**Module Assignment**

**Time Series**



Write a simple R script to execute the following data preprocessing and statistical analysis. Where required show analytical output and interpretations.

**Preprocessing**

1. Load the file "6304 Module 7 Data.xlsx" into R. This data shows the number of visitors to the United States from the People's Republic of China on a quarterly basis from years 1998 to 2012. The data shown is scaled in thousands of people.
2. Create a new "index" variable in the data frame which will be an identifying sequential numbering of rows from 1 to the number of rows in the data frame.

**Analysis**

1. Show a plot of the data using the number of visitors as the "y" variable in the plot.
2. Using all the data parameterize a base time series simple regression model using "index" as the independent variable. Show the summary of your regression output.
3. Drawing on Analysis Part 1 above, show a properly titled plot of the time series data with the simple regression line layered on the graph in a contrasting color.
4. Execute and interpret a Durbin-Watson test on your model results.
5. Note the original data appears to have a pronounced cyclical pattern. Assuming the complete cycles are four quarters long, construct a set of seasonal indices which describe the typical annual fluctuations in visitors. Use these indices to deseasonalize the visitors data. Store this deseasonalized data in a column in the original data frame.
6. Using the deseasonalized data parameterize four different regression models. A simple regression model will be the base case to be followed by second order, third order, and fourth order polynomial models which attempt to describe the longer-term secular fluctuations in the deseasonalized data.
7. Reseasonalize the fitted values for each of the four models, storing the reseasonalized values in separate columns in the original data frame. Drawing on Analysis Part 3 above, construct a plot showing the original data and the fitted values for each of the four regression models. Show the four sets of fitted values plots in contrasting colors and title the graph appropriately.
8. Select the model which in your view is the best fit to the deseasonalized data. Give a brief justification as to why you believe your selection is the best fit.

Your deliverable will be a single MS-Word file showing 1) the R script which executes the above preprocessing and analysis instructions and 2) the results of those instructions and needed written interpretations. The first line of your script file should be a “#” comment line showing your name as it appears in Canvas. Results should be presented in the order in which they are listed here. Deliverable due time will be announced in class and on Canvas. This is an individual assignment to be completed before you leave the classroom. No collaboration of any sort is allowed on this assignment.

**Preprocessing:**

**#Varun Teja Kolluru**

**#preprocessing**

**#1**

**rm(list = ls())**

**library(rio)**

**library(moments)**

**library(car)**

**install.packages('readxl')**

**library(readxl)**

**## Loading required package: carData**

**my\_data=import('6304 Module 7 Assignment Data.xlsx')**

**colnames(my\_data)=tolower(make.names(colnames(my\_data)))**

**#2**

**my\_data$index=seq(1:nrow(my\_data))**

**names(my\_data)**

**attach(my\_data)**

Using ‘rm’ command we are clearing all the variables and vectors in the environment window. All the required packages are imported into our R project. Data is imported using the import command.

As per the 2nd question a new ‘index’ variable is added into the data with sequential numbers.

**Analysis:**

1. Show a plot of the data using the number of visitors as the "y" variable in the plot.

**RCode:**

**#Analysis**

**#1**

**plot(index,china.visitors,main = "No of China Visitors",pch=20,type = 'o')**

Chart, line chart, scatter chart

Description automatically generated

In the Time series regression, ‘time’ is taken as x-axis and ‘china.visitors’ is takes in y-axis.

We can see there is a secular trend in the graph and it is a increasing trend. There might be cyclical and seasonal trend as well.

1. Using all the data parameterize a base time series simple regression model using "index" as the independent variable. Show the summary of your regression output.

**RCode:**

**#2**

**output=lm(china.visitors~index,data=my\_data)**

**summary(output)**

Output in Console Window:

> #2

> output=lm(china.visitors~index,data=my\_data)

> summary(output)

Call:

lm(formula = china.visitors ~ index, data = my\_data)

Residuals:

Min 1Q Median 3Q Max

-2273.0 -1071.4 -436.2 739.3 4929.8

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3294.93 477.58 6.899 7.17e-09 \*\*\*

index 67.07 15.11 4.439 4.73e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1730 on 52 degrees of freedom

Multiple R-squared: 0.2748, Adjusted R-squared: 0.2609

F-statistic: 19.71 on 1 and 52 DF, p-value: 4.728e-05

Both the beta Coefficient significant. Intercept value is 3294.92 and slope of time is 67.07.

The Y estimated equation is Y = 3294.92 + 67.07\* index.

As per the Beta Coefficients, in year 1998 for 3rd quarter, number of china visitors are 3294.92, because that is where the regression line starts. And for every quarter, there is 67.07 increase in number of visitors.

Residual standard error or standard deviation is 1730, which is very wide graph.

Multiple R-squared value is 27% and the Adjusted R-squared value is 26% which is not a good value and the data is not following linear regression.

1. Drawing on Analysis Part 1 above, show a properly titled plot of the time series data with the simple regression line layered on the graph in a contrasting color.

**RCode:**

**plot(index,china.visitors,main = "Actual vs Fitted values",pch=20,type = 'o',lwd=2)**

**abline(output,col="red",lwd=3)**

Chart, line chart

Description automatically generated

**Analysis:**

By this graph and by the Adjusted R-square value which we got from the above summary, we can clearly tell that, the data is not following linear regression.

From 0 to 10 value, the graph follows a positive linear shape and from mid 10 to 20 there is a sudden increase in the number of visitor and drop in number of visitor. There might be any reason for this sudden peak. And from mid 20 there is a cyclicality trend in the graph.

1. Execute and interpret a Durbin-Watson test on your model results.

**RCode:**

**library(robustHD)**

**durbinWatsonTest(output)**

**Output in Console Window:**

**> durbinWatsonTest(output)**

**lag Autocorrelation D-W Statistic p-value**

**1 0.8219683 0.2701599 0**

**Alternative hypothesis: rho != 0**

**Analysis:**

Durbin Watson test is for checking the autocorrelation in the our data. If the D-W statistic value is 2 or in between 1.5 to 2.5, it is consider as normal.

If the value is 0 then it is a positive serial auto correlation and if the value is 4 then it is a negative serial autocorrelation .

In this test we got the value around 0.3, which is nearly equals to 0.

And the P-value is also 0, which means we are rejecting the null hypothesis and accepting the alternate hypothesis, where null hypothesis is ‘There is no autocorrelation’ and alternate hypothesis is ‘There is a autocorrelation’.

By considering these two values, we can say that there is autocorrelation in our data.

1. Note the original data appears to have a pronounced cyclical pattern. Assuming the complete cycles are four quarters long, construct a set of seasonal indices which describe the typical annual fluctuations in visitors. Use these indices to deseasonalize the visitors data. Store this deseasonalized data in a column in the original data frame.

**RCode:**

**#5**

**SI=data.frame(quater=1:4,average=0,index=0)**

**for(a in 1:4) {**

**count=0**

**for(b in 1:nrow(my\_data)) {**

**if(a==my\_data$quarter[b]) {**

**SI$average[a]=SI$average[a]+my\_data$china.visitors[b]**

**count=count+1 }**

**}**

**SI$average[a]=SI$average[a]/count**

**SI$index[a]=SI$average[a]/mean(my\_data$china.visitors)**

**}**

**#deseasonalize the visitors data**

**for (a in 1:4){**

**for(b in 1:nrow(my\_data)) {**

**if(a==my\_data$quarter[b]) {**

**my\_data$deseason[b]=my\_data$china.visitors[b]/SI$index[a]**

**} }**

**}**

**Analysis:**

For deseasoning, we need to calculate seasonal indices. Seasonal indices are the average value of number of visitors for the same quarter. And we get the index by the average value of the specific quarter divided by total mean of visitors.

After getting the seasonal indices, we divide the number of visitors by the quarter seasonal indices index to calculate Deseasoned values.

1. Using the deseasonalized data parameterize four different regression models. A simple regression model will be the base case to be followed by second order, third order, and fourth order polynomial models which attempt to describe the longer-term secular fluctuations in the deseasonalized data.

**Rcode:**

**#6**

**#1st order**

**output1=lm(my\_data$deseason~my\_data$index,data=my\_data)**

**summary(output1)**

**plot(my\_data$index,output1$residuals,type="o",pch=19,**

**lwd=3,main="Deseasonalized Residuals")**

**abline(0,0,col="red",lwd=3)**

**durbinWatsonTest(output1)**

**plot(my\_data$index,my\_data$deseason,type="o",**

**lwd=3,main="Deseasoned Data Plot")**

**points(my\_data$index,output1$fitted.values,**

**type="l",col="red",lwd=3)**

**P.T.O**

**Output in Console window:**

**> #6**

**> #1st order**

**> output1=lm(my\_data$deseason~my\_data$index,data=my\_data)**

**> summary(output1)**

**Call:**

**lm(formula = my\_data$deseason ~ my\_data$index, data = my\_data)**

**Residuals:**

**Min 1Q Median 3Q Max**

**-2339.6 -892.8 -595.9 457.4 4830.8**

**Coefficients:**

**Estimate Std. Error t value Pr(>|t|)**

**(Intercept) 3358.33 459.19 7.314 1.57e-09 \*\*\***

**my\_data$index 64.77 14.53 4.458 4.43e-05 \*\*\***

**---**

**Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1**

**Residual standard error: 1664 on 52 degrees of freedom**

**Multiple R-squared: 0.2765, Adjusted R-squared: 0.2626**

**F-statistic: 19.88 on 1 and 52 DF, p-value: 4.434e-05**

Both the Beta coefficient are significant. Intercept value is 3358.33 and slope is 64.77.

Multiple R-squared value is 27% and Adjusted R-Squared is 26%, which is not a good value and clearly from Adjusted R-squared value the data doesn’t follow first order regression.

**> plot(my\_data$index,output1$residuals,type="o",pch=19,**

**+ lwd=3,main="Deseasonalized Residuals")**

**> abline(0,0,col="red",lwd=3)**

Chart, line chart

Description automatically generated**Analysis:**

This is the Plot between index and linear model output residuals or error values.

The graph is widely spread because the peak point is at 5000 standard deviations, which is normally an outlier and the points below are also outliers.

As discussed previously, it follows secular trend and seasonal trend.

**> durbinWatsonTest(output1)**

**lag Autocorrelation D-W Statistic p-value**

**1 0.9039553 0.1323618 0**

**Alternative hypothesis: rho != 0**

The D-W statistic value is nearly 0, and the p-value is also 0, rejecting the null hypothesis and accepting the alternate hypothesis, telling us there is a positive serial auto correlation.

**> plot(my\_data$index,my\_data$deseason,type="o",**

**+ lwd=3,main="Deseasoned Data Plot")**

**> points(my\_data$index,output1$fitted.values,**

**+ type="l",col="red",lwd=3)**

Chart, line chart

Description automatically generated

**Analysis:**

This graph is plotted between the index and deseason data to check if our first order polynomial regression is a fit or not.

Clearly from the graph the data is not a best fit for first order polynomial regression. There is large peak and sudden drop and at the end there is a rise once again.

Uncertainty of simple linear regression because of cyclicality

**Rcode:**

**# Second order**

**my\_data$index2=my\_data$index^2**

**output2=lm(deseason~index+index2,data=my\_data)**

**plot(my\_data$index,output2$residuals,type="o",pch=19,**

**lwd=3,main="Deseasonalized Residuals")**

**abline(0,0,col="red",lwd=3)**

**durbinWatsonTest(output2)**

**plot(my\_data$index,my\_data$deseason,type="o",**

**lwd=3,main="Deseasoned Data Plot 2")**

**points(my\_data$index,output2$fitted.values,**

**type="l",col="red",lwd=3)**

**Output in Console Window:**

**> # Second order**

**> my\_data$index2=my\_data$index^2**

**> output2=lm(deseason~index+index2,data=my\_data)**

**> plot(my\_data$index,output2$residuals,type="o",pch=19,**

**+ lwd=3,main="Deseasonalized Residuals")**

**> abline(0,0,col="red",lwd=3)**

Chart, line chart

Description automatically generated

**Analysis:**

There is no much difference between the first and second order polynomial regression. The outlier standard deviation is decreased to 3000.

**> durbinWatsonTest(output2)**

**lag Autocorrelation D-W Statistic p-value**

**1 0.8972268 0.1399817 0**

**Alternative hypothesis: rho != 0**

There is no change in the D-W Statistic value. It is almost nearly equal to 0 and there is a positive serial autocorrelation.

**> plot(my\_data$index,my\_data$deseason,type="o",**

**+ lwd=3,main="Deseasoned Data Plot 2")**

**> points(my\_data$index,output2$fitted.values,**

**+ type="l",col="red",lwd=3)**

Chart, line chart, scatter chart

Description automatically generated

**Analysis:**

Second order polynomial is trying to fit the data, but still we can do better fit and second order is not a good fit. There are some many outliers to be covered.

**Rcode:**

**#3rd order**

**my\_data$index3=my\_data$index^3**

**output3=lm(deseason~index+index2+index3,data=my\_data)**

**plot(my\_data$index,output3$residuals,type="o",pch=19,**

**lwd=3,main="Deseasonalized Residuals")**

**abline(0,0,col="red",lwd=3)**

**durbinWatsonTest(output3)**

**plot(my\_data$index,my\_data$deseason,type="o",**

**lwd=3,main="Deseasoned Data Plot 3")**

**points(my\_data$index,output3$fitted.values,**

**type="l",col="red",lwd=3)**

**Output in Concole window:**

**> #3rd order**

**> my\_data$index3=my\_data$index^3**

**> output3=lm(deseason~index+index2+index3,data=my\_data)**

**> plot(my\_data$index,output3$residuals,type="o",pch=19,**

**+ lwd=3,main="Deseasonalized Residuals")**

**> abline(0,0,col="red",lwd=3)**

Chart, line chart

Description automatically generated

**> durbinWatsonTest(output3)**

**lag Autocorrelation D-W Statistic p-value**

**1 0.8482239 0.2783585 0**

**Alternative hypothesis: rho != 0**

**> plot(my\_data$index,my\_data$deseason,type="o",**

**+ lwd=3,main="Deseasoned Data Plot 3")**

**> points(my\_data$index,output3$fitted.values,**

**+ type="l",col="red",lwd=3)**

Chart, line chart

Description automatically generated

**Analysis:**

This graph shows there is a better fit for third order polynomial regression. The regression line is trying to cover all the peak points as well as the low points.

If we see the D-W statistics value there is a slight improvement in the value, which is around 0.29. But there is a positive serial autocorrelation.

**Rcode:**

**#4th order**

**my\_data$index4=my\_data$index^4**

**output4=lm(deseason~index+index2+index3+index4,data=my\_data)**

**plot(my\_data$index,ouput4$residuals,type="o",pch=19,**

**lwd=3,main="Deseasonalized Residuals")**

**abline(0,0,col="red",lwd=3)**

**durbinWatsonTest(output4)**

**plot(my\_data$index,my\_data$deseason,type="o",**

**lwd=3,main="Deseasoned Data Plot 4")**

**points(my\_data$index,output4$fitted.values,**

**type="l",col="red",lwd=3)**

**Output in Concole window:**

**> #4th order**

**> my\_data$index4=my\_data$index^4**

**> output4=lm(deseason~index+index2+index3+index4,data=my\_data)**

**> plot(my\_data$index,output4$residuals,type="o",pch=19,**

**+ lwd=3,main="Deseasonalized Residuals")**

**> abline(0,0,col="red",lwd=3)**

Chart, line chart

Description automatically generated

**> durbinWatsonTest(output4)**

**lag Autocorrelation D-W Statistic p-value**

**1 0.8318296 0.2925526 0**

**Alternative hypothesis: rho != 0**

**> plot(my\_data$index,my\_data$deseason,type="o",**

**+ lwd=3,main="Deseasoned Data Plot 4")**

**> points(my\_data$index,output4$fitted.values,**

**+ type="l",col="red",lwd=3)**

Chart, line chart

Description automatically generated

**Analysis:**

There 4th order polynomial regression is covering more than 3rd order polynomial regression line. The residual graph is also better than other polynomial regression.

The D-W statistics value is 0.29, there is no difference between 4th and 3rd order polynomial. There is still positive serial auto correlation.

1. Reseasonalize the fitted values for each of the four models, storing the reseasonalized values in separate columns in the original data frame. Drawing on Analysis Part 3 above, construct a plot showing the original data and the fitted values for each of the four regression models. Show the four sets of fitted values plots in contrasting colors and title the graph appropriately.

**Rcode:**

**#7**

**#Reseasonalize the values for 4 models**

**#1st model**

**for(j in 1:nrow(my\_data)) {**

**xx=my\_data$quarter[j]**

**my\_data$reseason.y.hat[j]=output1$fitted.values[j]\*SI$index[xx]**

**my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**}**

**plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 1")**

**points(my\_data$index,my\_data$china.visitors,pch=19,**

**type="o",lwd=2,col="red")**

**cor(my\_data$index,my\_data$reseason.y.hat)**

**cor(my\_data$index,my\_data$reseason.y.hat)^2**

**Ouput in Console window:**

**> #7**

**> #Reseasonalize the values for 4 models**

**> #1st model**

**> for(j in 1:nrow(my\_data)) {**

**+ xx=my\_data$quarter[j]**

**+ my\_data$reseason.y.hat[j]=output1$fitted.values[j]\*SI$index[xx]**

**+ my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**+ }**

**> plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 1")**

**> points(my\_data$index,my\_data$china.visitors,pch=19,**

**+ type="o",lwd=2,col="red")**

**> cor(my\_data$index,my\_data$reseason.y.hat)**

**[1] 0.9126722**

**> cor(my\_data$index,my\_data$reseason.y.hat)^2**

**[1] 0.8329705**

Chart, line chart

Description automatically generated

**Rcode:**

**#second order**

**for(j in 1:nrow(my\_data)) {**

**xx=my\_data$quarter[j]**

**my\_data$reseason.y.hat[j]=**

**output2$fitted.values[j]\*SI$index[xx]**

**my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**}**

**plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 2")**

**points(my\_data$index,my\_data$china.visitors,pch=19,**

**type="o",lwd=2,col="red")**

**cor(my\_data$index,my\_data$reseason.y.hat)**

**cor(my\_data$index,my\_data$reseason.y.hat)^2**

**Ouput in Console window:**

**> #second order**

**> for(j in 1:nrow(my\_data)) {**

**+ xx=my\_data$quarter[j]**

**+ my\_data$reseason.y.hat[j]=**

**+ output2$fitted.values[j]\*SI$index[xx]**

**+ my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**+ }**

**> plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 2")**

**> points(my\_data$index,my\_data$china.visitors,pch=19,**

**+ type="o",lwd=2,col="red")**

**> cor(my\_data$index,my\_data$reseason.y.hat)**

**[1] 0.8724063**

**> cor(my\_data$index,my\_data$reseason.y.hat)^2**

**[1] 0.7610927**

Chart, line chart

Description automatically generated

**Rcode:**

**#third order**

**for(j in 1:nrow(my\_data)) {**

**xx=my\_data$quarter[j]**

**my\_data$reseason.y.hat[j]=**

**output3$fitted.values[j]\*SI$index[xx]**

**my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**}**

**plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 3")**

**points(my\_data$index,my\_data$china.visitors,pch=19,**

**type="o",lwd=2,col="red")**

**cor(my\_data$index,my\_data$reseason.y.hat)**

**cor(my\_data$index,my\_data$reseason.y.hat)^2**

**Ouput in Console window:**

**> #third order**

**> for(j in 1:nrow(my\_data)) {**

**+ xx=my\_data$quarter[j]**

**+ my\_data$reseason.y.hat[j]=**

**+ output3$fitted.values[j]\*SI$index[xx]**

**+ my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**+ }**

**> plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 3")**

**> points(my\_data$index,my\_data$china.visitors,pch=19,**

**+ type="o",lwd=2,col="red")**

**> cor(my\_data$index,my\_data$reseason.y.hat)**

**[1] 0.6100224**

**> cor(my\_data$index,my\_data$reseason.y.hat)^2**

**[1] 0.3721273**

Chart, scatter chart

Description automatically generated

**Rcode:**

**#fourth order**

**for(j in 1:nrow(my\_data)) {**

**xx=my\_data$quarter[j]**

**my\_data$reseason.y.hat[j]=**

**output4$fitted.values[j]\*SI$index[xx]**

**my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**}**

**plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 4")**

**points(my\_data$index,my\_data$china.visitors,pch=19,**

**type="o",lwd=2,col="red")**

**cor(my\_data$index,my\_data$reseason.y.hat)**

**cor(my\_data$index,my\_data$reseason.y.hat)^2**

**Ouput in Console window:**

**> #fourth order**

**> for(j in 1:nrow(my\_data)) {**

**+ xx=my\_data$quarter[j]**

**+ my\_data$reseason.y.hat[j]=**

**+ output4$fitted.values[j]\*SI$index[xx]**

**+ my\_data$reseason.error[j]=my\_data$production[j]-my\_data$reseason.y.hat[j]**

**+ }**

**> plot(my\_data$index,my\_data$reseason.y.hat,pch=19,type="o",lwd=3, main="Actual v. Reseasoned Values for model 4")**

**> points(my\_data$index,my\_data$china.visitors,pch=19,**

**+ type="o",lwd=2,col="red")**

**> cor(my\_data$index,my\_data$reseason.y.hat)**

**[1] 0.608804**

**> cor(my\_data$item,my\_data$reseason.y.hat)^2**

**> cor(my\_data$index,my\_data$reseason.y.hat)^2**

**[1] 0.3706423**

Chart, scatter chart

Description automatically generated

**Analysis:**

**Simple linear regression doesn’t fit for this data. we have secular trend and seasonal trend**

**where secular trend is good for regression and seasonal is not good for regression**

**we have split the trends and done forecast for each and combine the forecast**

**into a single value.**

From the 1st order and 2nd order graph the forecast is not accurate and reseason is not a good fit.

There is a large deviation in the actual and forecast.

we have got much more accurate forecast for 3rd and 4th order polynomial and in the 4th order the forecast and reseason is matching more accurate. At the tail end the graph has cyclical trend and the 4th order forecast matches the peaks and low perfectly.

1. Select the model which in your view is the best fit to the deseasonalized data. Give a brief justification as to why you believe your selection is the best fit.

**Rcode:**

**#8th**

**summary(output3)**

**Ouput in Console window:**

**> #8th**

**> summary(output3)**

**Call:**

**lm(formula = deseason ~ index + index2 + index3, data = my\_data)**

**Residuals:**

**Min 1Q Median 3Q Max**

**-1477.9 -880.3 -128.4 389.7 3254.2**

**Coefficients:**

**Estimate Std. Error t value Pr(>|t|)**

**(Intercept) -996.37824 650.74976 -1.531 0.132**

**index 883.74782 101.52939 8.704 1.38e-11 \*\*\***

**index2 -34.34425 4.26835 -8.046 1.41e-10 \*\*\***

**index3 0.39543 0.05104 7.747 4.10e-10 \*\*\***

**---**

**Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1**

**Residual standard error: 1113 on 50 degrees of freedom**

**Multiple R-squared: 0.6884, Adjusted R-squared: 0.6697**

**F-statistic: 36.83 on 3 and 50 DF, p-value: 1.045e-12**

**Analysis:**

All the index values are significant. Intercept is not significant.

Residual Standard error or standard error is 1113, when compared to simple regression the graph is tighter, which is good.

Multiple R-Squared is 68% and Adjusted R-squared is 67%, which is a good number.

**Rcode:**

**summary(output4)**

**Ouput in Console window:**

**> summary(output4)**

**Call:**

**lm(formula = deseason ~ index + index2 + index3 + index4, data = my\_data)**

**Residuals:**

**Min 1Q Median 3Q Max**

**-1679.9 -825.7 -104.5 338.1 3177.8**

**Coefficients:**

**Estimate Std. Error t value Pr(>|t|)**

**(Intercept) -1.563e+03 8.496e+02 -1.839 0.07192 .**

**index 1.074e+03 2.099e+02 5.118 5.16e-06 \*\*\***

**index2 -4.958e+01 1.532e+01 -3.237 0.00217 \*\***

**index3 8.237e-01 4.166e-01 1.977 0.05366 .**

**index4 -3.893e-03 3.759e-03 -1.036 0.30538**

**---**

**Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1**

**Residual standard error: 1113 on 49 degrees of freedom**

**Multiple R-squared: 0.6951, Adjusted R-squared: 0.6702**

**F-statistic: 27.93 on 4 and 49 DF, p-value: 4.142e-12**

**Analysis:**

Index4 value is not significant. Index and index2 are highly significant.

Intercept and index3 value less significant. Their p-values are just around the 5%.

Residual Standard error or standard error is 1113, when compared to simple regression the graph is tighter, which is good.

Multiple R-Squared is 70% and Adjusted R-squared is 67%, which is a good number.

There is no much difference between the Adjusted R-squared values for 3rd and 4th order. But we go back and see the reseason graphs 4th order forecast is most accurate overall. So, that we can say that 4th order is the best fit for this model.