Shetty\_S\_Assignment\_10

Sooraj Shetty

April 14, 2017

library(RCurl)

## Loading required package: bitops

library(plyr)

## Warning: package 'plyr' was built under R version 3.3.3

library(forecast)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: timeDate

## This is forecast 7.3

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.3

source(url("http://lib.stat.cmu.edu/general/tsa2/Rcode/itall.R"))

## itall has been installed

# Assignment

*NOTE: It's your choice to submit EITHER Lesson 02 (ARIMA models) OR Lesson 03 (Trend Analysis) for the module Time Series Analysis.*

Answer the following questions:  
\* Can some form of Time series analysis help in your research project to forcast a prediction? \*\*Arima Model can be used in our research project to forecast prediction

\* If it can apply Time series analysis to your research project? Does it help?  
\*\* Yes it does  
  
\* If (and only if) you can't use some form of Time series analysis help in your research project then apply a form of Time series analysis to a data set you find at the [UC Irvine Machine Learning Repository](http://archive.ics.uci.edu/ml/)  
\*\* Using our project

## I will be using my research project for the time series analysis using ARIMA model to forecast a prediction, It will be helpful as a base for creating the prediction model of the project

mymodel <- read.csv("train.csv",header = TRUE)

## Converting the data columns into factor

# Wrangling the data for use  
  
mymodel$Dates <- as.POSIXct(mymodel$Dates, format = "%Y-%m-%d %H:%M:%S")  
mymodel$Date <- (format(mymodel$Dates, "%d"))  
mymodel$Year <- format(mymodel$Dates, "%Y")  
mymodel$months <- (format(mymodel$Dates, "%m"))  
mymodel$Hours <- (format(mymodel$Dates, "%H"))  
  
  
mymodel$Category <- as.factor(mymodel$Category)  
mymodel$DayOfWeek <- as.factor(mymodel$DayOfWeek)  
mymodel$PdDistrict <- as.factor(mymodel$PdDistrict)  
mymodel$Resolution <- as.factor(mymodel$Resolution)  
mymodel$Address <- as.factor(mymodel$Address)  
  
str(mymodel)

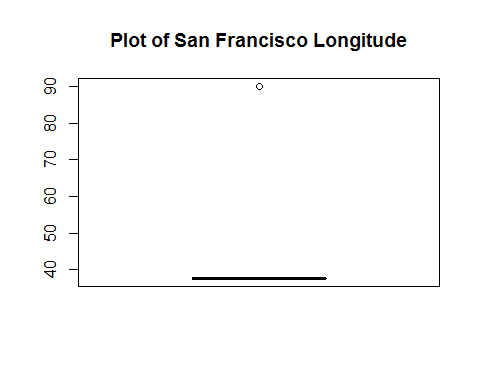
## 'data.frame': 878049 obs. of 13 variables:  
## $ Dates : POSIXct, format: "2015-05-13 23:53:00" "2015-05-13 23:53:00" ...  
## $ Category : Factor w/ 39 levels "ARSON","ASSAULT",..: 38 22 22 17 17 17 37 37 17 17 ...  
## $ Descript : Factor w/ 879 levels "ABANDONMENT OF CHILD",..: 867 811 811 405 405 407 740 740 405 405 ...  
## $ DayOfWeek : Factor w/ 7 levels "Friday","Monday",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ PdDistrict: Factor w/ 10 levels "BAYVIEW","CENTRAL",..: 5 5 5 5 6 3 3 1 7 2 ...  
## $ Resolution: Factor w/ 17 levels "ARREST, BOOKED",..: 1 1 1 12 12 12 12 12 12 12 ...  
## $ Address : Factor w/ 23228 levels "0 Block of HARRISON ST",..: 19791 19791 22698 4267 1844 1506 13323 18055 11385 17659 ...  
## $ X : num -122 -122 -122 -122 -122 ...  
## $ Y : num 37.8 37.8 37.8 37.8 37.8 ...  
## $ Date : chr "13" "13" "13" "13" ...  
## $ Year : chr "2015" "2015" "2015" "2015" ...  
## $ months : chr "05" "05" "05" "05" ...  
## $ Hours : chr "23" "23" "23" "23" ...

## converting the columns to numeric for the year, date, months and hours

mymodel$Year <- as.numeric(mymodel$Year)  
mymodel$Date <- as.numeric(mymodel$Date)  
mymodel$months <- as.numeric(mymodel$months)  
mymodel$Hours <- as.numeric(mymodel$Hours)

# Removing the anomalies/outliers?

boxplot(mymodel$Y, main = "Plot of San Francisco Longitude")



nrow(mymodel)

## [1] 878049

long <- which(mymodel$Y == 90, arr.ind = T)  
length(long)

## [1] 67

mymodel <- mymodel[-long,]  
nrow(mymodel)

## [1] 877982

#removing 47 rows where date is NA  
  
mymodel <- mymodel[-which(is.na(mymodel$Dates)),]  
nrow(mymodel)

## [1] 877935

head(mymodel)

## Dates Category Descript  
## 1 2015-05-13 23:53:00 WARRANTS WARRANT ARREST  
## 2 2015-05-13 23:53:00 OTHER OFFENSES TRAFFIC VIOLATION ARREST  
## 3 2015-05-13 23:33:00 OTHER OFFENSES TRAFFIC VIOLATION ARREST  
## 4 2015-05-13 23:30:00 LARCENY/THEFT GRAND THEFT FROM LOCKED AUTO  
## 5 2015-05-13 23:30:00 LARCENY/THEFT GRAND THEFT FROM LOCKED AUTO  
## 6 2015-05-13 23:30:00 LARCENY/THEFT GRAND THEFT FROM UNLOCKED AUTO  
## DayOfWeek PdDistrict Resolution Address X  
## 1 Wednesday NORTHERN ARREST, BOOKED OAK ST / LAGUNA ST -122.4259  
## 2 Wednesday NORTHERN ARREST, BOOKED OAK ST / LAGUNA ST -122.4259  
## 3 Wednesday NORTHERN ARREST, BOOKED VANNESS AV / GREENWICH ST -122.4244  
## 4 Wednesday NORTHERN NONE 1500 Block of LOMBARD ST -122.4270  
## 5 Wednesday PARK NONE 100 Block of BRODERICK ST -122.4387  
## 6 Wednesday INGLESIDE NONE 0 Block of TEDDY AV -122.4033  
## Y Date Year months Hours  
## 1 37.77460 13 2015 5 23  
## 2 37.77460 13 2015 5 23  
## 3 37.80041 13 2015 5 23  
## 4 37.80087 13 2015 5 23  
## 5 37.77154 13 2015 5 23  
## 6 37.71343 13 2015 5 23

count(mymodel$Year)

## x freq  
## 1 2003 73881  
## 2 2004 73405  
## 3 2005 70741  
## 4 2006 69909  
## 5 2007 68008  
## 6 2008 70168  
## 7 2009 68992  
## 8 2010 66539  
## 9 2011 66616  
## 10 2012 71731  
## 11 2013 75604  
## 12 2014 74760  
## 13 2015 27581

## Adding a column as CategoryMap which will have factor values for category

mymodel$CategoryMap <- mymodel$Category  
  
levels(mymodel$CategoryMap) <- gsub("ARSON", 1, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("ASSAULT", 2, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("BAD CHECKS", 3, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("BRIBERY", 4, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("BURGLARY", 5, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("DISORDERLY CONDUCT", 6, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("DRIVING UNDER THE INFLUENCE", 7, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("DRUG/NARCOTIC", 8, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("DRUNKENNESS", 9, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("EMBEZZLEMENT", 10, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("EXTORTION", 11,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("FAMILY OFFENSES", 12,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("FRAUD", 13, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("FORGERY/COUNTERFEITING", 14,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("GAMBLING", 15, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("KIDNAPPING", 16, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("LARCENY/THEFT", 17, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("LIQUOR LAWS", 18,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("LOITERING", 19, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("MISSING PERSON", 20, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("NON-CRIMINAL", 21, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("OTHER OFFENSES", 22, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("PORNOGRAPHY/OBSCENE MAT", 23, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("PROSTITUTION", 24, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("RECOVERED VEHICLE", 25, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("ROBBERY", 26, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("RUNAWAY", 27, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("SECONDARY CODES", 28, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("SEX OFFENSES FORCIBLE", 29, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("SEX OFFENSES NON FORCIBLE", 30,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("STOLEN PROPERTY", 31,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("SUICIDE", 32, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("SUSPICIOUS OCC", 33, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("TREA", 34,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("TRESPASS", 35,levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("VANDALISM", 36, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("VEHICLE THEFT", 37, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("WARRANTS", 38, levels(mymodel$CategoryMap))  
levels(mymodel$CategoryMap) <- gsub("WEAPON LAWS", 39, levels(mymodel$CategoryMap))

# Seasonal Models

When there are patterns that repeat over known, fixed periods of time (i.e. day, week, month, quarter, year, etc.) within the data set it is considered to be seasonal variation. One has a model for the periodic fluctuations based on knowledge of the domain.

The seasonal ARIMA model incorporates both non-seasonal and seasonal factors in a multiplicative model. In a seasonal ARIMA model, seasonal parameters predict xt using data values and errors at times with lags that are multiples of S (the span of the seasonality). Before we model for a given data set, one must have an initial guess about the data generation process, that is the span of the seasonality (i.e.day, week, month, quarter, year, etc.)

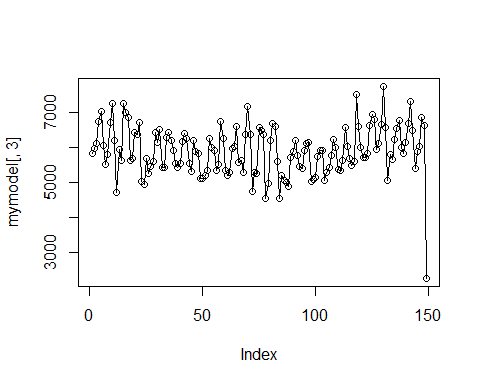
# Seasonal Models in R

Here we are using an ARIMA model to identify seasonality trends by looking for signficant seasonal differences.

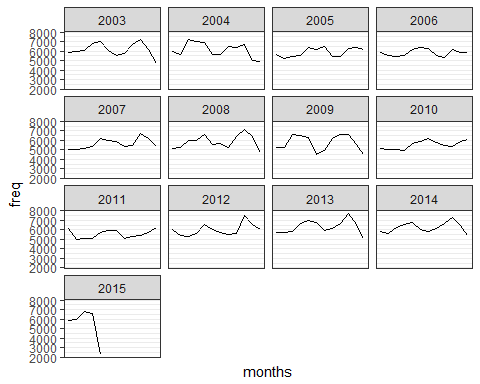
library(ggplot2)  
head(mymodel)

## Dates Category Descript  
## 1 2015-05-13 23:53:00 WARRANTS WARRANT ARREST  
## 2 2015-05-13 23:53:00 OTHER OFFENSES TRAFFIC VIOLATION ARREST  
## 3 2015-05-13 23:33:00 OTHER OFFENSES TRAFFIC VIOLATION ARREST  
## 4 2015-05-13 23:30:00 LARCENY/THEFT GRAND THEFT FROM LOCKED AUTO  
## 5 2015-05-13 23:30:00 LARCENY/THEFT GRAND THEFT FROM LOCKED AUTO  
## 6 2015-05-13 23:30:00 LARCENY/THEFT GRAND THEFT FROM UNLOCKED AUTO  
## DayOfWeek PdDistrict Resolution Address X  
## 1 Wednesday NORTHERN ARREST, BOOKED OAK ST / LAGUNA ST -122.4259  
## 2 Wednesday NORTHERN ARREST, BOOKED OAK ST / LAGUNA ST -122.4259  
## 3 Wednesday NORTHERN ARREST, BOOKED VANNESS AV / GREENWICH ST -122.4244  
## 4 Wednesday NORTHERN NONE 1500 Block of LOMBARD ST -122.4270  
## 5 Wednesday PARK NONE 100 Block of BRODERICK ST -122.4387  
## 6 Wednesday INGLESIDE NONE 0 Block of TEDDY AV -122.4033  
## Y Date Year months Hours CategoryMap  
## 1 37.77460 13 2015 5 23 38  
## 2 37.77460 13 2015 5 23 22  
## 3 37.80041 13 2015 5 23 22  
## 4 37.80087 13 2015 5 23 17  
## 5 37.77154 13 2015 5 23 17  
## 6 37.71343 13 2015 5 23 17

mymodel <- count(mymodel, c("Year","months"))  
plot(mymodel[,3], type='o') # it is difficult to identify seasonality trends here. So,aggregate the data by month to better understand this trend



ggplot(mymodel, aes(x=months, y= freq))+  
 stat\_summary(geom = 'line', fun.y='mean')+ # take the mean of each month  
 scale\_x\_discrete(breaks=seq(1,12,1), labels=seq(1,12,1))+  
 theme\_bw()+ # add a little style  
 facet\_wrap(~Year) # visualize year by year



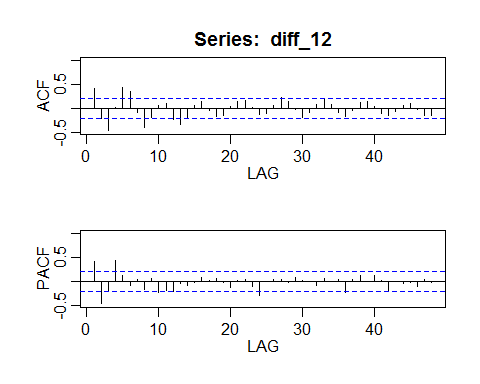
#Since we hypothesize that there is seasonality,  
#we can take the seasonal difference (create a variable that gives the 12TH differences), then look at the ACF and PACF.  
  
mydata<-ts(mymodel[1:100,][,3])  
mydata

## Time Series:  
## Start = 1   
## End = 100   
## Frequency = 1   
## [1] 5831 5963 6099 6750 7024 6045 5502 5798 6703 7259 6194 4713 5937 5626  
## [15] 7259 6985 6862 5611 5679 6435 6361 6695 5011 4944 5668 5251 5448 5585  
## [29] 6426 6134 6511 5421 5423 6276 6418 6180 5896 5537 5418 5524 6177 6393  
## [43] 6246 5523 5312 6183 5868 5832 5094 5093 5202 5336 6253 5984 5894 5331  
## [57] 5509 6733 6253 5326 5182 5284 5968 6028 6597 5556 5631 5275 6367 7173  
## [71] 6371 4736 5272 5237 6572 6472 6355 4543 4960 6199 6671 6593 5581 4537  
## [85] 5179 5063 4994 4890 5708 5888 6207 5758 5453 5395 5906 6098 6130 5029  
## [99] 5068 5123

diff\_12 <- diff(mydata, 12)  
diff\_12

## Time Series:  
## Start = 13   
## End = 100   
## Frequency = 1   
## [1] 106 -337 1160 235 -162 -434 177 637 -342 -564 -1183  
## [12] 231 -269 -375 -1811 -1400 -436 523 832 -1014 -938 -419  
## [23] 1407 1236 228 286 -30 -61 -249 259 -265 102 -111  
## [34] -93 -550 -348 -802 -444 -216 -188 76 -409 -352 -192  
## [45] 197 550 385 -506 88 191 766 692 344 -428 -263  
## [56] -56 858 440 118 -590 90 -47 604 444 -242 -1013  
## [67] -671 924 304 -580 -790 -199 -93 -174 -1578 -1582 -647  
## [78] 1345 1247 -441 -1218 -1198 325 1561 951 -34 74 233

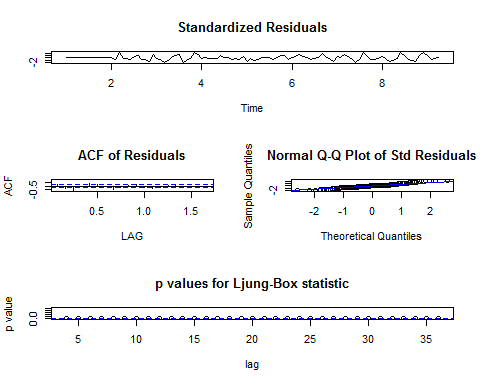
acf2(diff\_12, 48)



## ACF PACF  
## [1,] 0.41 0.41  
## [2,] -0.22 -0.47  
## [3,] -0.46 -0.20  
## [4,] 0.03 0.43  
## [5,] 0.44 0.12  
## [6,] 0.35 -0.08  
## [7,] -0.08 0.03  
## [8,] -0.40 -0.17  
## [9,] -0.18 0.07  
## [10,] 0.06 -0.24  
## [11,] 0.10 -0.19  
## [12,] -0.22 -0.22  
## [13,] -0.33 -0.04  
## [14,] -0.20 -0.09  
## [15,] 0.07 -0.02  
## [16,] 0.14 0.09  
## [17,] -0.05 0.01  
## [18,] -0.18 0.05  
## [19,] -0.14 -0.02  
## [20,] 0.05 -0.13  
## [21,] 0.15 0.02  
## [22,] 0.16 0.03  
## [23,] 0.03 -0.10  
## [24,] -0.13 -0.29  
## [25,] -0.10 0.00  
## [26,] 0.06 0.03  
## [27,] 0.23 0.03  
## [28,] 0.14 -0.02  
## [29,] -0.03 0.07  
## [30,] -0.18 0.01  
## [31,] -0.09 -0.01  
## [32,] 0.08 -0.10  
## [33,] 0.19 0.05  
## [34,] 0.08 -0.01  
## [35,] -0.08 0.03  
## [36,] -0.17 -0.22  
## [37,] -0.05 0.03  
## [38,] 0.13 0.12  
## [39,] 0.14 0.00  
## [40,] 0.03 0.11  
## [41,] -0.11 0.01  
## [42,] -0.15 -0.22  
## [43,] -0.07 -0.01  
## [44,] 0.06 -0.05  
## [45,] 0.11 -0.02  
## [46,] -0.02 -0.12  
## [47,] -0.15 0.03  
## [48,] -0.15 -0.03

#we see that for both the ACF and PACF we have significant autocorrelation at seasonal (12, 24, 36) lags. The ACF has a cluster around 12,   
#and not much else besides a tapering pattern throughout. Further, the PACF also has spikes on two multiples of S, AR(2)  
# Try, ARIMA (1,0, 0) x (2, 1, 0)12  
mydata<-ts(mydata, freq=12)  
mod1<-sarima(mydata, 1,0,0,2,1,0,12)

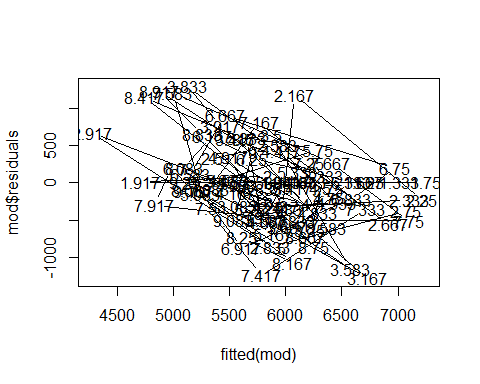
## initial value 6.442927   
## iter 2 value 6.277628  
## iter 3 value 6.273156  
## iter 4 value 6.272725  
## iter 5 value 6.272723  
## iter 6 value 6.272723  
## iter 6 value 6.272723  
## iter 6 value 6.272723  
## final value 6.272723   
## converged  
## initial value 6.373315   
## iter 2 value 6.369545  
## iter 3 value 6.369096  
## iter 4 value 6.369009  
## iter 5 value 6.368994  
## iter 6 value 6.368993  
## iter 6 value 6.368993  
## iter 6 value 6.368993  
## final value 6.368993   
## converged



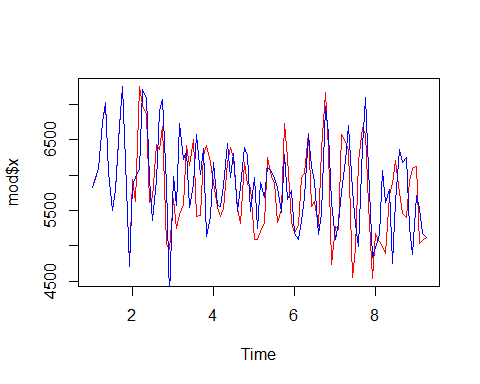
# install.packages("forecast")  
# Fit the Model  
mod<-Arima(mydata,order=c(1, 0, 0),  
 seasonal=list(order=c(2, 1, 0), period=12))  
mod

## Series: mydata   
## ARIMA(1,0,0)(2,1,0)[12]   
##   
## Coefficients:  
## ar1 sar1 sar2  
## 0.4213 -0.2858 -0.3744  
## s.e. 0.0958 0.1194 0.1406  
##   
## sigma^2 estimated as 339116: log likelihood=-685.82  
## AIC=1379.63 AICc=1380.11 BIC=1389.54

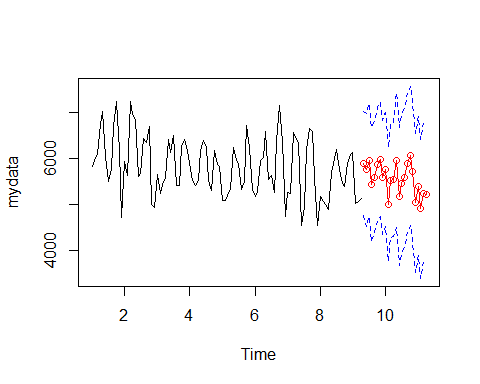
plot(fitted(mod), mod$residuals)



plot(mod$x, col='red')  
lines(fitted(mod), col='blue')



# Now, we have a reasonable prediction, we can forecast the model, say 24 months into the future.  
sarima.for(mydata, 24, 1,0,0,2,1,0,12)



## $pred  
## Jan Feb Mar Apr May Jun Jul Aug  
## 9 5897.148 5756.684 5973.110 5432.301  
## 10 5765.488 4993.503 5529.664 5537.997 5972.551 5177.381 5461.873 5585.953  
## 11 5405.542 4905.619 5250.381 5212.845   
## Sep Oct Nov Dec  
## 9 5596.200 5865.561 5994.268 5588.680  
## 10 5901.027 6063.650 5733.392 5042.389  
## 11   
##   
## $se  
## Jan Feb Mar Apr May Jun Jul Aug  
## 9 568.8625 613.7687 620.8282 621.9790  
## 10 622.2047 622.2047 622.2048 622.2048 737.8071 755.0902 757.8889 758.3473  
## 11 758.4372 758.4372 758.4372 758.4372   
## Sep Oct Nov Dec  
## 9 622.1677 622.1987 622.2038 622.2046  
## 10 758.4224 758.4348 758.4368 758.4371  
## 11

predict(mod, n.ahead=24)

## $pred  
## Jan Feb Mar Apr May Jun Jul Aug  
## 9 5961.243 5873.560 6097.938 5536.388  
## 10 5893.035 5103.857 5637.696 5648.759 6131.128 5374.081 5662.193 5764.840  
## 11 5604.671 5095.196 5447.188 5411.272   
## Sep Oct Nov Dec  
## 9 5686.537 5954.222 6108.799 5726.376  
## 10 6075.853 6242.978 5929.157 5248.093  
## 11   
##   
## $se  
## Jan Feb Mar Apr May Jun Jul Aug  
## 9 582.3368 631.9049 640.3017 641.7805  
## 10 642.0991 642.0991 642.0991 642.0991 765.0485 784.8601 788.3243 788.9375  
## 11 789.0698 789.0698 789.0698 789.0698   
## Sep Oct Nov Dec  
## 9 642.0426 642.0891 642.0974 642.0988  
## 10 789.0463 789.0656 789.0691 789.0697  
## 11