF and PACF of the *non-transformed* data look substantially the same.

Applying both the Augmented Dickey-Fuller test and the KPSS test we confirm that the data is stationary.

The Augmented Dickey-Fuller Test has a small p-value, suggesting stationarity.

```
suppressWarnings(adf.test(new_train)$p.value)
```

```
## [1] 0.01
```

The KPSS Test has a large p-value, suggesting stationarity.

```
suppressWarnings(kpss.test(new_train)$p.value)
```

```
## [1] 0.1
```

Question 4

Build two ARIMA(p,d,q)(P,D,Q)s models using the training dataset and auto.arima() function.

- *Model 1: Let the auto.arima() function determine the best order of non-seasonal and seasonal differencing.*

```
model1<-auto.arima(new_train, seasonal=TRUE)
```

- *Model 2: Set the order of seasonal-differencing d to 1 and D to 1.*

```
model2<-auto.arima(new_train, d=1, D=1)
```

Report the resulting p,d,q,P,D,Q,s and the coefficient values for all cases and compare their AICc and BIC values.

For Model 1 we get ARIMA(1,0,0)(2,0,0)[12] with non-zero mean as well as the following coefficient values and AICc, BIC.

```
model1
```

```
## Series: new_train
## ARIMA(1,0,0)(2,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1    sar1    sar2    mean
##       0.6929  0.3973  0.3853  2.1067
## s.e.  0.0700  0.0859  0.0927  0.0262
##
## sigma^2 estimated as 0.0007656:  log likelihood=230.73
## AIC=-451.45   AICc=-450.86   BIC=-438.04
```

For Model 2 we get ARIMA(1,1,1)(2,1,0)[12] as well as the following coefficient values and AICc, BIC.

```
model2
```

```
## Series: new_train
## ARIMA(1,1,1)(2,1,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1     sar2
##       0.6164  -0.9566  -0.7541  -0.3823
## s.e.  0.1108   0.0638   0.1171   0.1152
##
## sigma^2 estimated as 0.0007327:  log likelihood=205.13
## AIC=-400.25   AICc=-399.58   BIC=-387.48
```

The AICc and BIC values are significantly lower for Model 2.

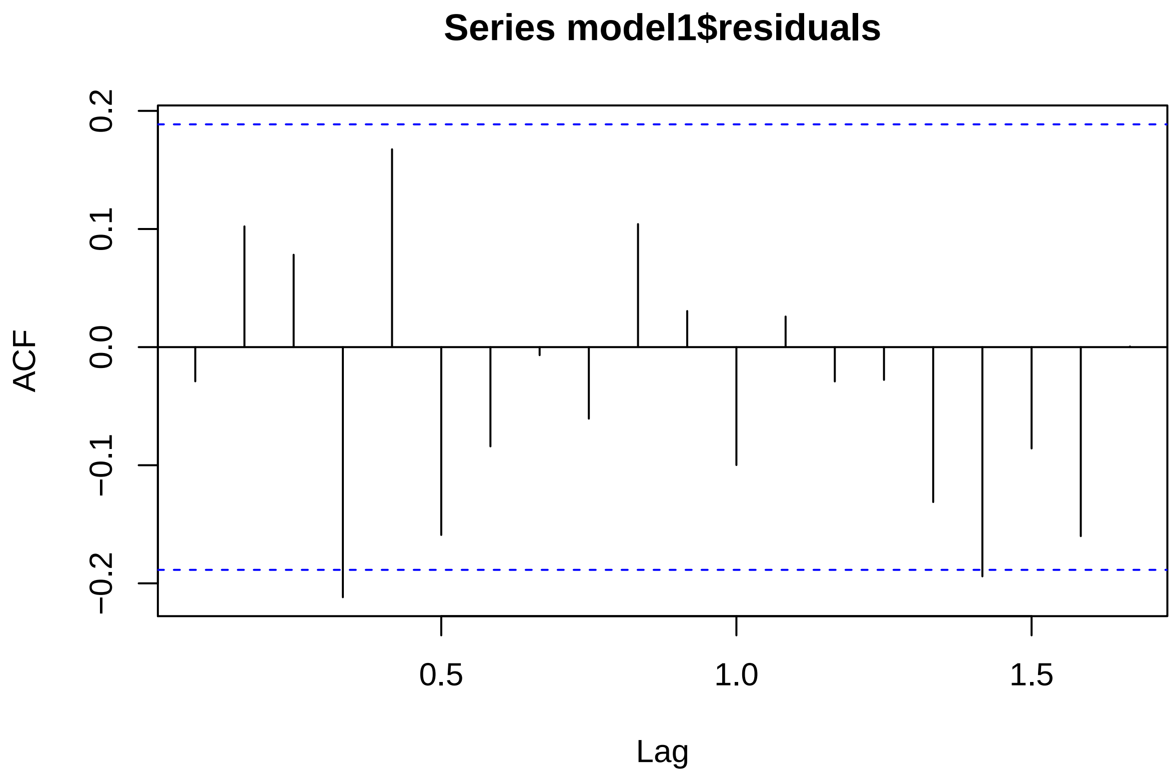
Question 5

Plot the residuals ACF of both models from part 4 and use the Ljung-Box Test with lag 12 to verify your conclusion.

If the model is good we would expect our residuals to be white noise. When we do an ACF plot of the residuals for each model we see that the auto-correlation of the residuals are within our confidence interval (with some probably coincidental spikes) and so are just white noise.

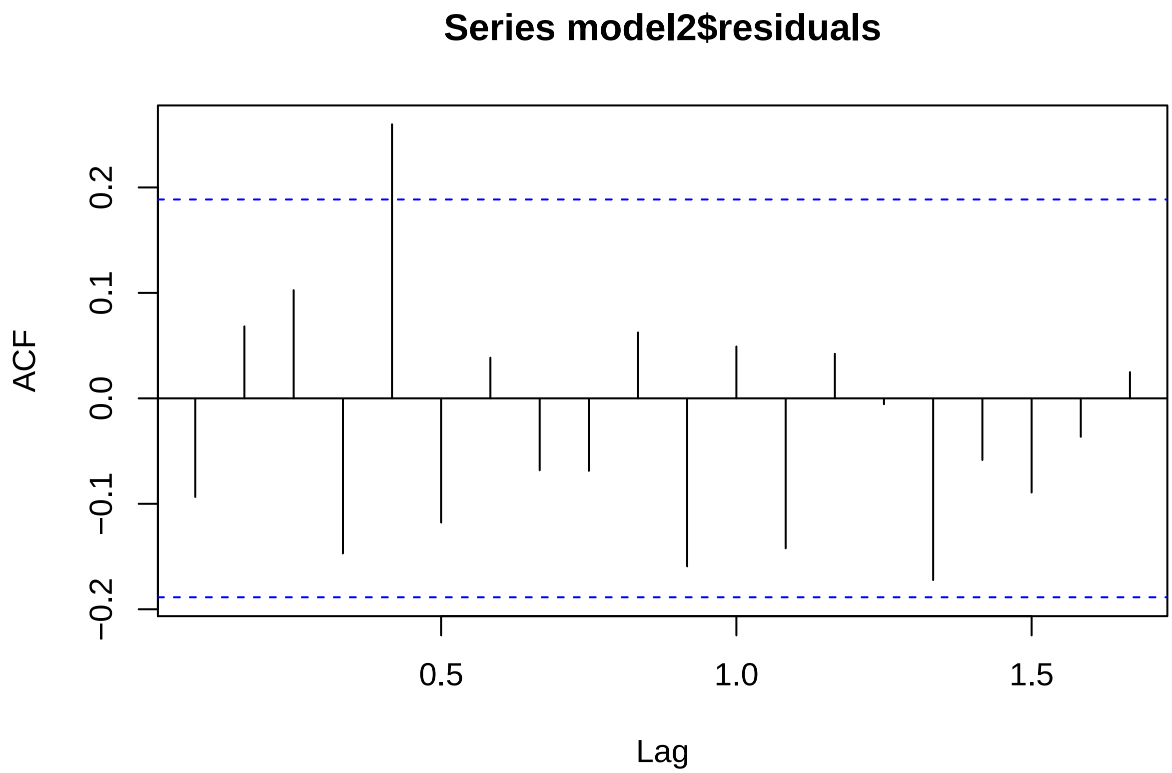
The ACF of the Model 1 residuals:

```
acf(model1$residuals)
```



The ACF of the Model 2 residuals:

```
acf(model2$residuals)
```



We can also use the Ljung-Box test to confirm. The null hypothesis for the test is that the data are independently distributed. A p-value of 0.05 means we would fail to reject the null hypothesis.

For Model 1, the Ljung-Box test yields a p-value of 0.1913.

```
Box.test(model1$residuals, lag=12)$p.value
```

```
## [1] 0.1912829
```

And for Model 2, the p-value is 0.1069.

```
Box.test(model2$residuals, lag=12)$p.value
```

```
## [1] 0.106915
```

So we fail to reject the null hypothesis. The residuals are white noise.

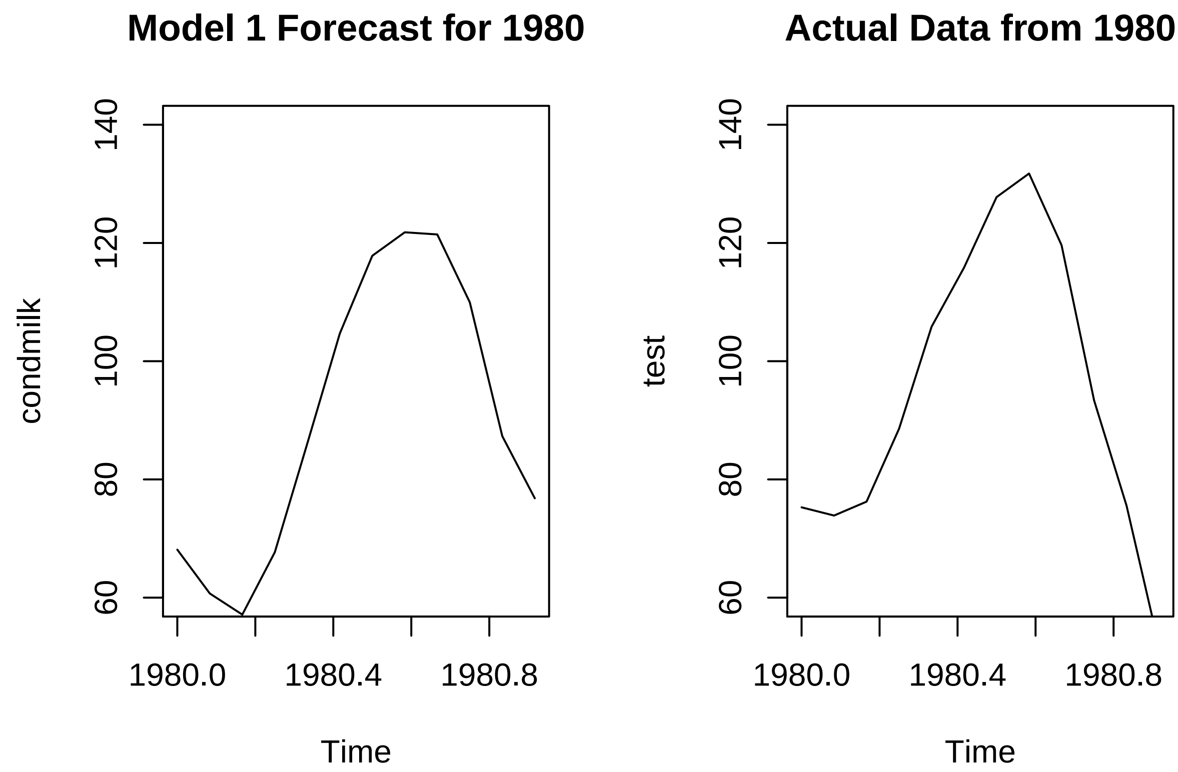
Question 6

Use both models from part 4 and the h-period argument in the forecast() function to forecast each month of 1980. Plot the test dataset and forecasted values.

For Model 1 the forecasted and actual values compare as follows:

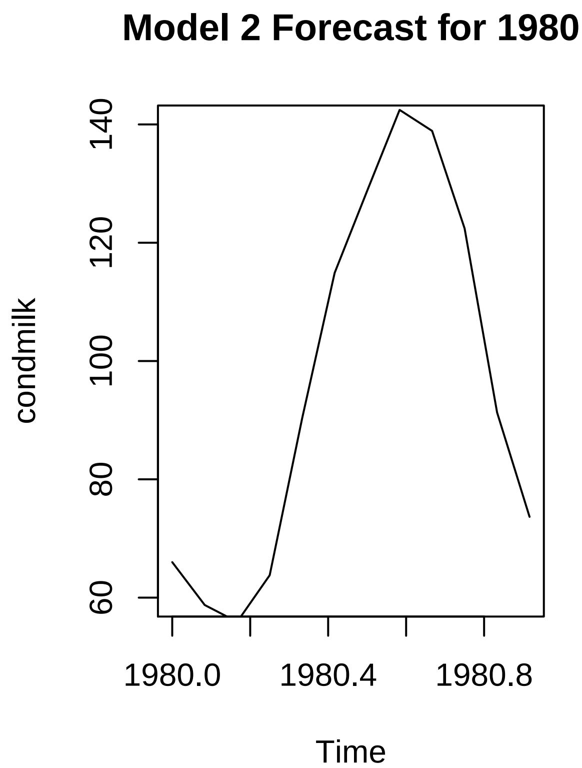
```
#create the forecast 12 months ahead
model1_forecast<-forecast(model1, h=12)
par(mfrow=c(1,2))
#make sure to de-transform the data so it is comparable to the raw test data. Set y-axis so comparison i
```

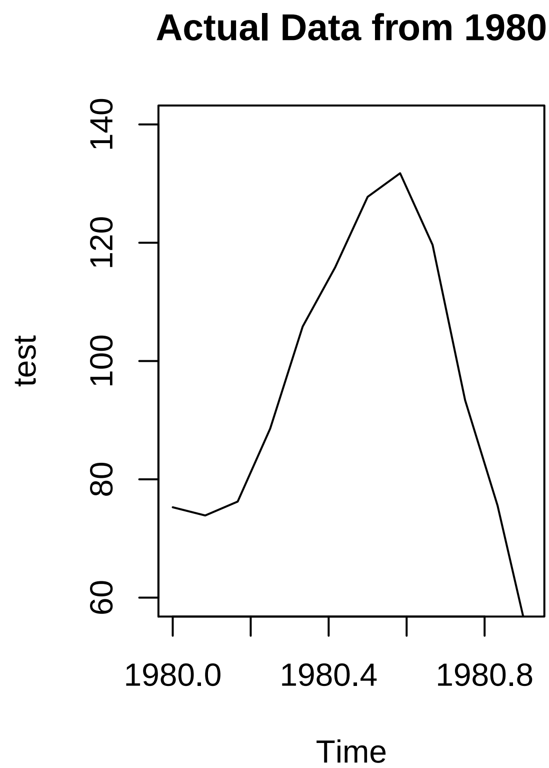
```
plot(InvBoxCox(model1_forecast$mean, lambda = my_lambda), main="Model 1 Forecast for 1980", ylab="condmi
plot(test, main="Actual Data from 1980", ylim=c(60,140))
```



For Model 2 the forecasted and actual values compare as follows:

```
#create the forecast 12 months ahead
model2_forecast<-forecast(model2, h=12)
par(mfrow=c(1,2))
#make sure to de-transform the data so it is comparable to the raw test data. Set y-axis so comparison i
plot(InvBoxCox(model2_forecast$mean, lambda = my_lambda), main="Model 2 Forecast for 1980", ylab="condmi
plot(test, main="Actual Data from 1980", ylim=c(60,140))
```





What we see is that both models underestimate actual values early in the year. Model 1 underestimates actual in the middle of the year while Model 2 overestimates. Both overestimate at the end of the year.

Question 7

Compare the forecast with the actual test data by calculating the Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE). Which models better to forecast the Manufacturer's Stocks for each month of 1980 (Jan, Feb, ., Dec)?

For Model 1 the MAPE (average deviation by percentage from actual value) is:

```
#the forecast on the de-transformed scale
model1_points<-InvBoxCox(model1_forecast$mean, lambda = my_lambda)
#calculate the MAPE
model1_mape<-sum((abs(model1_points-test)/test)*100)/length(test)
model1_mape
```

```
## [1] 16.89702
```

and its MSE is:

```
model1_mse<-sum((model1_points-test)^2)/length(test)
model1_mse
```

```
## [1] 231.7675
```

For Model 2 its MAPE is:

```
#the forecast on the de-transformed scale
model2_points<-InvBoxCox(model2_forecast$mean, lambda = my_lambda)
#calculate the MAPE
model2_mape<-sum((abs(model2_points-test)/test)*100)/length(test)
model2_mape
```

```
## [1] 18.51091
```

and its MSE is:

```
model2_mse<-sum((model2_points-test)^2)/length(test)
model2_mse
```

```
## [1] 303.4596
```

Taken as a whole, Model 1 performs better in terms of MAPE and MSE.

To compare how each forecast performs by month we will just look at how much each forecast is off by month.

```
m1<-model1_points-test
m2<-model2_points-test
cbind(Model_1=m1,Model_2=m2)
```

```
##             Model_1     Model_2
## Jan 1980  -7.160087  -9.2806781
## Feb 1980 -13.189022 -15.1348141
## Mar 1980 -19.108982 -20.3325828
## Apr 1980 -20.899411 -24.7937613
## May 1980 -19.746211 -15.4988784
## Jun 1980 -11.160928  -0.9362131
## Jul 1980  -9.922061   1.0043910
## Aug 1980  -9.944662  10.7068780
## Sep 1980   1.810064  19.2810731
## Oct 1980  16.576400  29.0377411
## Nov 1980  11.763744  15.7556574
## Dec 1980  25.015853  21.8688837
```

We see that Model 1 performs better January through April, while Model 2 performs better May through July. The models are essentially tied in August, and Model 1 performs better throughout the rest of the year, with the exception of December. So one model tends to perform better in the colder months and one performs better in the warmer months.