ORIE5550 - Homework 2

Luis Alonso Cendra Villalobos (lc2234)

2024-02-09

# Problem 2.

Consider the time series deaths in the R package itsmr with the monthly accidental death numbers in the US, 1973-1978. Do the following for this time series:

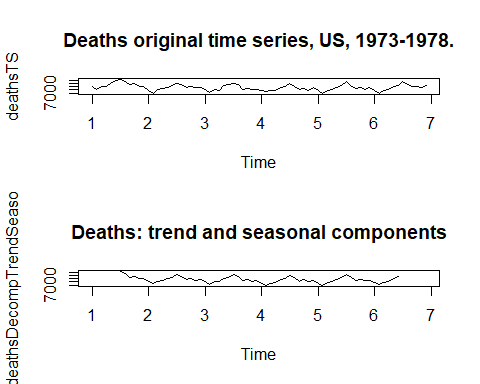
## 2.a.

Use the R function decompose to obtain trend and seasonal components of the time series and plot the results; Produce a time plot depicting the time series and the sum of the trend and seasonal components; Produce a time plot and a correlogram of the residuals obtained after removing the trend and seasonal components from the series; Include the R code;

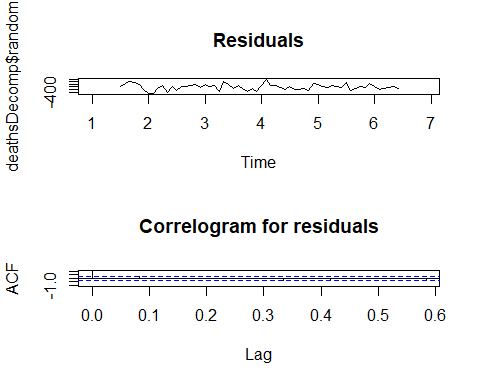
#install.packages("itsmr")  
  
library(itsmr)  
deaths

## [1] 9007 8106 8928 9137 10017 10826 11317 10744 9713 9938 9161 8927  
## [13] 7750 6981 8038 8422 8714 9512 10120 9823 8743 9129 8710 8680  
## [25] 8162 7306 8124 7870 9387 9556 10093 9620 8285 8433 8160 8034  
## [37] 7717 7461 7776 7925 8634 8945 10078 9179 8037 8488 7874 8647  
## [49] 7792 6957 7726 8106 8890 9299 10625 9302 8314 8850 8265 8796  
## [61] 7836 6892 7791 8129 9115 9434 10484 9827 9110 9070 8633 9240

deathsTS = ts(deaths, frequency=12)  
  
deathsDecomp <- decompose(deathsTS, "additive")  
  
deathsDecompTrendSeasonal <- deathsDecomp$trend + deathsDecomp$seasonal  
  
par(mfrow=c(2,1))  
plot.ts(deathsTS, main = "Deaths original time series, US, 1973-1978.")  
plot.ts(deathsDecompTrendSeasonal, main = "Deaths: trend and seasonal components")



par(mfrow=c(2,1))  
plot.ts(deathsDecomp$random, main = "Residuals")  
acf(deathsDecomp$random, lag.max = 7,   
 ylim = c(-1,1), main = "Correlogram for residuals",  
 na.action = na.pass)



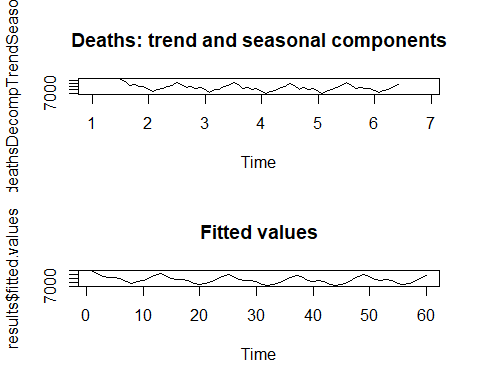
## 2.b.

Fit through least squares R function lm the quadratic trend and seasonal component (see HW paper for the equation) to the series and produce the fit results; Produce a time plot depicting the time series and the sum of the fitted trend and seasonal components; Produce a time plot and a correlogram of the residuals obtained after removing the trend and seasonal components from the series; Include the R code.

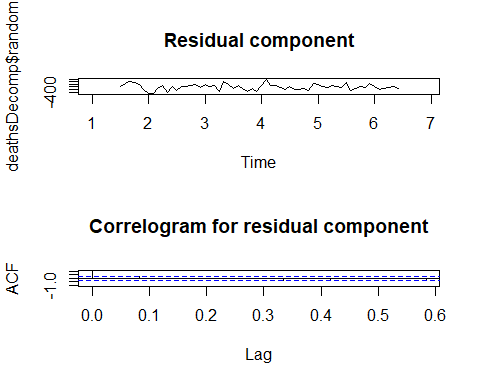
tt <- seq(1,72)  
y <- deathsDecompTrendSeasonal  
  
#fit regression model  
results <- lm( y ~ poly(tt, 2) +  
 cos(2\*tt\*pi/12) + sin(2\*tt\*pi/12) +  
 cos(2\*tt\*pi/6) + sin(2\*tt\*pi/6) +  
 cos(2\*tt\*pi/4) + sin(2\*tt\*pi/4), data = deathsDecompTrendSeasonal)  
# poly trend and sine-cos is seasonal variance, 12 yerly, 6 semester, 4 quarter  
  
#view model summary  
summary(results)

##   
## Call:  
## lm(formula = y ~ poly(tt, 2) + cos(2 \* tt \* pi/12) + sin(2 \*   
## tt \* pi/12) + cos(2 \* tt \* pi/6) + sin(2 \* tt \* pi/6) + cos(2 \*   
## tt \* pi/4) + sin(2 \* tt \* pi/4), data = deathsDecompTrendSeasonal)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -401.08 -209.45 21.03 183.86 461.17   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8781.49 37.13 236.495 < 2e-16 \*\*\*  
## poly(tt, 2)1 -1950.14 345.25 -5.648 7.25e-07 \*\*\*  
## poly(tt, 2)2 2476.48 407.40 6.079 1.54e-07 \*\*\*  
## cos(2 \* tt \* pi/12) -716.12 47.18 -15.179 < 2e-16 \*\*\*  
## sin(2 \* tt \* pi/12) -754.52 47.67 -15.827 < 2e-16 \*\*\*  
## cos(2 \* tt \* pi/6) 398.97 47.15 8.462 2.77e-11 \*\*\*  
## sin(2 \* tt \* pi/6) 82.16 47.23 1.740 0.08795 .   
## cos(2 \* tt \* pi/4) 157.13 47.15 3.333 0.00161 \*\*   
## sin(2 \* tt \* pi/4) -213.82 47.15 -4.535 3.52e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 258 on 51 degrees of freedom  
## (12 observations deleted due to missingness)  
## Multiple R-squared: 0.9317, Adjusted R-squared: 0.921   
## F-statistic: 87 on 8 and 51 DF, p-value: < 2.2e-16

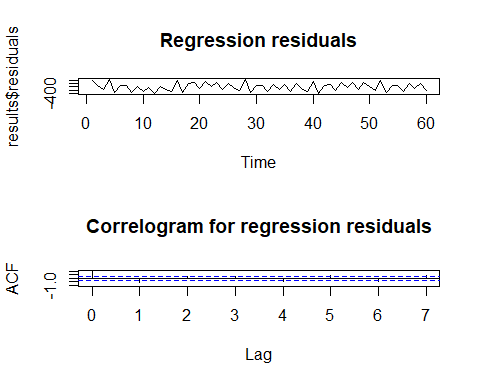
par(mfrow=c(2,1))  
plot.ts(deathsDecompTrendSeasonal, main = "Deaths: trend and seasonal components")  
plot.ts(results$fitted.values, main = "Fitted values")



par(mfrow=c(2,1))  
plot.ts(deathsDecomp$random, main = "Residual component")  
acf(deathsDecomp$random, lag.max = 7,   
 ylim = c(-1,1), main = "Correlogram for residual component",  
 na.action = na.pass)



par(mfrow=c(2,1))  
plot.ts(results$residuals, main = "Regression residuals")  
acf(results$residuals, lag.max = 7,   
 ylim = c(-1,1), main = "Correlogram for regression residuals",  
 na.action = na.pass)



# Problem 4.

Thoughsample ACFwaspresented in class as a tool for stationary models, note that it could also be computed for any time series. In fact, if time series have trends or/and seasonal components, this could also be seen from their correlograms.

## 4.a.

Read the time series data prices from the R package fpp3 and produce a time plot of the series prices$wheat; As you will see, the series clearly has a decreasing trend. Now, produce and include a correlogram of this series up to lag 30; Explain the pattern that you see; Supplement your explanation by drawing a few scatter plots of xt versus xt−h, the correlation between which yields the sample ACF at lag h.

#install.packages("fpp3")  
library("fpp3")

## Warning: package 'fpp3' was built under R version 4.3.2

## ── Attaching packages ────────────────────────────────────────────── fpp3 0.5 ──

## ✔ tibble 3.2.1 ✔ tsibble 1.1.4  
## ✔ dplyr 1.1.4 ✔ tsibbledata 0.4.1  
## ✔ tidyr 1.3.1 ✔ feasts 0.3.1  
## ✔ lubridate 1.9.3 ✔ fable 0.3.3  
## ✔ ggplot2 3.4.4 ✔ fabletools 0.3.4

## Warning: package 'tibble' was built under R version 4.3.2

## Warning: package 'dplyr' was built under R version 4.3.2

## Warning: package 'tidyr' was built under R version 4.3.2

## Warning: package 'lubridate' was built under R version 4.3.2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Warning: package 'tsibble' was built under R version 4.3.2

## Warning: package 'tsibbledata' was built under R version 4.3.2

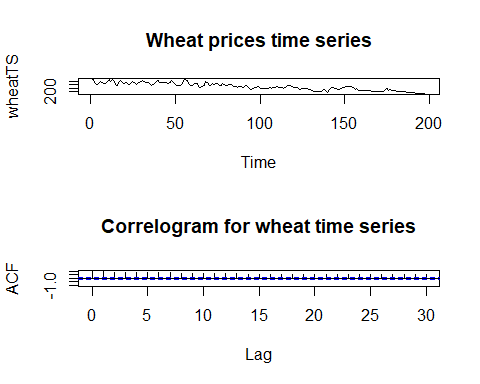
## Warning: package 'feasts' was built under R version 4.3.2

## Warning: package 'fabletools' was built under R version 4.3.2

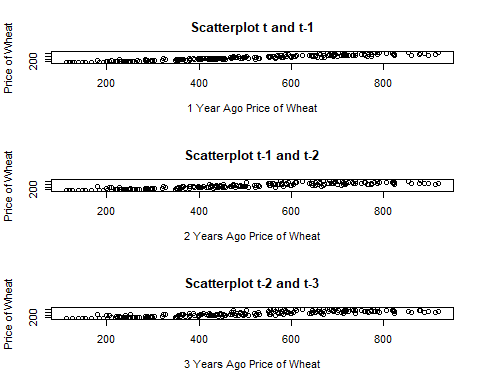
## Warning: package 'fable' was built under R version 4.3.2

## ── Conflicts ───────────────────────────────────────────────── fpp3\_conflicts ──  
## ✖ lubridate::date() masks base::date()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ fabletools::forecast() masks itsmr::forecast()  
## ✖ tsibble::intersect() masks base::intersect()  
## ✖ tsibble::interval() masks lubridate::interval()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ tsibble::setdiff() masks base::setdiff()  
## ✖ tsibble::union() masks base::union()

wheatTS = ts(prices$wheat, frequency=1)  
  
  
tt <- length(wheatTS)  
  
par(mfrow=c(2,1))  
plot.ts(wheatTS, main = "Wheat prices time series")  
acf(wheatTS, lag.max = 30,   
 ylim = c(-1,1), main = "Correlogram for wheat time series",  
 na.action = na.pass)



par(mfrow=c(3,1))  
plot(y=wheatTS[2:tt], x=wheatTS[1:(tt-1)], ylab = 'Price of Wheat', xlab = '1 Year Ago Price of Wheat', main = "Scatterplot t and t-1")  
plot(y=wheatTS[3:tt], x=wheatTS[1:(tt-2)], ylab = 'Price of Wheat', xlab = '2 Years Ago Price of Wheat', main = "Scatterplot t-1 and t-2")  
plot(y=wheatTS[4:tt], x=wheatTS[1:(tt-3)], ylab = 'Price of Wheat', xlab = '3 Years Ago Price of Wheat', main = "Scatterplot t-2 and t-3")



### Explanation:

From the Autocorrelation Functions (ACF) we can observe that lags, even up to 30, are significant and decay very slowly. This indicates the wheat prices today are auto correlated with prices up to 30 years prior. This could be because the series is not stationary (although we cannot state it from the ACFs alone), which is exhibited by the decreasing trend in the plot.Some reasons may be an overall increase in wheat producers globally (Argentina, China, etc.), the entry of wheat substitutes to the market, changes in global consumer preferences (as living standards rise, wheat consumption may not), etc. Additionally, from the scatter plots we can see that prices are likely correlated across years.

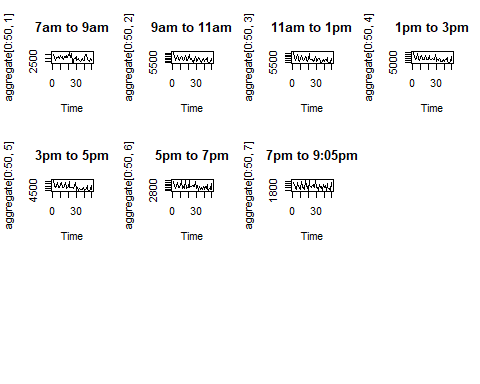
## 4.b.

Read the time series data bank calls from the R package fpp3. The data is a five-minute call volume handled on weekdays between 7:00am and 9:05pm, beginning 3 March 2003 for 164 days; Aggregate the volume data into 2-hour blocks, 7am-9am, 9am-11am, …, with the every last block 7pm-9:05pm of each day being 5 minutes longer; Produce the time plot of the aggregated time series for the first 50 days; As you will see, the series clearly exhibits seasonal variations. Now, produce and include a correlogram of this series up to lag 40; Explain the pattern that you see; Supplement your explanation by drawing a few scatter plots of xt versus xt−h as the problem above.

df <- as.data.frame(bank\_calls) # transform to dataframe  
  
#remove UTC tag  
date\_as\_posix <- strptime(df$DateTime, format="%Y-%m-%d %H:%M:%S", tz="UTC")   
invisible(strftime(date\_as\_posix, format="%Y-%m-%d %H:%M", tz="UTC"))  
  
# matrix with 164 rows (days) and 7 cols (intervals)  
aggregate <- matrix(0, nrow = 164, ncol = 7)  
  
# start on day 1  
day\_count = 1  
  
#start looping through the matrix starting in day 1 until day 164   
for (i in seq\_along(df$DateTime)) {  
   
 # if day changes, sum one to the index pointer  
 if (date(df$DateTime[i]) != date(df$DateTime[i-1]) && i > 1) {  
 day\_count = day\_count + 1  
 }  
   
 # Check under which interval the timestamp falls into and sum the calls  
 if (hour(df$DateTime[i]) >= 7 && hour(df$DateTime[i]) < 9 ) {  
 aggregate[day\_count,1] = aggregate[day\_count,1] + df$Calls[i]  
 }  
 if (hour(df$DateTime[i]) >= 9 && hour(df$DateTime[i]) < 11 ) {  
 aggregate[day\_count,2] = aggregate[day\_count,2] + df$Calls[i]  
 }  
 if (hour(df$DateTime[i]) >= 11 && hour(df$DateTime[i]) < 13 ) {  
 aggregate[day\_count,3] = aggregate[day\_count,3] + df$Calls[i]  
 }  
 if (hour(df$DateTime[i]) >= 13 && hour(df$DateTime[i]) < 15 ) {  
 aggregate[day\_count,4] = aggregate[day\_count,4] + df$Calls[i]  
 }  
 if (hour(df$DateTime[i]) >= 15 && hour(df$DateTime[i]) < 17 ) {  
 aggregate[day\_count,5] = aggregate[day\_count,5] + df$Calls[i]  
 }  
 if (hour(df$DateTime[i]) >= 17 && hour(df$DateTime[i]) < 19 ) {  
 aggregate[day\_count,6] = aggregate[day\_count,6] + df$Calls[i]  
 }  
 if (hour(df$DateTime[i]) >= 19 && hour(df$DateTime[i]) <= 21 ) {  
 aggregate[day\_count,7] = aggregate[day\_count,7] + df$Calls[i]  
 }  
}  
  
print("First 50 days of each interval are plotted below.")

## [1] "First 50 days of each interval are plotted below."

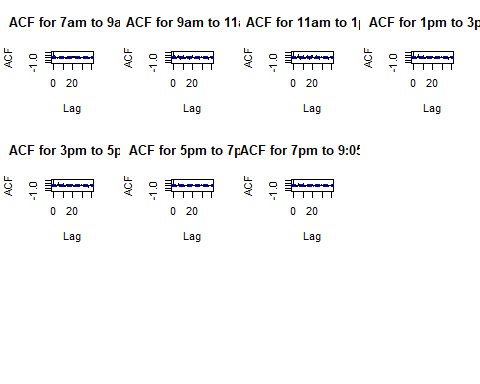
par(mfrow=c(3,4))  
plot.ts(aggregate[0:50,1], main = "7am to 9am")  
plot.ts(aggregate[0:50,2], main = "9am to 11am")  
plot.ts(aggregate[0:50,3], main = "11am to 1pm")  
plot.ts(aggregate[0:50,4], main = "1pm to 3pm")  
plot.ts(aggregate[0:50,5], main = "3pm to 5pm")  
plot.ts(aggregate[0:50,6], main = "5pm to 7pm")  
plot.ts(aggregate[0:50,7], main = "7pm to 9:05pm")  
  
  
par(mfrow=c(3,4))



print("Correlograms up to lag 40 for each interval are plotted below.")

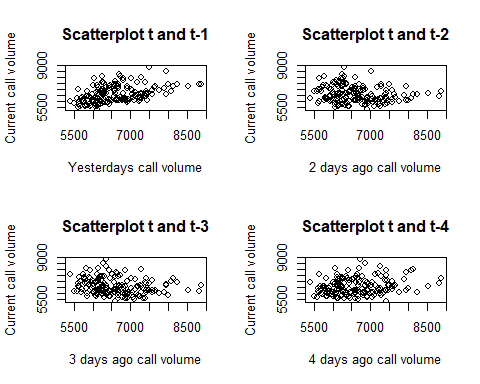
## [1] "Correlograms up to lag 40 for each interval are plotted below."

acf(aggregate[,1], lag.max = 40,   
 ylim = c(-1,1), main = "ACF for 7am to 9am",  
 na.action = na.pass)  
acf(aggregate[,2], lag.max = 40,   
 ylim = c(-1,1), main = "ACF for 9am to 11am",  
 na.action = na.pass)  
acf(aggregate[,3], lag.max = 40,   
 ylim = c(-1,1), main = "ACF for 11am to 1pm",  
 na.action = na.pass)  
acf(aggregate[,4], lag.max = 40,   
 ylim = c(-1,1), main = "ACF for 1pm to 3pm",  
 na.action = na.pass)  
acf(aggregate[,5], lag.max = 40,   
 ylim = c(-1,1), main = "ACF for 3pm to 5pm",  
 na.action = na.pass)  
acf(aggregate[,6], lag.max = 40,   
 ylim = c(-1,1), main = "ACF for 5pm to 7pm",  
 na.action = na.pass)  
acf(aggregate[,7], lag.max = 40,   
 ylim = c(-1,1), main = "ACF for 7pm to 9:05pm",  
 na.action = na.pass)



### Due to space constraints, I will only present the scatterplots for the 9am to 11am time interval

tt <- length(aggregate[,2])  
  
par(mfrow=c(2,2))  
plot(y=aggregate[2:tt,2], x=aggregate[1:(tt-1),2], ylab = 'Current call volume',  
 xlab = 'Yesterdays call volume', main = "Scatterplot t and t-1")  
plot(y=aggregate[3:tt,2], x=aggregate[1:(tt-2),2], ylab = 'Current call volume',  
 xlab = '2 days ago call volume', main = "Scatterplot t and t-2")  
plot(y=aggregate[4:tt,2], x=aggregate[1:(tt-3),2], ylab = 'Current call volume',  
 xlab = '3 days ago call volume', main = "Scatterplot t and t-3")  
plot(y=aggregate[5:tt,2], x=aggregate[1:(tt-4),2], ylab = 'Current call volume',  
 xlab = '4 days ago call volume', main = "Scatterplot t and t-4")



### Explanation:

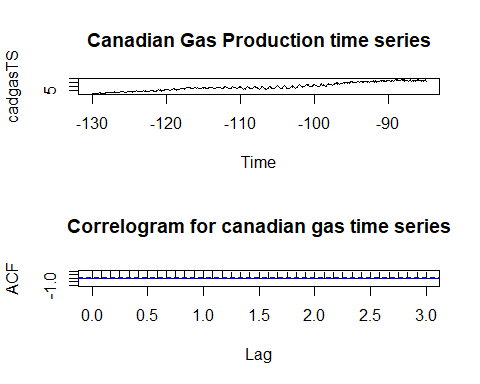
From the first 50 days plot, we can see that there are seasonal variations for this series. Interestingly, all plots start at a peak, and the first date of the series is March 3rd, 2003, which is a Monday; further, the number of peaks in the graphs is close to the number of Mondays in the time interval (50/5 ~ 10). We could make a hypothesis that bank calls are higher during Mondays. Possible reasons are: there is no customer service during the weekends (which is exhibited in the database), therefore, cases start queuing until they are resolved on Monday; or maybe, people wait until Monday to “get up to date” with financial responsibilities; or, most problems related with bank clients occur during the weekends (e.g., denied or lost cards), etc.

From the Autocorrelation Functions (ACF) we can observe that for most 2-hour blocks, the lags 1 and multiples of 5 (5,10, etc.) appear to be significant further motivating the Monday’s hypothesis. The rest of lags do not seem to be significant and decay very sharply. This indicates the bank calls are not auto correlated with previous days calls, which could be an indicator of successful customer service (a client whose problem has been solved wont call back) or rather a reflection of the characteristics of the call (for instance, these could be calls asking for one-off information or help, like “I cant access my online banking.” or “I lost my card, can you replace it?”).

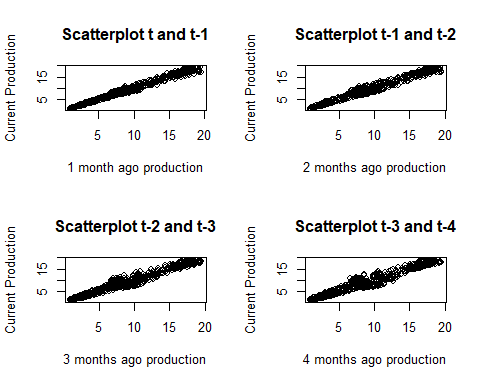
## 4.c.

Read the time series data canadian gas from the R package fpp3, giving monthly Canadian gas production, billions of cubic metres, January 1960- February 2005; Produce its time plot; As you will see, the series clearly has a trend and exhibits seasonal variations. Now, produce and include a correlogram of this series; Explain the pattern that you see; Supplement your explanation by drawing a few scatter plots of xt versus xt−h as the problems above.

cadgasTS = ts(canadian\_gas$Volume, canadian\_gas$Month, frequency=12)  
  
  
tt <- length(cadgasTS)  
  
par(mfrow=c(2,1))  
plot.ts(cadgasTS, main = "Canadian Gas Production time series")  
acf(cadgasTS, lag.max = 36,   
 ylim = c(-1,1), main = "Correlogram for canadian gas time series",  
 na.action = na.pass)



par(mfrow=c(2,2))  
plot(y=cadgasTS[2:tt], x=cadgasTS[1:(tt-1)], ylab = 'Current Production', xlab = '1 month ago production', main = "Scatterplot t and t-1")  
plot(y=cadgasTS[3:tt], x=cadgasTS[1:(tt-2)], ylab = 'Current Production', xlab = '2 months ago production', main = "Scatterplot t-1 and t-2")  
plot(y=cadgasTS[4:tt], x=cadgasTS[1:(tt-3)], ylab = 'Current Production', xlab = '3 months ago production', main = "Scatterplot t-2 and t-3")  
plot(y=cadgasTS[5:tt], x=cadgasTS[1:(tt-4)], ylab = 'Current Production', xlab = '4 months ago production', main = "Scatterplot t-3 and t-4")



### Explanation:

From the Autocorrelation Functions (ACF) we can observe that lags, even up to 36, are significant and decay very slowly. This indicates the Canadian gas monthly production today are auto correlated with prices up to 36 months prior. This could be because the series is not stationary (although we cannot state it from the ACFs alone), which is exhibited by the seasonality variations and trend in the plot. Additionally, from the scatter plots we can see that current production is correlated with previous months production. Some reasons may that shocks that affect gas production are persistent through time, gas production could be regulated and quotas implemented, gas is consumed the highest during winter months, among others.