MA 641: TIME SERIES I
Midterm Exam

[redacted]

Due: July 15, 2008

I pledge my honor that I have abided by the Stevens Honor System.

[redacted]

Table of Contents

List of Tables

List of Figures

List of Appendixes

Problem Statement

The following items are the deliverables for the midterm exam:

Data Gathering Step

1. Go to Yahoo! Industry Center. We are interested in *Major Integrated Oil & Gas* and its relationship with *Major Airlines* and *Auto Manufacturers Major.*

2. Repeat the following steps for each of the three industries:

- Click on the industry of interest. There will be a list of the top five major corporations based on Market capitalization (not interested in Intraday performance).

- Download daily historical data for each of the five corporations for a period of at least 1.5 years. Use the same period for all the stocks.

- Import the data in R and for each of the stocks, calculate the daily return.

- Create a new vector equal to the average of the five return vectors just calculated, which is the new index for that industry.

Oil Analysis

3. Construct a time series model for the oil index just created. Use whichever model you wish (chose amongst the models presented in class) and report the one model you consider the best. Justify your choice using selected numbers from the output and graphs of your choice.

Bonus Using the best model you created for this exam, forecast the next 4 observations. Then using the data from July 15-July 18, calculate the sum of the squared errors and sum of absolute deviations. Report these numbers. Each reported value (justified of course) will receive 10 extra points. The best model (smallest error) will receive 25 extra points.

Relationship with Air and Car Industries

4. It is well known that the profit of airline companies is 90% determined by the price of the oil. So we should see a negative association between the two. Same but in lesser extent should be true about the car manufacturing business. So construct two regression models with time series errors relating oil and airline and oil and car industries.

5. Comment on the results obtained.

Mathematical Techniques

This midterm exam has many components. The oil analysis section deals mainly with linear time series analysis, in the form of autoregressive (AR) models, moving average (MA) models, autoregressive moving average (ARMA) models, and seasonal models. The bonus section to the oil analysis involves forecasting. The last part of this exam which deals with the air and car industries makes use of regression models with time series errors.

Methodology

This midterm exam is split into three main parts: data gathering, oil analysis, and the relationship with air and car industries.

First, the data will be downloaded from the Yahoo! website, and imported into R. Next, vectors of continuously compounded returns will be generated for each of the fifteen (15) series. Finally, three indexes will be generated, one for each industry: oil and gas, airline, and automotive industries.

For the oil analysis, five models will be generated to see which best fits the model. An AR, MA, ARMA, and two seasonal models will be generated. The coefficients of all models will be estimated and diagnostic plots will be generated to examine the residuals.

For the final part of this exam, two regression models with time series errors will be generated; one to model the relationship between the oil and airline industries, and one to model the relationship between the oil and automotive industries. Most importantly, the residuals of the models will be tested for correlation, normality, and any patterns. If significance within the residuals is found, they will be fit to another model in order to generate the most accurate regression model.

Application of Methodology

Data Gathering

First, the data for each industry was downloaded from the Yahoo! Industry Center website. The following are the different industries and stocks used in this analysis:

1. Major Integrated Oil and Gas
    - Exxon Mobil CP (XOM)
    - Royal Dutch Shell (RDS-B)
    - Petrochina Co Ads (PTR)
    - BP LPC (BP)
    - Chevron Corp (CVX)
2. Major Airlines
    - TAM S.A. Ads (TAM)

- China South Air Ltd (ZNH)
- Delta Air Lines New (DAL)
- China Eastern Airline (CEA)
- Northwest Airline (NWA)

3. Auto Manufacturers – Major
    - Toyota MTR CP Ads (TM)
    - Honda Motor Co Adr (HMC)
    - Daimler AG (DAI)
    - Ford Motor Co (F)
    - Gen Motors (GM)

The data was then imported into R. Next, the data was converted into vectors of simple returns by reversing the data (since the data from Yahoo! has the most recent data first) and then generating a vector of continuously compounded returns (log returns). The following code is an example of how the data was converted into a vector of log returns.

```
> BPrev=rev(GasBP$AdjClose)
> BP.Ret=returns(BPrev,method="compound")[-1]
```

Then, a new vector was created which was equal to the average of the five return vectors for each industry. The three indexes are as follows:

```
> GasIndex=(BP.Ret+Exxon.Ret+Shell.Ret+Chevron.Ret+PChina.Ret)/5
> AirIndex=(CEA.Ret+Delta.Ret+NWA.Ret+ZNH.Ret+TAM.Ret)/5
> AutoIndex=(DAI.Ret+GM.Ret+TM.Ret+Ford.Ret+HMC.Ret)/5
```

Oil Analysis

Next, the Gas Index was used to construct time series models. The models generated included an AR, MA, ARMA, and various seasonal models, which are discussed below.

<u>AR Model</u>

First, the autocorrelation function (ACF) of the Gas Index series was generated, and is shown below in Figure 1. The ACF does not show any trends of a unit root.

The following code was used to fit the Gas Index series to an AR model to check for unit root non-stationarity:

```
> GasAR1=ar(GasIndex)
```

This AR model produced a coefficient (ϕ_1) of -0.1167, with σ^2 estimated as 0.0001956. The ADF Test results show a Dickey-Fuller (DF) statistic of -15.819, with a p-value less than 0.01, which indicates no unit root.

This series was then fit to an ARIMA model using the following code:

> GasAR=arima(GasIndex,order=c(1,0,0),include.mean=F)

This AR model produced a coefficient (ϕ_1) of -0.1161, with σ^2 estimated as 0.0001951.

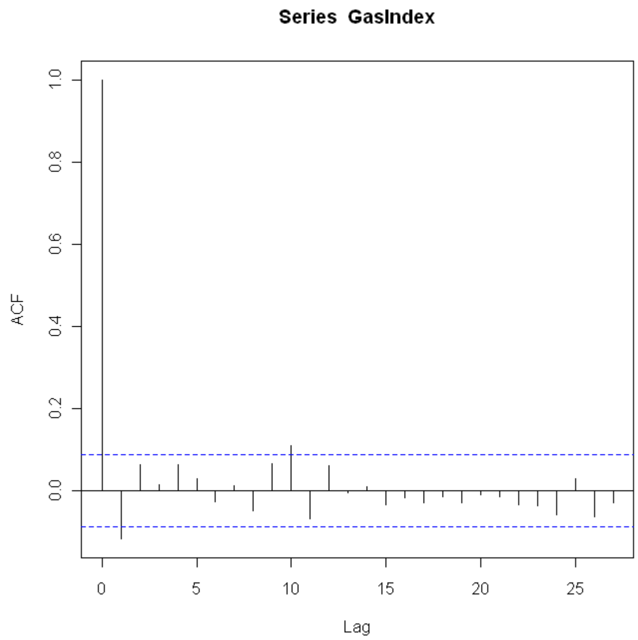

Figure 1: ACF for Gas Index AR Model

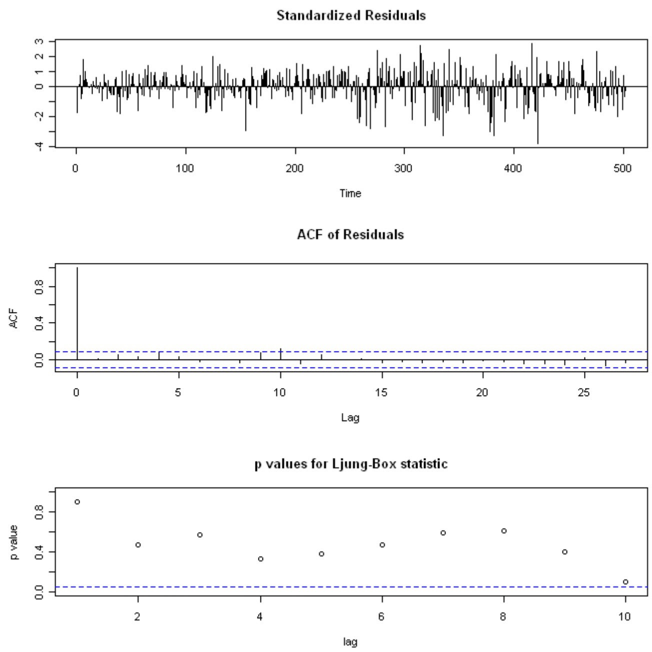

Figure 2: Diagnostic Plots for Gas Index AR Model

Diagnostic plots were then generated for the AR model as shown in Figure 2. These plots do not show patterns and autocorrelation among the residuals. However, in the Ljung-Box chart, the last p-value is very close to being below the confidence intervals. Perhaps an MA model would be a better choice.

MA Model

To generate an MA model, the Gas Index series was then fit to an ARIMA model using the following code:

> GasMA=arima(GasIndex,order=c(0,0,1),include.mean=T)

This MA model produced a coefficient (θ_1) of -0.1043, an intercept of 4E-04 with σ^2 estimated as 0.0001951.

Diagnostic plots were then generated for the MA model as shown in Figure 3. These plots do not show patterns and autocorrelation among the residuals. However, in the Ljung-Box chart, the last p-value is very close to being below the confidence intervals. Perhaps an ARMA model would be a better choice.

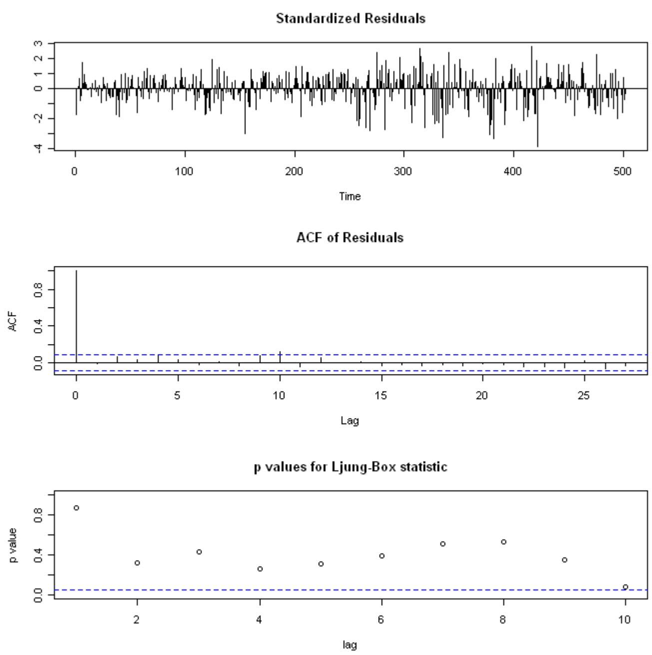

Figure 3: Diagnostic Plots for Gas Index MA Model

ARMA Model

To generate an ARMA model, the lowest AIC matrix must me generated using the following code:

```
> .q=15
> model.aic=matrix(,nrow=max.p+1,ncol=max.q+1)
 >for (p in 0:max.p){
>   for (q in 0:max.q){
>   model.aic[p+1,q+1]=arima(GasIndex,order=c(p,0,q))$aic
>   }}
> print(model.aic)
```

After the AIC matrix has printed, the following code was used to find the location of the lowest AIC criterion:

```
> model.aic[is.na(model.aic)]=1000
> which.min(model.aic)
> which(model.aic==min(model.aic),arr.ind=T)
```

For this model, the lowest AIC was found in row 2, column 1, which implies an ARMA (1,0) model, which is the same as an AR (1) model. The following code was used to generate the ARMA (1,0) model:

```
> GasARMA=arima(GasIndex,order=c(1,0,0),include.mean=F)
```

The coefficient and variance of the model is the same as the AR model, which is expected.

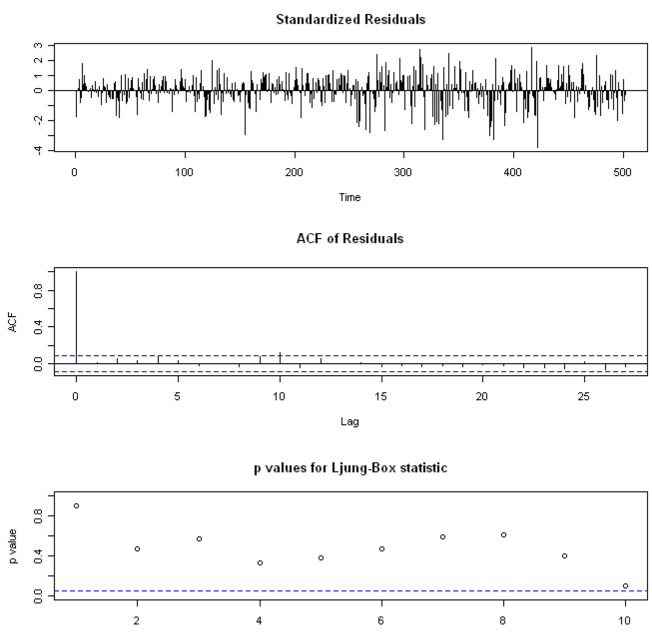

Figure 4: Diagnostic Plots for Gas Index ARMA Model

The diagnostic plots shown above look identical to the AR plots already presented. Perhaps a seasonal model would model the series more accurately.

Seasonal Models

Two seasonal models were generated: a quarterly model and a monthly model. Before the models were generated, a vector of the average of the five stock price vectors had to be generated, using the following code:

> *> GasIndexSeasonal=(BPrev+Shellrev+Exxonrev+PChinarev+Chevronrev)/5*

The following is a plot of the GasIndexSeasonal series. A seasonal pattern is obviously seen.

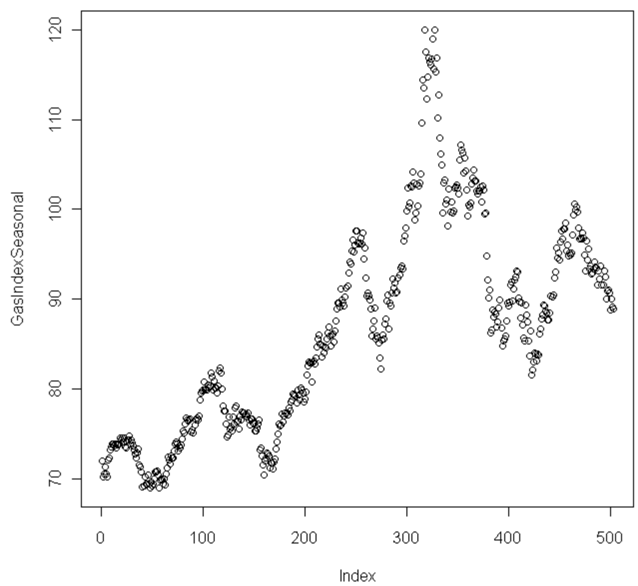

Figure 5: Plot of GasIndexSeasonal for Seasonal Modeling

The ACF of the GasIndexSeasonal series was generated and is shown below in Figure 6. The pattern of the ACF shows the presence of a unit root.

An ADF Test was performed on the GasIndexSeasonal series, and the DF statistic was found to be 0.4247, with a p-value of 0.7517. This indicated a unit root is present.

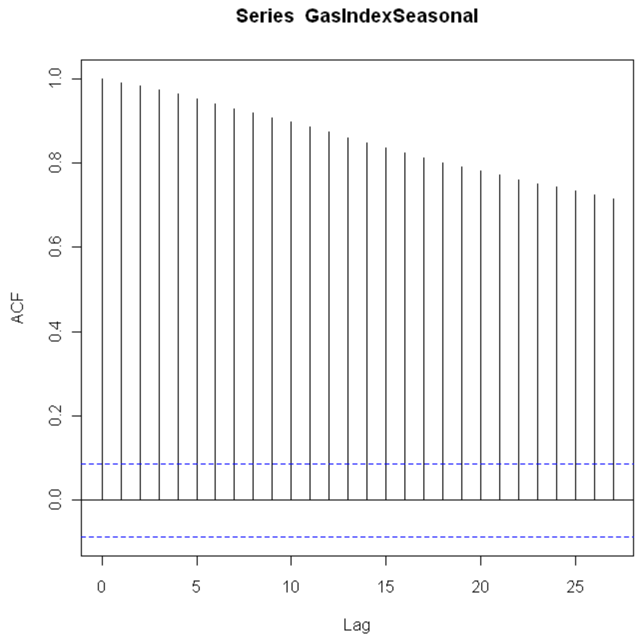

Figure 6: ACF for GasIndexSeasonal Series

<u>Quarterly Model</u>

The following code was used to generate quarterly data from the Gas Index series:

```
> Quarterly=seq(1,length(GasIndexSeasonal),by=63)
> GasIndexSeasonal=GasIndexSeasonal[Quarterly]
> GasIndexS.log=log(GasIndexSeasonal)
```

An ADF test was then conducted for the GasIndexS.log series. The DF statistic was found to be 0.7034, with a p-value of 0.8405. Again, this indicates that a unit root is present.

To remove the unit root, the series was differenced using the following code:

```
> Quart.diff=diff(GasIndexS.log)
```

An ADF test was conducted on the Quart.diff series. The DF statistic was found to be -15.7398, with a p-value less than 0.01. This indicates that the unit root has been removed.

Finally, the Quart.diff series was fit to a quarterly seasonal ARMA(1,1) model using the following code:

```
> seasonalquart=arima(Quart.diff,order=c(1,0,1),
seasonal=list(order=c(1,0,1),period=4))
```

The following are the coefficients of the seasonal model.

Table 1: Coefficients of GasIndex Quarterly Seasonal Model

ϕ_1	θ_1	SAR1	SMA1	Intercept	σ^2
-0.3364	0.2055	-0.6223	0.7146	4E-04	0.0002424

The following figure shows the diagnostic plots for the above quarterly seasonal model. There does not appear to be patterns or autocorrelation among the residuals, and the Ljung-Box plot shows higher p-values than the AR, MA, and ARMA models. Perhaps a monthly seasonal model would be best.

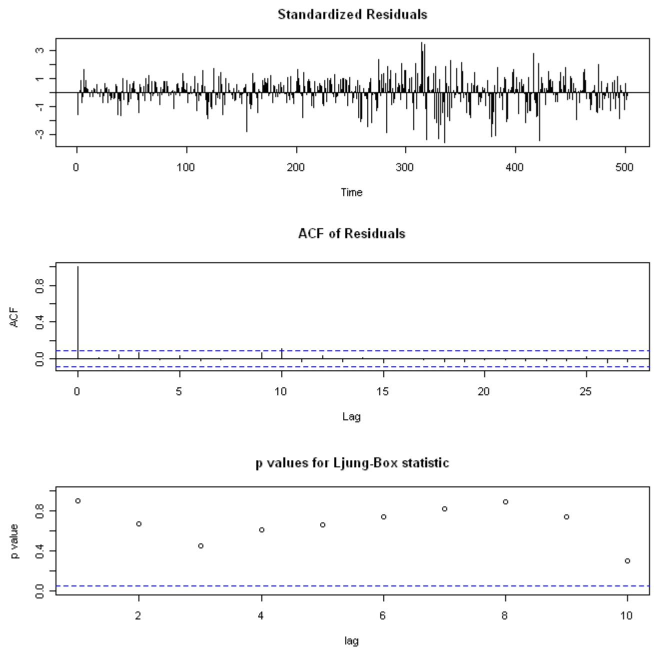

Figure 7: Diagnostic Plots for GasIndex Quarterly Seasonal Model

Monthly Seasonal Model

The following code was used to generate monthly data from the Gas Index series:

```
> Monthly=seq(1,length(GasIndexSeasonal),by=21)
> GasIndexSeasonal=GasIndexSeasonal[Monthly]
> GasIndexS.log=log(GasIndexSeasonal)
```

An ADF test was then conducted for the GasIndexS.log series. The DF statistic was found to be 0.6031, with a p-value of 0.7994. Again, this indicates that a unit root is present.

To remove the unit root, the series was differenced using the following code:

```
> Month.diff=diff(GasIndexS.log)
```

An ADF test was conducted on the Quart.diff series. The DF statistic was found to be -3.9764, with a p-value less than 0.01. This indicates that the unit root has been removed.

Finally, the Month.diff series was fit to a quarterly seasonal ARMA(1,1) model using the following code:

> *seasonalmonth=arima(Month.diff,order=c(1,0,1),*
> *seasonal=list(order=c(1,0,1),period=12))*

The following are the coefficients of the seasonal model.

Table 2: Coefficients of GasIndex Quarterly Seasonal Model

ϕ_1	θ_1	SAR1	SMA1	Intercept	σ^2
0.6011	-1.000	0.6801	-0.6251	0.0141	0.005371

The following figure shows the diagnostic plots for the above quarterly seasonal model. There does not appear to be patterns or autocorrelation among the residuals, and the Ljung-Box plot shows higher p-values than the AR, MA, ARMA, and quarterly seasonal models. This proves to be the best model.

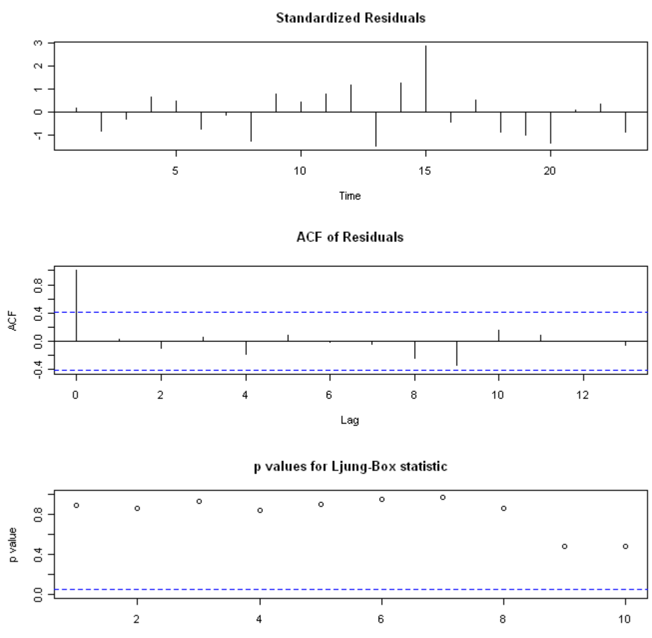

Figure 8: Diagnostic Plots for GasIndex Monthly Seasonal Model

Relationship with Air and Car Industries

In this analysis, two regression models will be generated: one relating the oil and airline industries, and one relating the oil and car industries.

Oil and Airline Industries

First, the Gas Index was plotted against the Air Index, as shown below. This plot does not show much of a linear trend between the two industries; however, let's try a linear regression.

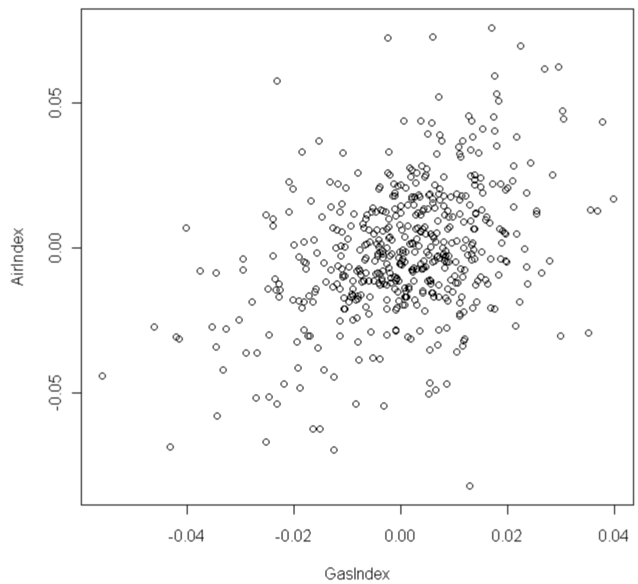

Figure 9: Air Index vs. Gas Index

The following code was used to generate a linear regression model for the Air Index and Gas Index series:

> model1=lm(AirIndex~GasIndex)

The summary statistics for this regression model can be found in Appendix B. The Multiple R^2 value of this model is 0.1584, and the Adjusted R^2 value is 0.1567, which are both extremely small.

The following are residual plots of the regression model. You can see that the residuals may need to be fit to another model in order to make the regression model more accurate.

The residuals were also tested for unit root non-stationarity using the ADF Test. The DF statistic was found to be -15.8277, with a p-value less than 0.01, meaning no unit root.

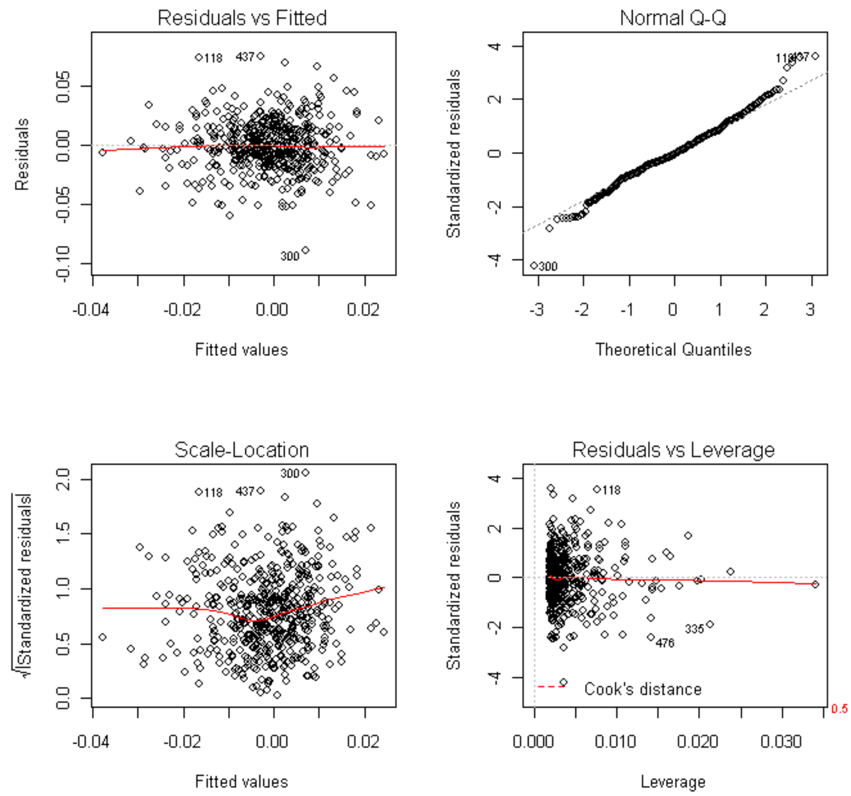

Figure 10: Residual Regression Plots for Air/Oil Regression

Therefore, the residuals of this regression model were first fit to an ARMA model. The lowest AIC criterion matrix was generated using the residuals from the regression model and the minimum AIC criterion location was used to determine the order of the ARMA model (as seen in the previous section). Using the following code, an ARMA (1,6) model was generated to model the residuals of the regression model:

```
> model1ARMA=arima(model1$residuals,order=c(1,0,6),include.mean=F)
```

The following show the coefficients of the ARMA (1,6) model.

Table 3: Coefficients for ARMA (1,6) of Residuals of Regression of Air/Oil

ϕ_1	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	σ^2
-0.7721	0.8811	0.0232	-0.0250	-0.0031	-0.1361	-0.1806	0.0004221

The following plot shows the diagnostic plots of the ARMA (1,6) model. There does not seem to be patterns and autocorrelations among the residuals, and the Ljung-Box plot has very high p-values. This seems to be a very good model for the residuals of the regression.

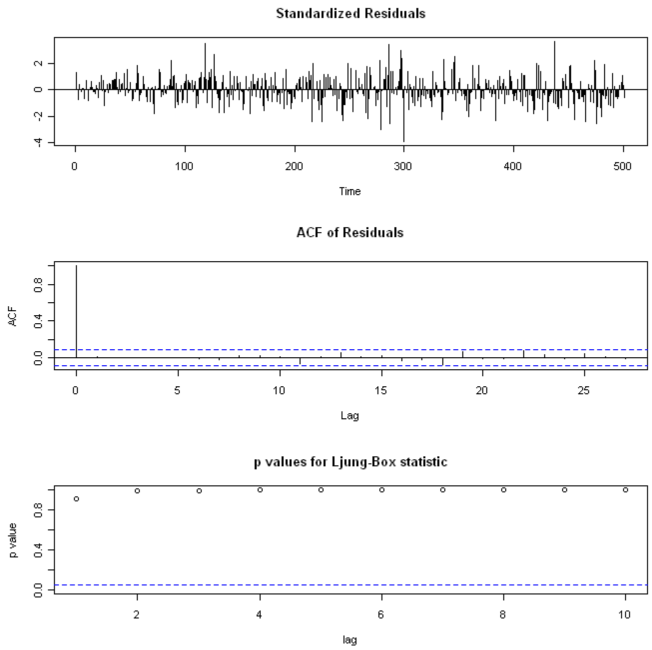

Figure 11: Diagnostic Residual Plots for Air/Oil Regression

Oil and Automotive Industries

First, the Gas Index was plotted against the Auto Index, as shown below. This plot does not show much of a linear trend between the two industries; however, let's try a linear regression.

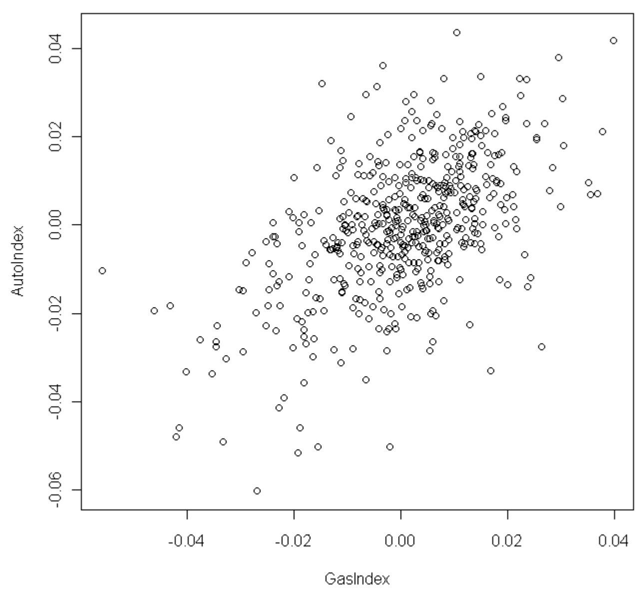

Figure 12: Auto Index vs. Gas Index

The following code was used to generate a linear regression model for the Auto Index and Gas Index series:

> model2=lm(AutoIndex~GasIndex)

The summary statistics for this regression model can be found in Appendix B. The Multiple R^2 value of this model is 0.2913, and the Adjusted R^2 value is 0.2899, which are both extremely small.

The following are residual plots of the regression model. You can see that the residuals may need to be fit to another model in order to make the regression model more accurate.

The residuals were also tested for unit root non-stationarity using the ADF Test. The DF statistic was found to be -13.7795, with a p-value less than 0.01, meaning no unit root.

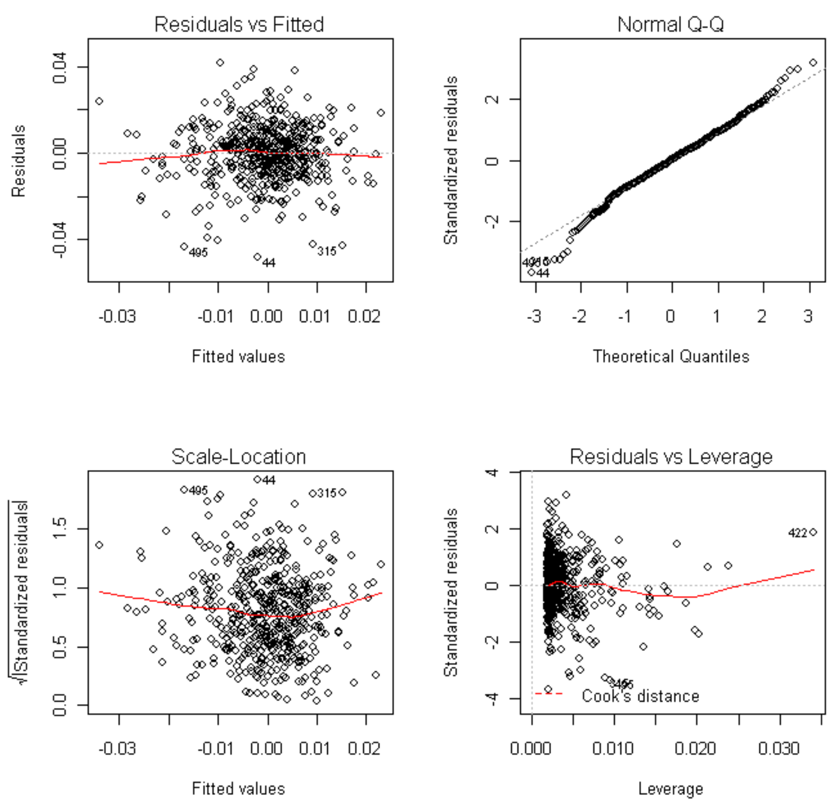

Figure 13: Residual Regression Plots for Auto/Oil Regression

Therefore, the residuals of this regression model were first fit to an ARMA model. The lowest AIC criterion matrix was generated using the residuals from the regression model and the minimum AIC criterion location was used to determine the order of the ARMA model (as seen in the previous section). Using the following code, an ARMA (2,0) model was generated to model the residuals of the regression model:

> *model2ARMA=arima(model2$residuals,order=c(2,0,0),include.mean=F)*

The following tables shows the coefficients for the ARMA (2,0,0) model.

Table 4: Coefficients for ARMA (2,0) of Residuals of Regression of Auto/Oil

ϕ_1	ϕ_2	σ^2
0.0894	0.0810	0.0001684

The following plot shows the diagnostic plots of the ARMA (1,6) model. There does not seem to be patterns and autocorrelations among the residuals, and the Ljung-Box plot has very high p-values. This seems to be a very good model for the residuals of the regression.

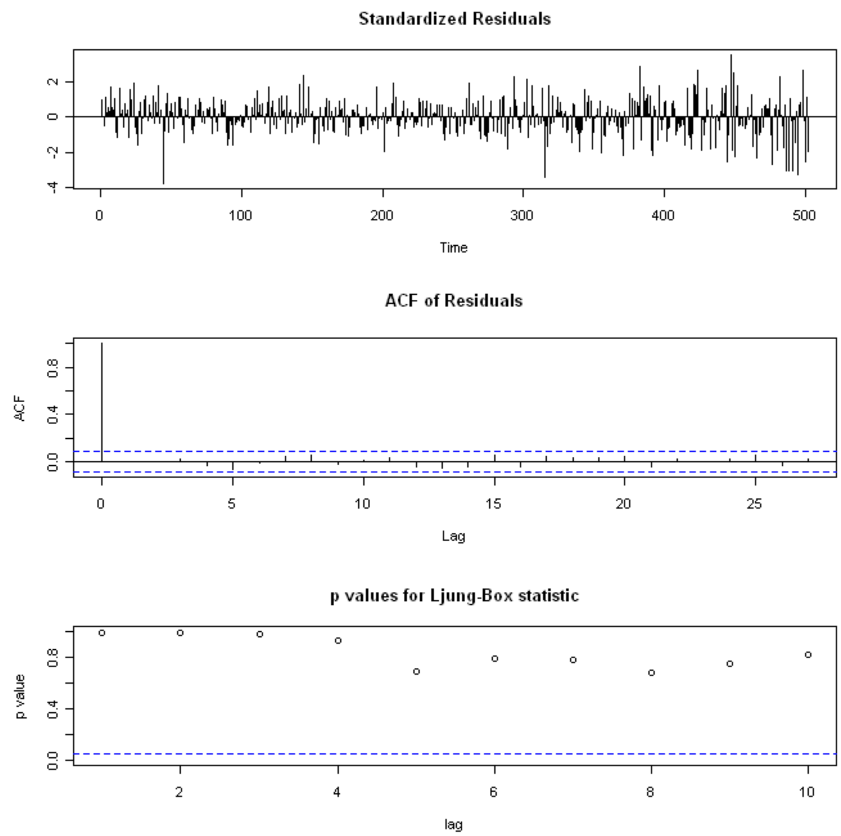

Figure 14: Diagnostic Residual Plots for Auto/Oil Regression

Conclusions

Oil Analysis

A total of five models were generated to model the Gas Index series: an AR, MA, ARMA, quarterly seasonal, and monthly seasonal model. By examining the diagnostic plots for all models, it is clear that both seasonal models out-performed the AR, MA, and

ARMA models. The monthly seasonal model would be considered the best model. The p-values of the Ljung-Box diagnostic plot are the highest of all the models tested. The final model chosen (monthly seasonal model) has the following coefficients:

ϕ_1	θ_1	SAR1	SMA1	Intercept	σ^2
0.6011	-1.000	0.6801	-0.6251	0.0141	0.005371

Relationship with Air and Car Industries

Oil and Airline Industries

The final regression model for the relationship between the oil and airline industries is as follows:

$$r_{AIR} = 0.6496999 r_{GAS} + e_t$$

$$e_t = -0.7721 r_{t-1} + a_t - 0.8811 a_{t-1} - 0.0232 a_{t-2} + 0.0250 a_{t-3} + 0.0031 a_{t-4} + 0.1361 a_{t-5} + 0.1806 a_{t-6}$$

The p-value of the intercept of the regression model was found not to be significant, and hence, was not added to the model.

Oil and Automotive Industries

The final regression model for the relationship between the oil and automotive industries is as follows:

$$r_{AUTO} = 0.5966725 r_{GAS} + e_t$$

$$e_t = 0.0894 r_{t-1} + 0.0810 r_{t-2} + a_t$$

The p-value of the intercept of the regression model was found not to be significant, and hence, was not added to the model.

It is interesting that both regression models have close coefficients, and the intercept is not significant. The major differences in these two models lie in the residuals. The residuals of the regression model for the oil/airline industries follows an ARMA (1,6) model, and the residuals for the oil/auto industries follows an ARMA (2,0) or AR (2) model. However, while the problem statement said there should be a negative association seen in both regression models, both models show a positive association.

Appendix A: Code for Oil Analysis

```
> #Midterm exam, due July 15, 2008
> #by Alicia M. Mahon
>
> library(Rmetrics)
>
> #Import gasoline files
> GasBP<-read.csv("F:/MA641/Midterm/MajIntOil_Gas/BP.csv",header=T)
> GasExxon<-read.csv("F:/MA641/Midterm/MajIntOil_Gas/Exxon.csv",header=T)
> GasShell<-read.csv("F:/MA641/Midterm/MajIntOil_Gas/Shell.csv",header=T)
> GasChevron<-read.csv("F:/MA641/Midterm/MajIntOil_Gas/Chevron.csv",header=T)
> GasPChina<-read.csv("F:/MA641/Midterm/MajIntOil_Gas/Petrochina.csv",header=T)
>
> #Import airline files
> AirCEA<-read.csv("F:/MA641/Midterm/MajAirlines/CEA.csv",header=T)
> AirDelta<-read.csv("F:/MA641/Midterm/MajAirlines/Delta.csv",header=T)
> AirNWA<-read.csv("F:/MA641/Midterm/MajAirlines/NWA.csv",header=T)
> AirZNH<-read.csv("F:/MA641/Midterm/MajAirlines/ZNH.csv",header=T)
> AirTAM<-read.csv("F:/MA641/Midterm/MajAirlines/TAM.csv",header=T)
>
> #Import auto files
> AutoDAI<-read.csv("F:/MA641/Midterm/Auto/DAI.csv",header=T)
> AutoGM<-read.csv("F:/MA641/Midterm/Auto/GM.csv",header=T)
> AutoTM<-read.csv("F:/MA641/Midterm/Auto/TM.csv",header=T)
> AutoFord<-read.csv("F:/MA641/Midterm/Auto/Ford.csv",header=T)
> AutoHMC<-read.csv("F:/MA641/Midterm/Auto/HMC.csv",header=T)
>
> #Returns for gas stocks
> BPrev=rev(GasBP$AdjClose)
> Exxonrev=rev(GasExxon$AdjClose)
> Shellrev=rev(GasShell$AdjClose)
> Chevronrev=rev(GasChevron$AdjClose)
> PChinarev=rev(GasPChina$AdjClose)
>
> BP.Ret=returns(BPrev,method="compound")[-1]
> Exxon.Ret=returns(Exxonrev,method="compound")[-1]
> Shell.Ret=returns(Shellrev,method="compound")[-1]
> Chevron.Ret=returns(Chevronrev,method="compound")[-1]
> PChina.Ret=returns(PChinarev,method="compound")[-1]
>
> #Returns for airline stocks
> CEArev=rev(AirCEA$AdjClose)
> Deltarev=rev(AirDelta$AdjClose)
> NWArev=rev(AirNWA$AdjClose)
> ZNHrev=rev(AirZNH$AdjClose)
> TAMrev=rev(AirTAM$AdjClose)
>
> CEA.Ret=returns(CEArev,method="compound")[-1]
```

Appendix A: Code for Oil Analysis

```
> Delta.Ret=returns(Deltarev,method="compound")[-1] #Not full data set
> NWA.Ret=returns(NWArev,method="compound")[-1] #Not full data set
> ZNH.Ret=returns(ZNHrev,method="compound")[-1]
> TAM.Ret=returns(TAMrev,method="compound")[-1]
>
> #Returns for auto stocks
> DAIrev=rev(AutoDAI$AdjClose)
> GMrev=rev(AutoGM$AdjClose)
> TMrev=rev(AutoTM$AdjClose)
> Fordrev=rev(AutoFord$AdjClose)
> HMCrev=rev(AutoHMC$AdjClose)
>
> DAI.Ret=returns(DAIrev,method="compound")[-1]
> GM.Ret=returns(GMrev,method="compound")[-1]
> TM.Ret=returns(TMrev,method="compound")[-1]
> Ford.Ret=returns(Fordrev,method="compound")[-1]
> HMC.Ret=returns(HMCrev,method="compound")[-1]
>
> #Gas index
> GasIndex=(BP.Ret+Exxon.Ret+Shell.Ret+Chevron.Ret+PChina.Ret)/5
>
> #Airline index
> AirIndex=(CEA.Ret+Delta.Ret+NWA.Ret+ZNH.Ret+TAM.Ret)/5
Warning messages:
1: In CEA.Ret + Delta.Ret :
  longer object length is not a multiple of shorter object length
2: In CEA.Ret + Delta.Ret + NWA.Ret :
  longer object length is not a multiple of shorter object length
>
> #Auto index
> AutoIndex=(DAI.Ret+GM.Ret+TM.Ret+Ford.Ret+HMC.Ret)/5
>
> #OIL ANALYSIS
>
> #AR Model
> GasAR1=ar(GasIndex)
> GasAR1

Call:
ar(x = GasIndex)

Coefficients:
      1
-0.1167

Order selected 1  sigma^2 estimated as  0.0001956
> adfTest(GasIndex)
```

Appendix A: Code for Oil Analysis

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -15.819
  P VALUE:
    0.01

Description:
 Tue Jul 15 13:37:19 2008 by user: Alicia Mahon

Warning message:
In adfTest(GasIndex) : p-value smaller than printed p-value
> GasAR=arima(GasIndex,order=c(1,0,0),include.mean=F)
> GasAR

Call:
arima(x = GasIndex, order = c(1, 0, 0), include.mean = F)

Coefficients:
          ar1
      -0.1161
s.e.   0.0444

sigma^2 estimated as 0.0001951:  log likelihood = 1431.79,  aic = -2859.57
> acf(GasIndex)
> tsdiag(GasAR)
>
> #MA Model
> GasMA=arima(GasIndex,order=c(0,0,1),include.mean=T)
> GasMA

Call:
arima(x = GasIndex, order = c(0, 0, 1), include.mean = T)

Coefficients:
          ma1  intercept
      -0.1043      4e-04
s.e.   0.0415      6e-04

sigma^2 estimated as 0.0001951:  log likelihood = 1431.68,  aic = -2857.37
> tsdiag(GasMA)
>
```

Appendix A: Code for Oil Analysis

```
> #ARMA Model
> max.p=15
> max.q=15
> model.aic=matrix(,nrow=max.p+1,ncol=max.q+1)
> for (p in 0:max.p){
+   for (q in 0:max.q){
+   model.aic[p+1,q+1]=arima(GasIndex,order=c(p,0,q))$aic
+   }
+ }                  ## takes forever
Error in optim(init[mask], armafn, method = "BFGS", hessian = TRUE, control = optim.control,  :
  non-finite finite-difference value [12]
In addition: There were 40 warnings (use warnings() to see them)
>  print(model.aic)
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
 [1,] -2853.203 -2857.365 -2857.597 -2856.093 -2857.216 -2855.270 -2854.036 -2852.060 -2851.333
 [2,] -2858.119 -2857.092 -2857.775 -2855.813 -2855.243 -2853.728 -2852.029 -2850.036 -2850.175
 [3,] -2857.441 -2857.780 -2855.825 -2853.807 -2855.957 -2854.137 -2856.001 -2852.499 -2853.902
 [4,] -2855.906 -2855.808 -2853.789 -2851.993 -2854.229 -2855.356 -2853.194 -2855.261 -2853.785
 [5,] -2856.165 -2854.656 -2855.898 -2854.048 -2857.841 -2850.141 -2852.802 -2853.417 -2851.484
 [6,] -2855.131 -2853.586 -2854.073 -2852.229 -2856.164 -2854.107 -2853.715 -2851.812 -2849.520
 [7,] -2853.500 -2851.535 -2849.586 -2850.124 -2854.881 -2852.659 -2854.445 -2852.594 -2849.866
 [8,] -2851.501 -2849.709 -2847.697 -2848.203 -2851.084 -2850.679 -2849.289 -2849.827 -2853.008
 [9,] -2850.760 -2848.979 -2852.621 -2850.752 -2850.544 -2848.809 -2846.849 -2851.331 -2849.421
[10,] -2850.205 -2851.717 -2851.570 -2850.854 -2850.998 -2849.234 -2848.143 -2847.193 -2851.303
[11,] -2857.727 -2857.155 -2855.177 -2853.288 -2851.288 -2849.863 -2848.012 -2846.261 -2844.561
[12,] -2856.756 -2855.166 -2853.208 -2851.284 -2853.794 -2851.794 -2846.111 -2844.199 -2845.615
[13,] -2855.374 -2853.376 -2851.545 -2854.072 -2850.212 -2849.908 -2846.527 -2845.335 -2840.134
[14,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[15,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[16,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
         [,10]     [,11]     [,12]     [,13]     [,14]     [,15]     [,16]
 [1,] -2853.888 -2856.166 -2855.537 -2856.306 -2854.511 -2853.155 -2851.739
 [2,] -2854.717 -2857.492 -2855.493 -2854.652 -2852.654 -2851.537 -2849.720
 [3,] -2852.983 -2855.493 -2853.504 -2852.663 -2850.657 -2849.701 -2847.697
 [4,] -2853.421 -2854.067 -2852.180 -2851.402 -2849.577 -2847.713 -2850.356
 [5,] -2852.054 -2852.283 -2850.290 -2849.598 -2847.584 -2848.462 -2850.352
 [6,] -2851.920 -2850.367 -2849.284 -2849.140 -2852.112 -2846.252 -2842.159
 [7,] -2850.256 -2853.635 -2849.176 -2847.284 -2851.601 -2849.432 -2844.756
 [8,] -2848.279 -2850.155 -2850.723 -2849.146 -2847.048 -2847.897 -2842.742
 [9,] -2851.241 -2847.095 -2845.088 -2848.057 -2847.140 -2849.183 -2840.782
[10,] -2849.110 -2847.920 -2843.123 -2843.536 -2844.618 -2846.872 -2842.352
[11,] -2847.985 -2850.269 -2847.521 -2843.334 -2842.122 -2841.770 -2841.830
[12,] -2845.955 -2848.477 -2838.732 -2841.907 -2841.197 -2842.837 -2841.993
[13,] -2844.955 -2840.594        NA        NA        NA        NA        NA
[14,]        NA        NA        NA        NA        NA        NA        NA
[15,]        NA        NA        NA        NA        NA        NA        NA
[16,]        NA        NA        NA        NA        NA        NA        NA
```

Appendix A: Code for Oil Analysis

```
>
> model.aic[is.na(model.aic)]=1000
> which.min(model.aic)
[1] 2
> which(model.aic==min(model.aic),arr.ind=T)
     row col
[1,]   2   1
>
> GasARMA=arima(GasIndex,order=c(1,0,0),include.mean=F)
> GasARMA

Call:
arima(x = GasIndex, order = c(1, 0, 0), include.mean = F)

Coefficients:
          ar1
      -0.1161
s.e.   0.0444

sigma^2 estimated as 0.0001951:  log likelihood = 1431.79,  aic = -2859.57
> tsdiag(GasARMA)
>
> #Seasonal models
> #Quarterly seasonal model
> GasIndexSeasonal=(BPrev+Shellrev+Exxonrev+PChinarev+Chevronrev)/5
> acf(GasIndexSeasonal)
> adfTest(GasIndexSeasonal)

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: 0.4247
  P VALUE:
    0.7517

Description:
 Tue Jul 15 14:05:14 2008 by user: Alicia Mahon

> plot(GasIndexSeasonal)
> Quarterly=seq(1,length(GasIndexSeasonal),by=63)
> GasIndexSeasonal=GasIndexSeasonal[Quarterly]
> GasIndexS.log=log(GasIndexSeasonal)
> adfTest(GasIndexS.log)
```

Appendix A: Code for Oil Analysis

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: 0.4791
  P VALUE:
    0.76

Description:
 Tue Jul 15 14:05:14 2008 by user: Alicia Mahon

> acf(GasIndexS.log)
> Quart.diff=diff(GasIndexS.log)
> adfTest(Quart.diff)

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -1.4184
  P VALUE:
    0.1576

Description:
 Tue Jul 15 14:05:14 2008 by user: Alicia Mahon

> seasonalquart=arima(Quart.diff,order=c(1,0,1),seasonal=list(order=c(1,0,1),period=4))
> seasonalquart

Call:
arima(x = Quart.diff, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 1),
    period = 4))

Coefficients:
          ar1     ma1      sar1    sma1  intercept
      -0.4821  1.0000  -0.9841  0.8907     0.0318
s.e.   0.4398  0.6102   0.7394  2.5186     0.0435

sigma^2 estimated as 0.008022:  log likelihood = 5.51,  aic = 0.98
> tsdiag(seasonalquart)
```

Appendix A: Code for Oil Analysis

```
>
> #Monthly seasonal model
> GasIndexSeasonal=(BPrev+Shellrev+Exxonrev+PChinarev+Chevronrev)/5
> adfTest(GasIndexSeasonal)

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: 0.4247
  P VALUE:
    0.7517

Description:
 Tue Jul 15 14:05:14 2008 by user: Alicia Mahon

> plot(GasIndexSeasonal)
> Monthly=seq(1,length(GasIndexSeasonal),by=21)
> GasIndexSeasonal=GasIndexSeasonal[Monthly]
> acf(GasIndexSeasonal)
> GasIndexS.log=log(GasIndexSeasonal)
> adfTest(GasIndexS.log)

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: 0.6031
  P VALUE:
    0.7994

Description:
 Tue Jul 15 14:05:14 2008 by user: Alicia Mahon

> acf(GasIndexS.log)
> Month.diff=diff(GasIndexS.log)
> adfTest(Month.diff)

Title:
 Augmented Dickey-Fuller Test
```

Appendix A: Code for Oil Analysis

```
Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -3.9764
  P VALUE:
    0.01

Description:
 Tue Jul 15 14:05:14 2008 by user: Alicia Mahon

Warning message:
In adfTest(Month.diff) : p-value smaller than printed p-value
> seasonalmonth=arima(Month.diff,order=c(1,0,1),seasonal=list(order=c(1,0,1),period=12))
> seasonalmonth

Call:
arima(x = Month.diff, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 1),
    period = 12))

Coefficients:
         ar1      ma1    sar1     sma1  intercept
      0.6011  -1.0000  0.6801  -0.6251     0.0141
s.e.  0.1854   0.1211  9.3414   9.3547     0.0047

sigma^2 estimated as 0.005371:  log likelihood = 26.51,  aic = -41.02
> tsdiag(seasonalmonth)
>
```

Appendix B: Code for Regression Modeling

```
> #Relationship with air and car industries
> #Oil and Air
> plot(GasIndex,AirIndex)
> model1=lm(AirIndex~GasIndex)
> summary(model1)

Call:
lm(formula = AirIndex ~ GasIndex)

Residuals:
      Min        1Q    Median        3Q       Max
-0.089157 -0.012260 -0.001102  0.013106  0.075655

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0014844  0.0009417  -1.576    0.116
GasIndex     0.6496999  0.0669714   9.701   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02109 on 500 degrees of freedom
Multiple R-squared: 0.1584,     Adjusted R-squared: 0.1567
F-statistic: 94.11 on 1 and 500 DF,  p-value: < 2.2e-16

> par(mfrow=c(2,2))
> plot(model1)
> plot(model1$residuals)
> acf(model1$residuals)
> adfTest(model1$residuals)

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -15.8277
  P VALUE:
    0.01

Description:
 Tue Jul 15 14:08:41 2008 by user: Alicia Mahon

Warning message:
In adfTest(model1$residuals) : p-value smaller than printed p-value
>
```

Appendix B: Code for Regression Modeling

```
> #Model residuals using ARMA Model
> max.p=15
> max.q=15
> model.aic=matrix(,nrow=max.p+1,ncol=max.q+1)
> for (p in 0:max.p){
+   for (q in 0:max.q){
+   model.aic[p+1,q+1]=arima(model1$residuals,order=c(p,0,q))$aic
+   }
+ }                    ## takes forever
Error in arima(model1$residuals, order = c(p, 0, q)) :
  non-stationary AR part from CSS
>  print(model.aic)
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
 [1,] -2447.773 -2452.031 -2451.384 -2449.411 -2447.567 -2452.075 -2455.300 -2454.330 -2452.765
 [2,] -2451.332 -2451.296 -2449.399 -2447.387 -2449.830 -2452.867 -2457.741 -2456.509 -2454.509
 [3,] -2451.086 -2449.429 -2449.858 -2447.941 -2453.246 -2453.704 -2456.423 -2454.508 -2452.578
 [4,] -2449.421 -2447.511 -2456.190 -2452.216 -2454.524 -2451.703 -2454.687 -2453.084 -2452.803
 [5,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
 [6,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
 [7,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
 [8,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
 [9,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[10,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[11,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[12,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[13,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[14,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[15,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[16,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
         [,10]     [,11]     [,12]     [,13]     [,14]     [,15]     [,16]
 [1,] -2453.945 -2452.443 -2451.012 -2449.491 -2449.205 -2447.222 -2445.789
 [2,] -2452.256 -2450.308 -2449.107 -2449.348 -2448.387 -2446.424 -2445.075
 [3,] -2450.833 -2449.687 -2447.575 -2453.381 -2446.560 -2444.571 -2445.053
 [4,] -2457.071 -2454.857 -2451.011 -2451.897        NA        NA        NA
 [5,]        NA        NA        NA        NA        NA        NA        NA
 [6,]        NA        NA        NA        NA        NA        NA        NA
 [7,]        NA        NA        NA        NA        NA        NA        NA
 [8,]        NA        NA        NA        NA        NA        NA        NA
 [9,]        NA        NA        NA        NA        NA        NA        NA
[10,]        NA        NA        NA        NA        NA        NA        NA
[11,]        NA        NA        NA        NA        NA        NA        NA
[12,]        NA        NA        NA        NA        NA        NA        NA
[13,]        NA        NA        NA        NA        NA        NA        NA
[14,]        NA        NA        NA        NA        NA        NA        NA
[15,]        NA        NA        NA        NA        NA        NA        NA
[16,]        NA        NA        NA        NA        NA        NA        NA
>
```

Appendix B: Code for Regression Modeling

```
> model.aic[is.na(model.aic)]=1000
> which.min(model.aic)
[1] 98
> which(model.aic==min(model.aic),arr.ind=T)
     row col
[1,]   2   7
>
> model1ARMA=arima(model1$residuals,order=c(1,0,6),include.mean=F)
> model1ARMA

Call:
arima(x = model1$residuals, order = c(1, 0, 6), include.mean = F)

Coefficients:
          ar1     ma1     ma2      ma3      ma4      ma5      ma6
      -0.7721  0.8811  0.0232  -0.0250  -0.0031  -0.1361  -0.1806
s.e.   0.1197  0.1244  0.0600   0.0608   0.0603   0.0560   0.0491

sigma^2 estimated as 0.0004221:  log likelihood = 1237.87,  aic = -2459.74
> tsdiag(model1ARMA)
>
>
> #Oil and Auto
> plot(GasIndex,AutoIndex)
> model2=lm(AutoIndex~GasIndex)
> summary(model2)

Call:
lm(formula = AutoIndex ~ GasIndex)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0482799 -0.0079290  0.0005315  0.0079967  0.0416344

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0006617  0.0005853  -1.131    0.259
GasIndex     0.5966725  0.0416221  14.335   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01311 on 500 degrees of freedom
Multiple R-squared: 0.2913,     Adjusted R-squared: 0.2899
F-statistic: 205.5 on 1 and 500 DF,  p-value: < 2.2e-16

> par(mfrow=c(2,2))
> plot(model2)
```

Appendix B: Code for Regression Modeling

```
> plot(model2$residuals)
> acf(model2$residuals)
> adfTest(model2$residuals)

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -13.7795
  P VALUE:
    0.01

Description:
 Tue Jul 15 14:10:49 2008 by user: Alicia Mahon

Warning message:
In adfTest(model2$residuals) : p-value smaller than printed p-value
>
> #Model residuals using ARMA Model
> max.p=15
> max.q=15
> model.aic1=matrix(,nrow=max.p+1,ncol=max.q+1)
> for (p in 0:max.p){
+   for (q in 0:max.q){
+   model.aic1[p+1,q+1]=arima(model2$residuals,order=c(p,0,q))$aic
+   }
+ }                   ## takes forever
Error in arima(model2$residuals, order = c(p, 0, q)) :
  non-stationary AR part from CSS
In addition: There were 18 warnings (use warnings() to see them)
>  print(model.aic1)
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
 [1,] -2925.310 -2927.401 -2929.249 -2928.069 -2926.243 -2926.826 -2924.999 -2923.459 -2922.653
 [2,] -2928.088 -2928.110 -2927.677 -2925.444 -2925.142 -2924.908 -2923.054 -2922.129 -2920.977
 [3,] -2929.359 -2927.367 -2928.952 -2927.539 -2925.147 -2929.151 -2927.541 -2920.012 -2923.435
 [4,] -2927.382 -2925.454 -2927.869 -2924.988 -2923.132 -2921.499 -2925.698 -2917.939 -2922.096
 [5,] -2926.327 -2925.003 -2926.076 -2921.343 -2921.046 -2920.137 -2924.231 -2922.483 -2922.181
 [6,] -2926.141 -2924.150 -2929.225 -2920.191 -2919.952 -2918.939 -2921.634 -2921.106 -2915.652
 [7,] -2924.164 -2923.330 -2922.944 -2921.066 -2919.155 -2921.319 -2920.714 -2918.073        NA
 [8,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
 [9,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[10,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[11,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
[12,]        NA        NA        NA        NA        NA        NA        NA        NA        NA
```

```
[13,]      NA      NA      NA      NA      NA      NA      NA      NA      NA
[14,]      NA      NA      NA      NA      NA      NA      NA      NA      NA
[15,]      NA      NA      NA      NA      NA      NA      NA      NA      NA
[16,]      NA      NA      NA      NA      NA      NA      NA      NA      NA
        [,10]     [,11]     [,12]     [,13]     [,14]     [,15]     [,16]
 [1,] -2920.989 -2919.255 -2918.032 -2917.078 -2915.999 -2916.030 -2915.387
 [2,] -2918.994 -2917.387 -2917.699 -2916.319 -2913.789 -2916.517 -2914.953
 [3,] -2916.999 -2917.923 -2915.229 -2914.152 -2913.108 -2912.193 -2913.845
 [4,] -2919.166 -2918.173 -2914.861 -2914.314 -2916.752 -2915.418 -2913.709
 [5,] -2920.433 -2917.773 -2916.488 -2911.919 -2914.455 -2909.069 -2906.784
 [6,] -2916.290 -2916.377 -2916.140 -2915.210 -2920.173 -2911.056 -2909.904
 [7,]       NA        NA        NA        NA        NA        NA        NA
 [8,]       NA        NA        NA        NA        NA        NA        NA
 [9,]       NA        NA        NA        NA        NA        NA        NA
[10,]       NA        NA        NA        NA        NA        NA        NA
[11,]       NA        NA        NA        NA        NA        NA        NA
[12,]       NA        NA        NA        NA        NA        NA        NA
[13,]       NA        NA        NA        NA        NA        NA        NA
[14,]       NA        NA        NA        NA        NA        NA        NA
[15,]       NA        NA        NA        NA        NA        NA        NA
[16,]       NA        NA        NA        NA        NA        NA        NA
>
> model.aic1[is.na(model.aic1)]=1000
> which.min(model.aic1)
[1] 3
> which(model.aic1==min(model.aic1),arr.ind=T)
     row col
[1,]   3   1
>
> model2ARMA=arima(model2$residuals,order=c(2,0,0),include.mean=F)
> model2ARMA

Call:
arima(x = model2$residuals, order = c(2, 0, 0), include.mean = F)

Coefficients:
         ar1     ar2
      0.0894  0.0810
s.e.  0.0447  0.0447

sigma^2 estimated as 0.0001684:  log likelihood = 1468.68,  aic = -2931.36
> tsdiag(model2ARMA)
>
```