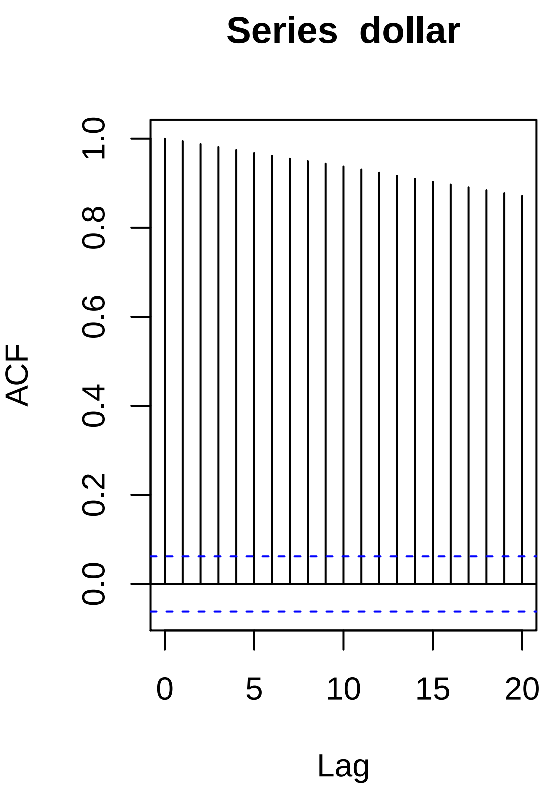
STA 457 Assignment 1

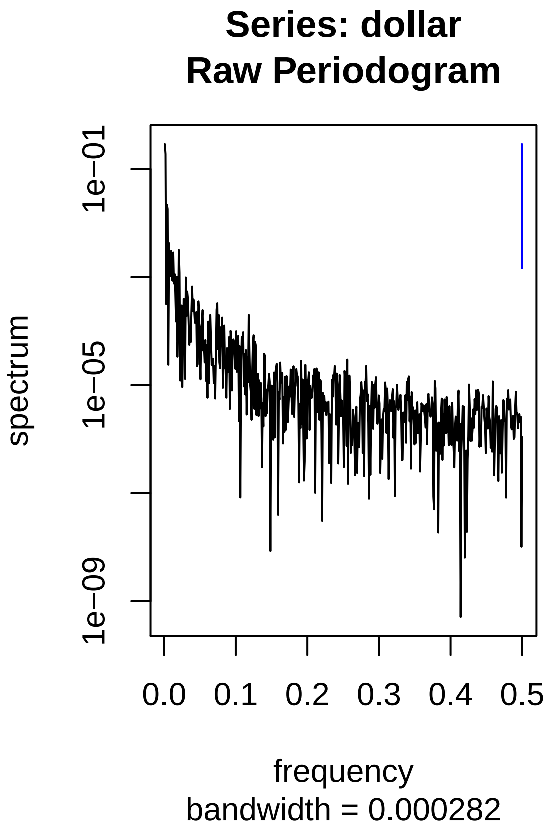
Last name: Deng
First name: Qi (Christina)
Student ID: 1001142408
Course section: STA457H1S

Jan. 21, 2017

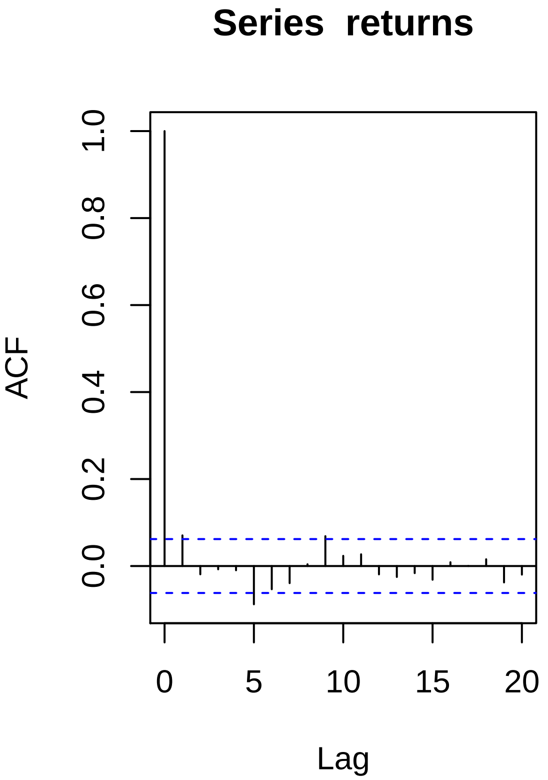
Q1:

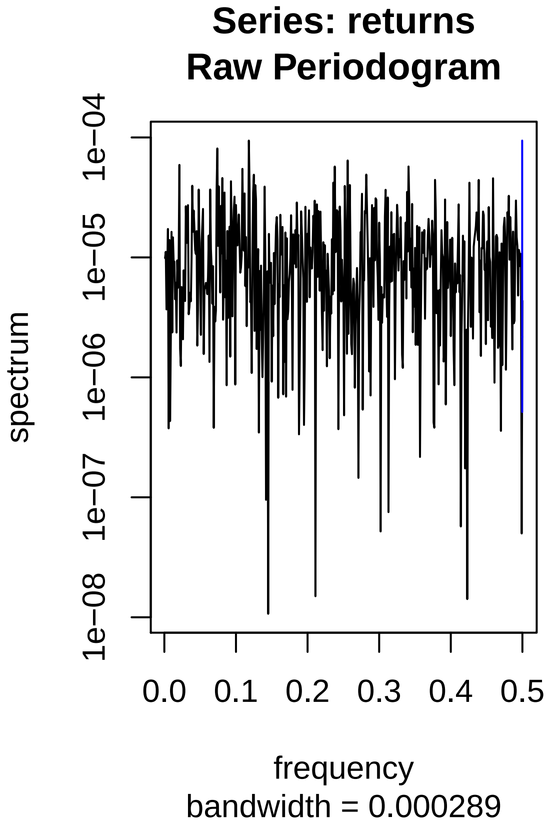
(a) Plot the correlogram and periodogram of the orginal data





(b) Plot the correlogram and periodogram of the first differences.





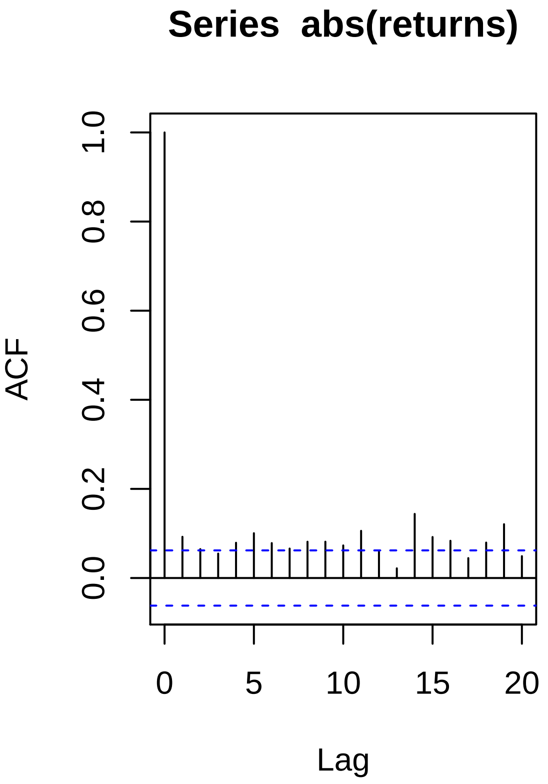
(c) Comment on the results obtained in parts (a) and (b).

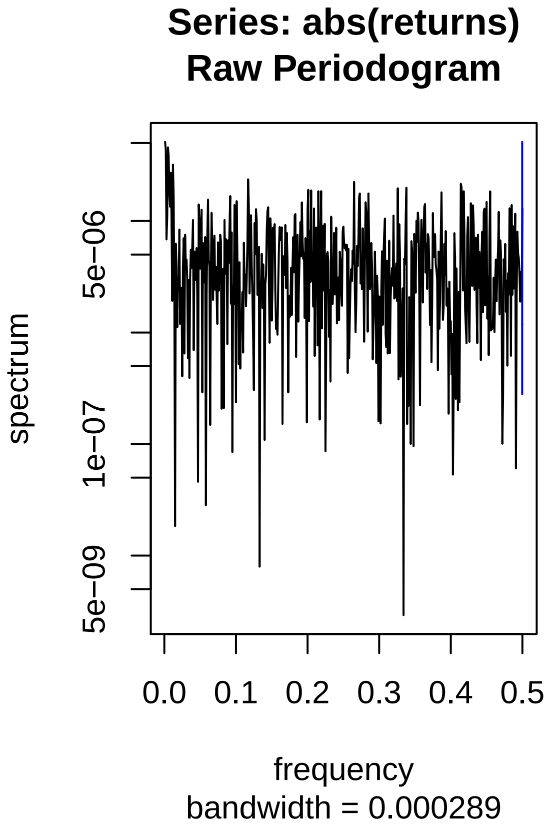
- In part (a), the ACF in correlogram decay to zero very **slowly** and **monotonically**, every ACF is above the upper dotted line. We can conclude that the time series has **long memory**. Slowly decaying autocorrelations correspond to periodograms that have **larger values for low frequencies**. The periodogram in part (a) demonstrates a **negative sloping** graph, and the original data reached the **maximum** when f is equal to zero.

- While in part (b), the ACF decay much **quickly**, most of ACF are inside the dotted lines and form a sin function. By definition, the time series has **short memory**. short memory indicates that immediate past gives some information about short term future but essentially no information about long term future. Note that White Noise is a speacial case of shot-memory. By looking at its periodogram, f **does not obtain maximum** at zero, and there is **no obious peak**, that is because there are **uncorrelated** random variables with zero mean and finite variance. There are **no trend** for the periodogram.

- By definition of White Noise & Random Walk, White Noise is the time series generated from uncorrelated variables is used as a model for noise, and Random Walk is known as a stochastic or random process, that describes a path that consists of a succession of random steps. According to the sample plots posted online, part (a) is consistent with **random walk** since the next data is always related to the previous one, and part (b) is more like **white noise** as all datas are irrelevant. In addition, (a) is not stationary while (b) is weak stationary.

(d) Now look at the correlogram and periodogram of the absolute values of the first differences. Comment on the differences between the results for returns and abs(returns), in particular, with respect to the applicability of the random walk model.

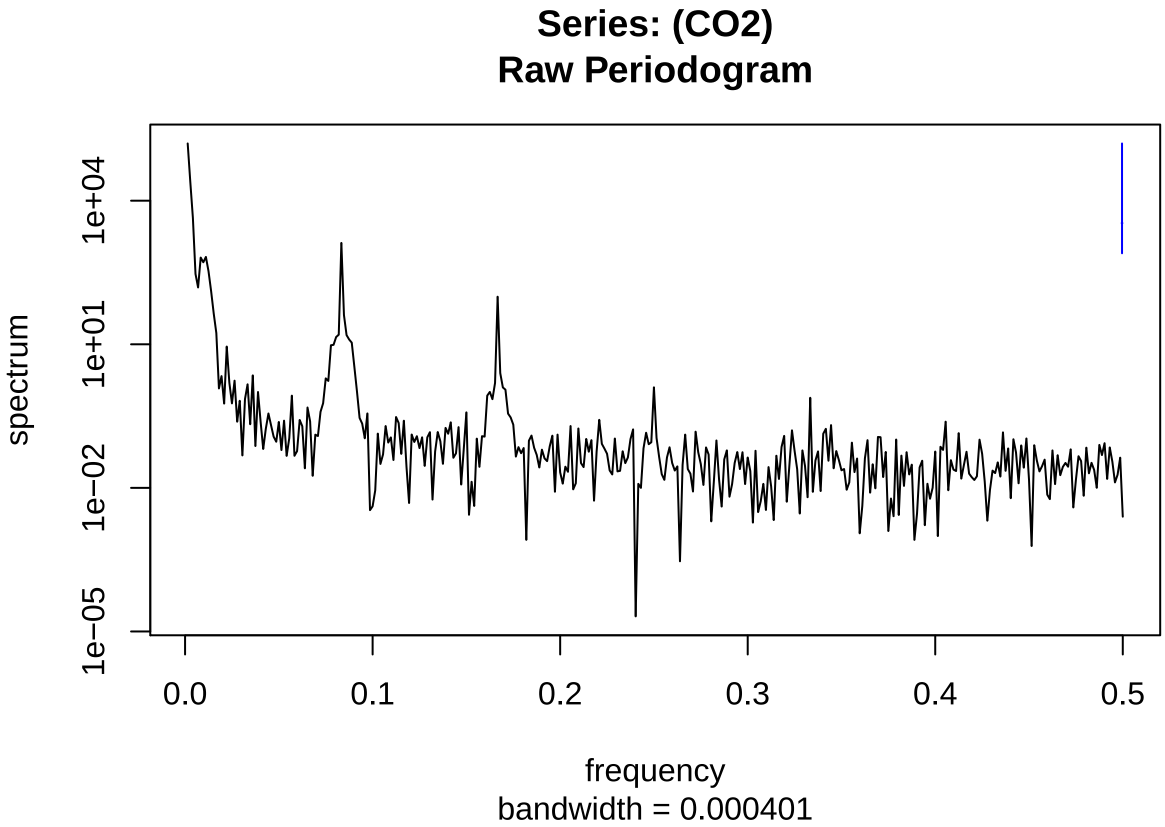




- From the correlogram, we conclude that the ACF decay **slower** than ACF without taking absolute value. There are **decreasing and increasing** of ACF, but most of ACF are above the upper dotted line and all ACF are positive unlike in part (b). Indeed, the data has different pattern than both first two datas.
- Moreover, its periodogram shows that f obtains its **maximum at zero** just like what we observe in part (a), but different than part (b).
- In term of applicability, absolute values of first differences cannot model either white noise nor random walk. Random Walk demonstrates a slowly and monotonically decreasing trend, while white noise contains irrelavant datas. Thus, abs(returns) **cannot** model either of Random Walk or White Noise.

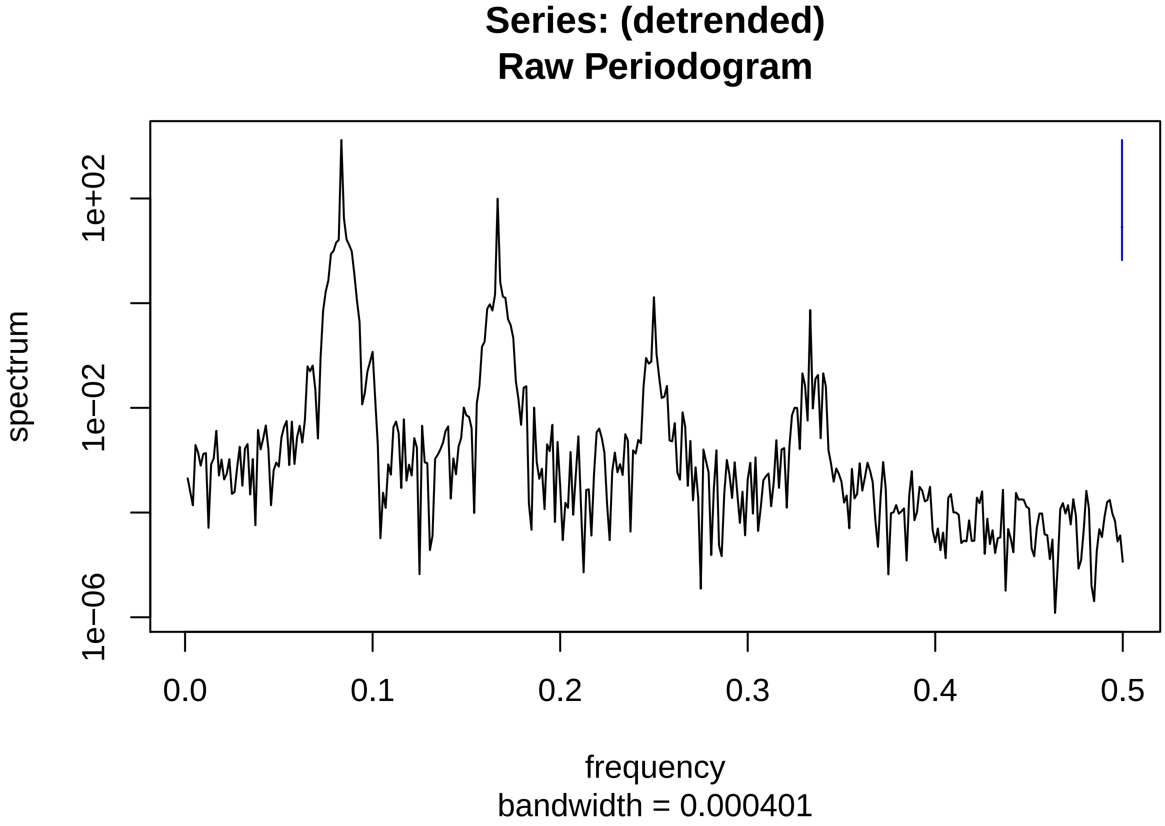
Q2

(a) Plot the periodogram of the time series. At what frequencies are there peaks? To which features of the time series do these peaks correspond?



- The peaks are obtained at f=0, 0.085, 0.17, 0.255.
- The peaks are reached at multiple of 0.085.

(b) Subtract the trend from the original data and look at the periodogram of the detrended data. Comment on the differences between the periodograms in parts (a) and (b). (It is useful here to overlay the two periodograms on the same plot.) In particular, how effective is the detrending in emphasizing the seasonality in the data?



- Unlike the original data, the detrended data has **no peaks at f=0**, since the trend is subtracted. But, the rest of the peaks occur at the **same frequencies** for both datas. A time series contains trend, seasonal, and noise terms, thus, by detrending, seasonal term will **still exist**, which means that the rest of peaks stay unchanged.

Source R code

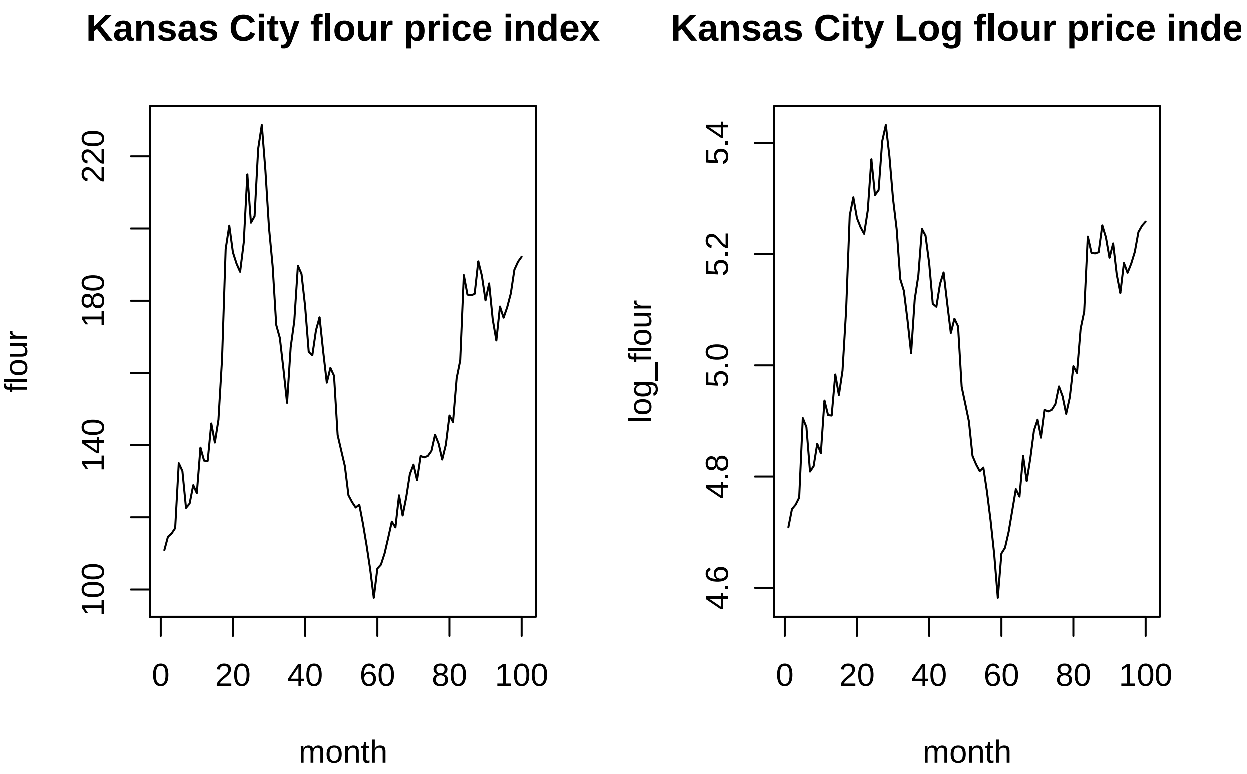
```r
# ---------> complete and run the following code for this assignment  <-------
#
# R code for STA457 assignment 1
# copyright by Christina (Qi) Deng
# date: Jan 21, 2017
#

## Q1:
# load in data
dollar <- scan ("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 1/dollar.txt")
# get the time series of log(dollar)
dollar <- ts(log(dollar))
# define the first difference
returns <- diff(dollar)
# (a) Plot the correlogram and periodogram of the orginal data (i.e. dollar)
acf(dollar,lag.max=20)
spec.pgram(dollar,demean=T,detrend=F)
# (b) Plot the correlogram and periodogram of the first differences.
acf(returns, lag.max=20)
spec.pgram(returns,demean=T,detrend=F) # max not at 0
# (c) Analysis
# (d) Plot the correlogram and periodogram of the absolute values of the first difference
acf(abs(returns), lag.max=20)
spec.pgram(abs(returns),demean=T,detrend=F) # max at 0

# Q2:
# load in data
CO2 <- scan("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 1/CO2.txt")
CO2trend <- scan("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 1/CO2-trend.txt")
# (a) Plot the periodogram of the time series.
spec.pgram((CO2),demean=T,detrend=F)
# (b)Plot periodogram of the detrended data.
detrended=CO2-CO2trend
spec.pgram((detrended), demean=T,detrend=F)
```

STA 457 Assignment 2

Last name: Deng
First name: Qi (Christina)
Student ID: 1001142408
Course section: STA457H1S

Feb 14, 2017

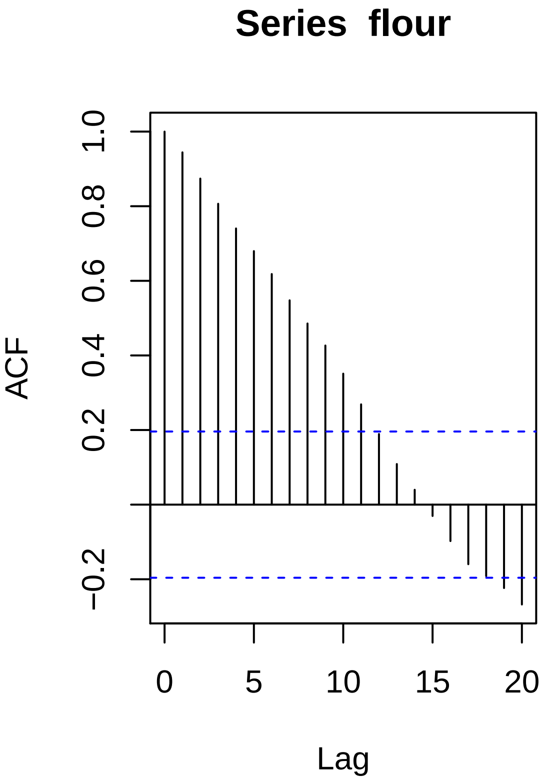
Q2

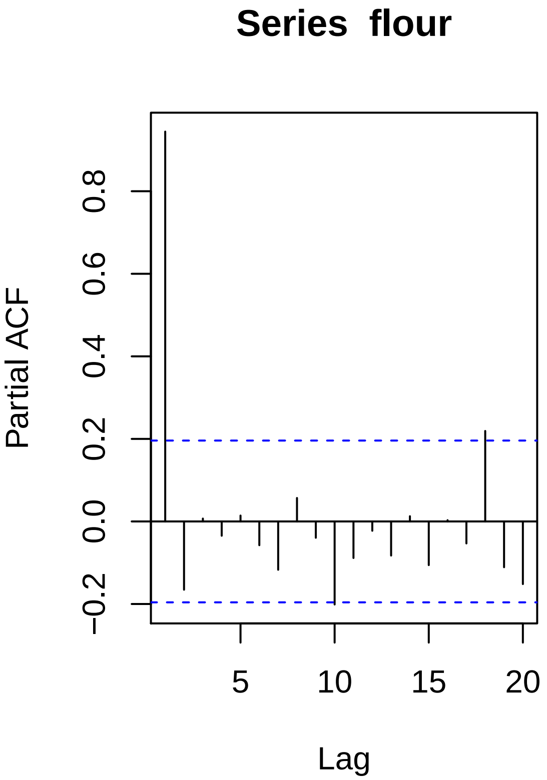


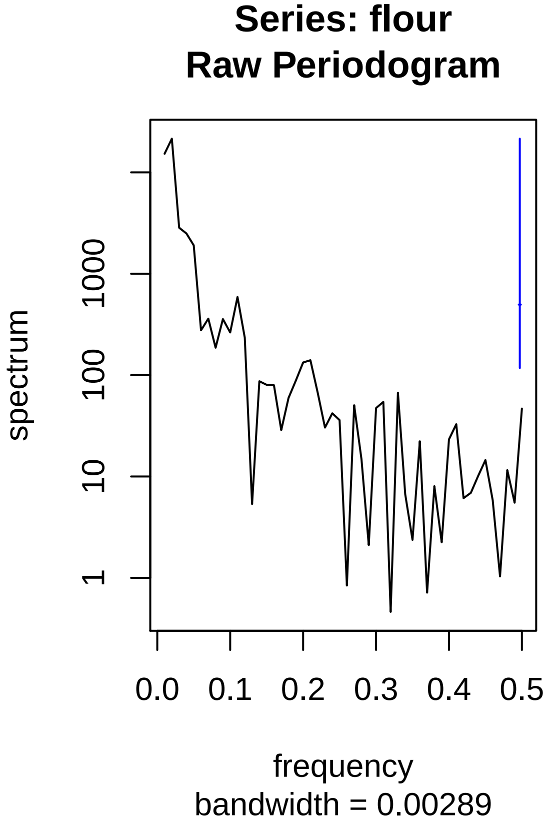
(a) Look at a time series plot of the data. Are there any obvious trends and/or periodicities in the data? Would it be worthwhile to transform the data?

- By looking at the first plot, there is **no obvious trends or periodicties** in the data. Taking log flour, the second plot is the same as the original plot; thus, it ia **no worthwhile** to transform the data.

(b) Plot the correlogram, partial correlogram and periodogram of the data.







(c) Which of the following models seems to be most appropriate for these data: an autore-gressive model, a moving average model or a random walk (ARIMA(0,1,0)) model?

```
MA1 <- arima(flour,c(0,0,1))
AR1 <- arima (flour, c(1,0,0))
RW <- arima(flour,c(0,1,0))
MA1$aic
```

```
## [1] 872.1211
```

```
AR1$aic
```

```
## [1] 722.5832
```

```
RW$aic
```

```
## [1] 710.1471
```

```
Box.test(MA1$residuals)
```

```
##
##  Box-Pierce test
##
## data:  MA1$residuals
## X-squared = 50.616, df = 1, p-value = 1.123e-12
```

```
Box.test(AR1$residuals)
```

```
##
##  Box-Pierce test
##
## data:  AR1$residuals
## X-squared = 4.7492, df = 1, p-value = 0.02931
```

```
Box.test(RW$residuals)
```

```
##
##  Box-Pierce test
##
## data:  RW$residuals
## X-squared = 4.6412, df = 1, p-value = 0.03121
```

- By the above data, **AR(1) and Random Walk** are the most appropriate models for these data. Since we choose not to transform the data, AR(1) will be the more appropriate one. AR(1) and Random Walk models have **smaller AIC and larger p-value**, thus, will be beter models for these data.

Source R code

```r
# ---------> complete and run the following code for this assignment  <-------
#
# R code for STA457 assignment 2
# copyright by Christina (Qi) Deng
# date: Feb 14, 2017
#



# Q2:
# load in data
flour <- scan ("/Users/christinadeng/Desktop/Winter 2017/STA 457/ Assignment 2/flour.txt")
# get the time series of flour data
flour <- ts(flour)
# (a) get time series plot of the data
par(mfrow=c(1,2))
plot(flour,main="Monthly values of the Kansas City flour price index",xlab="month")
log_flour <- ts(log(flour))
plot(log_flour,main="Monthly values of the Kansas City Log flour price index",xlab="month")
# (b) Plot the correlogram, partial correlogram and periodogram of the data.
acf(flour,lag.max=20)
pacf(flour, lag.max=20)
spec.pgram(flour,demean=T,detrend=F)
# (c) Which of the following models seems to be most appropriate for these data: an autore-gressive model, a moving
install.packages("forecast")
library(forecast)
auto.arima(flour)
# get MA(1), AR(1), AND Random Walk Model
MA1 <- arima(flour,c(0,0,1))
AR1 <- arima (flour, c(1,0,0))
RW <- arima(flour,c(0,1,0))
# generate AIC output
MA1$aic
AR1$aic
RW$aic
# do Box.test to check their p-value
Box.test(MA1$residuals)
Box.test(AR1$residuals)
Box.test(RW$residuals)
```

STA 457 Assignment 3

Last name: Deng
First name: Qi (Christina)
Student ID: 1001142408
Course section: STA457H1S

March 14, 2017

Q1

(a) Carry out the Augmented Dickey-Fuller (ADF) test of the null hypothesis that the data have a unit root for various lags. What is your conclusion?

```
## Warning: package 'tseries' was built under R version 3.3.2
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dollar
## Dickey-Fuller = -2.0419, Lag order = 5, p-value = 0.5606
## alternative hypothesis: stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dollar
## Dickey-Fuller = -2.1711, Lag order = 10, p-value = 0.5059
## alternative hypothesis: stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dollar
## Dickey-Fuller = -2.0697, Lag order = 15, p-value = 0.5488
## alternative hypothesis: stationary
```

Conclusion:

Carry out the ADF test of null hypothesis that the data have a unit root for various lags. (ie. h0: have unit root; hence non-stationary) Let's take lag=5, 10 and 15, we get ADF tests have p-values as **0.5606**, **0.5059** and **0.5488** respectively, and we observed that they are all **greater than 0.05**. Thus, we **fail to reject** the null hypothesis, which means that the Time Series of data on log-scale **have a unit root for various lags**, thus it is **non-stationary**.

(b) Do the first differences seem to be close to white noise? Use Bartlett's test and the Box-Pierce (Portmanteau) test to test this. Computes the statistic defined in class and its p-value

```
## $stat
## [1] 1.395205
##
## $p.value
## [1] 0.04076001
```

```
##
##  Box-Ljung test
##
## data:  dollar_d
## X-squared = 13.296, df = 5, p-value = 0.02076
```

```
##
##  Box-Ljung test
##
## data:  dollar_d
## X-squared = 23.075, df = 10, p-value = 0.01047
```

```
##
##  Box-Ljung test
##
## data:  dollar_d
## X-squared = 26.112, df = 15, p-value = 0.03686
```

Conclusion:

By applying Bartlett's test and Box-Pierce test to check if the first differences seem to be close to white noise (H0: correlation is zero, it is White Noise), we oberved that

- Bartlett test: $stat is 1.395205; $p.value is 0.04076001
- Box-Pierce with lag=5: X-squared = 13.296; p-value = 0.02076
- Box-Pierce with lag=10: X-squared = 23.075; p-value = 0.01047
- Box-Pierce with lag=15: X-squared = 26.112; p-value = 0.03686

All the p-values are **less than 0.05**, thus we **reject** the null hypothesis, which means that the first differences seem **not** to be close to white noise. Therefore, using White Noise to model the first differences is **inappropriate**.

Q2

(a) Do the data appear to be non-stationary? Does a random walk (ARIMA(0,1,0)) model appear to fit the data? (Use formal and informal white noise tests here.)

```
##
##  Augmented Dickey-Fuller Test
##
## data:  yield
## Dickey-Fuller = -3.8625, Lag order = 5, p-value = 0.01637
## alternative hypothesis: stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  yield
## Dickey-Fuller = -3.0598, Lag order = 10, p-value = 0.1296
## alternative hypothesis: stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  yield
## Dickey-Fuller = -3.1956, Lag order = 15, p-value = 0.08907
## alternative hypothesis: stationary
```

```
## $stat
## [1] 2.439901
##
## $p.value
## [1] 1.349638e-05
```

```
##
##  Box-Ljung test
##
## data:  yield_d
## X-squared = 46.44, df = 10, p-value = 1.194e-06
```

Conclusion:

For the first question here, we **cannot conclude** the data is stationary or non-stationary, since by doing adf.test, we have p-value less than 0.05 when lag is 5, and have p-value greater than 0.05 when lag is 10 or 15. According to our previous knowledge, we know that for the data to be Random Walk, its first difference has to be White Noise. However, by our formal and informal white noise tests here, p-value are **less than 0.05**, thus we **reject** the null hypothesis. In this question, h0 implies the data is White Noise (ρ=0), then rejecting h0 means the first difference is **not White Noise**. Therefore, we conlcude that **Random Walk is not the best model** for the data here.

(b) Using the arima function in R, fit ARIMA(p,1,p) models. Use AIC to determine the best fitting ARIMA(p,1,p) model.

```
arima1$aic
```

```
## [1] -119.3551
```

```
arima2$aic
```

```
## [1] -119.6975
```

```
arima3$aic
```

```
## [1] -131.1786
```

Conclusion:

By the above data, we observed that aic for ARIMA(1,1,1) is -119.3551, aic for ARIMA(2,1,2) is -119.6975, and aic for ARIMA(3,1,3) is -131.1786. We want the model with smallest aic, thus ARIMA(3,1,3) is the best fitting model here.

(c) Try a few other models and look at AIC for each of them, Which ARIMA(p,1,q) model seems to be the best?

```
arima_n1=arima(yield, order=c(2,1,3))
arima_n1$aic
```

```
## [1] -118.0788
```

```
arima_n2=arima(yield, order=c(3,1,2))
arima_n2$aic
```

```
## [1] -118.0088
```

```
arima_n3=arima(yield, order=c(1,1,2))
arima_n3$aic
```

```
## [1] -118.9223
```

```
arima_n4=arima(yield, order=c(2,1,1))
arima_n4$aic
```

```
## [1] -120.1476
```

```
arima_n5=arima(yield, order=c(1,1,3))
arima_n5$aic
```

```
## [1] -119.8564
```

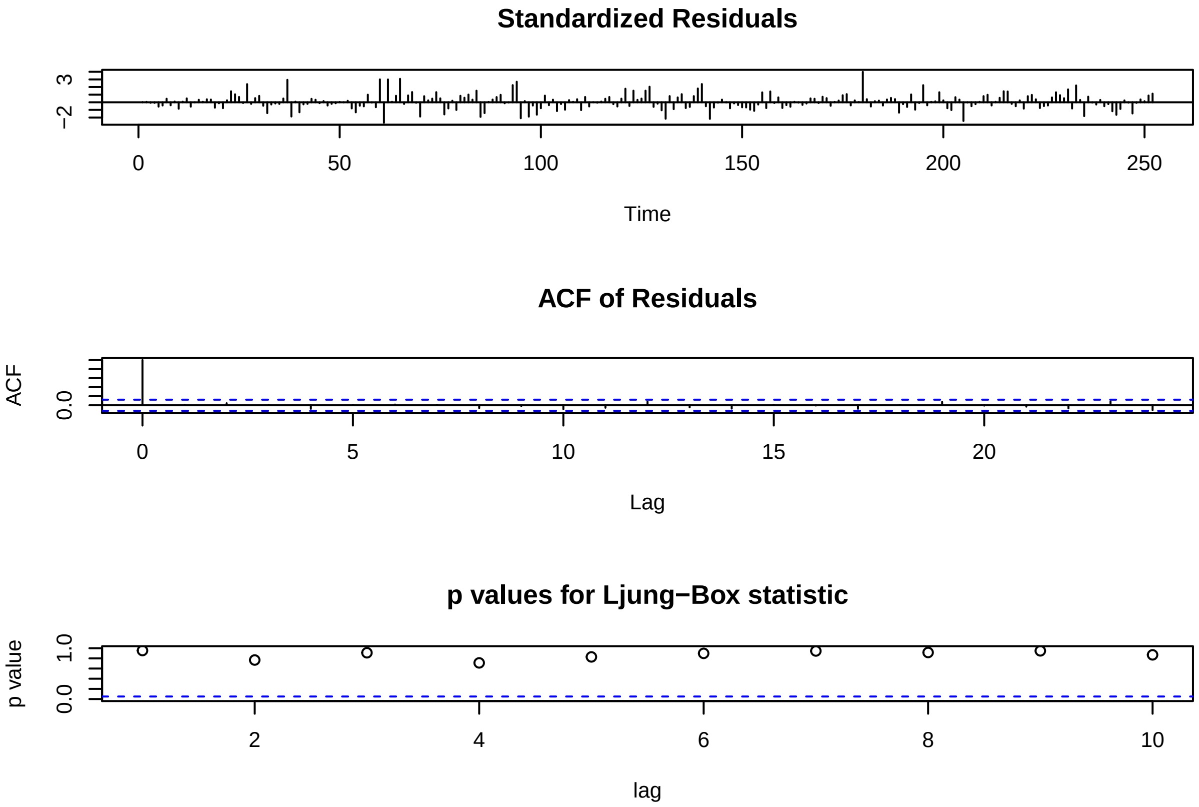
```
arima_n6=arima(yield, order=c(3,1,1))
arima_n6$aic
```

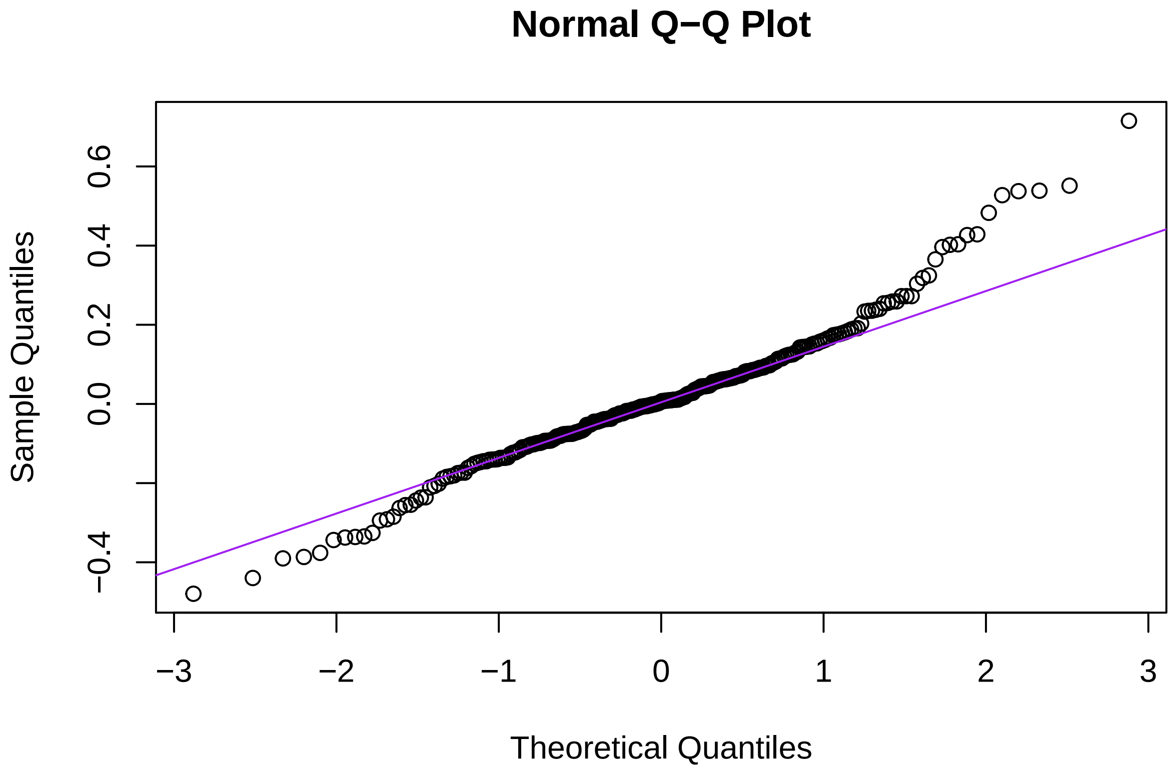
```
## [1] -119.8777
```

Conclusion:

By above calculation, we observed that aic of ARIMA(3,1,3) still have the smallest aic value than all other ARIMA models, thus we conclude that ARIMA(3,1,3) seems the best.

(d) For your best model, are the residuals close to a white noise process? Also check the normality of the residuals.



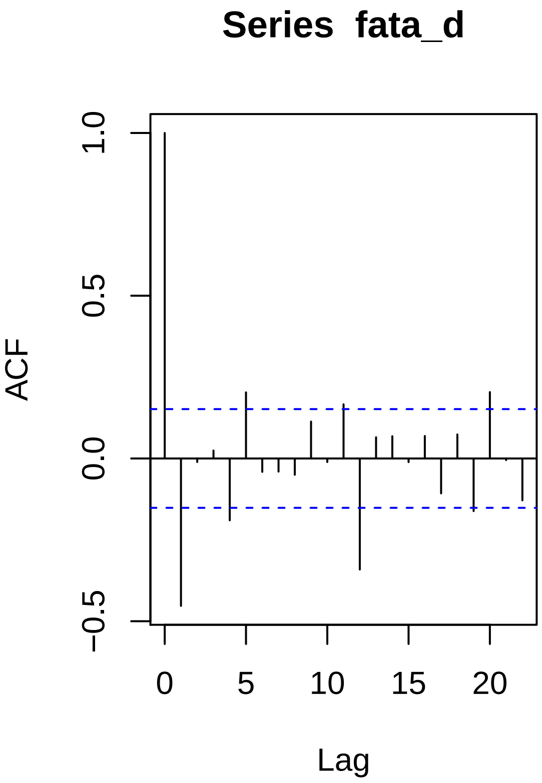


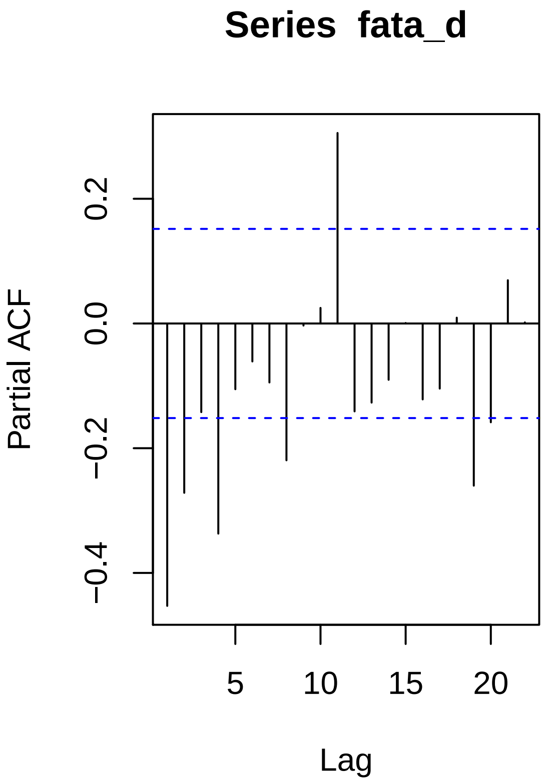
Conclusion:

By the ACF graph of residual, the acf decays very quickly, it is the property of **White noise**. Moreover, by definition, a normal Q-Q plot is a graphical tool to assess if a set of observations is normally distributed or not. If it does, then a normal QQ-plot of the observations will result in an approximately straight line. From our graph, most data points lie on straight line and this **normality** assumption seems good. By the p-value graph, all points are **above** 0.05, thus, we **fail to reject** the null hypothesis, then we conclude that **ARIMA(3,1,3) is a good model with residuals close to a white noise process and has normality**.

Q3

(a)Fit a seasonal ARIMA (SARIMA) model to the data using the following steps:



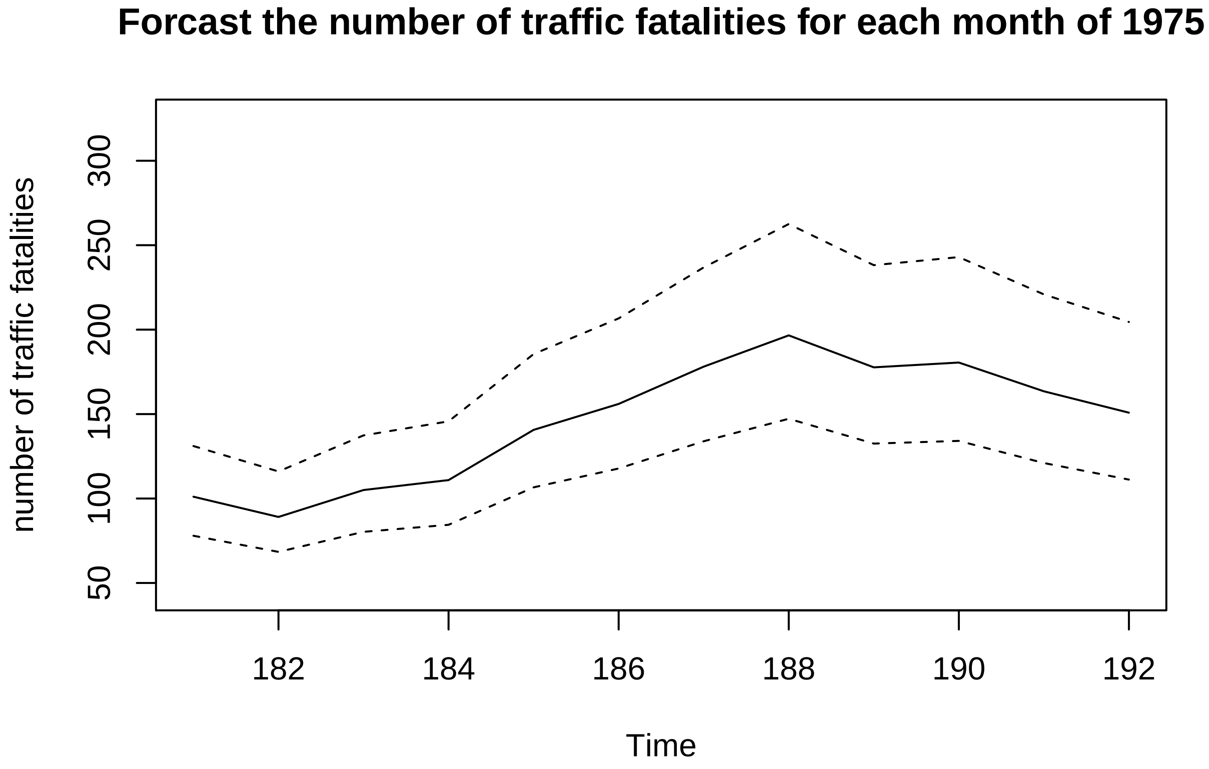


```r
f1=arima(fatalities,order=c(0,1,1), seasonal=list(order=c(0,1,1),period=12))
bartlett(f1$residuals)
```

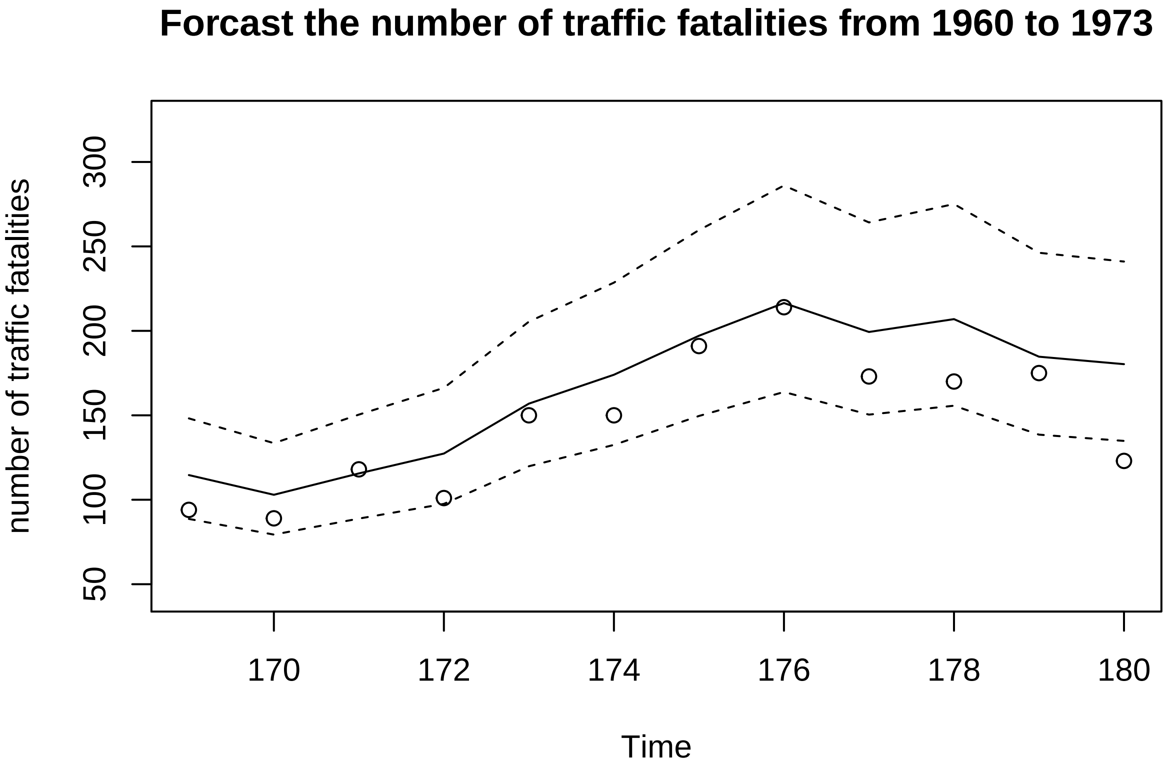
```
## $stat
## [1] 0.7034494
##
## $p.value
## [1] 0.7054855
```

Conclusion: From the correlogram and partial correlogram, also from the hint given, we choose SARIMA(0,1,1)x(0,1,1), and period = 12 (from the correlogram, there is a peak at lag=1). By calculation, we got **stat = 0.7034494** and **p.value = 0.7054855 > 0.05**. Therefore, we **fail** to reject the null hypothesis, the residuals from model are **close to white noise**.

(b) Use the R function predict to forecast the number of traffic fatalities for each month of 1975. Give 95% prediction limits for your forecasts.



(c) Use the data from 1960 to 1973 to estimate the parameters of your "best" model from part (a). Then use these parameter estimates to forecast the number of fatalities for 1974 and obtain 95% prediction limits. How do the true number of fatalities compare to these forecasts?



Conclusion:

From the graph, we conclude that the prediction here is accurate and appropriate. The expected value is very close to the actual value, since most of the circles are inside the prediction interval. Therefore, we say that the true number of fatalities are approximate to these forcasts.

```
# ---------> complete and run the following code for this assignment  <-------
#
# R code for STA457 assignment 3
# copyright by Christina (Qi) Deng
# date: March 14, 2017
#

#1(a)
library(tseries)
dollar <- scan ("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 1/dollar.txt")
dollar <- ts(log(dollar))
adf.test(dollar,k=5)
adf.test(dollar,k=10)
adf.test(dollar,k=15)

#(b)
dollar_d = diff(dollar)
bartlett <- function(x,plot=F) {
   x <- as.vector(x)
   x <- x - mean(x)
   n <- length(x)
   m <- floor(n/2)+1
   s <- Mod(fft(x)[1:m])^2/n
   sums <- sum(s)
   cumper <- cumsum(s)/sums
   v <- c(0:(m-1))/(m-1)
   upper <- v + 2*sqrt(v*(1-v))/sqrt(n/2)
   lower <- v - 2*sqrt(v*(1-v))/sqrt(n/2)
   a <- max(abs(cumper-v))*sqrt(n/2)
   sgn <- 1
   j <- 1
   incr <- 2*exp(-2*a^2)
   pval <- incr
   while ( incr > 1e-5 ) {
      sgn <- -sgn
      j <- j + 1
      incr <- 2*exp(-2*a^2*j^2)
      pval <- pval + sgn*incr
      }
   tit <- paste("Bartlett's statistic=",round(a,3),"p-value=",round(pval,4))
   if (plot) {
      plot(v,cumper,xlab=" ",ylab="Cumulative periodogram",pch=20)
      abline(0,1)
      lines(v,upper,lty=2)
      lines(v,lower,lty=2)
      title(sub=tit)
      }
   bartlett <- a
   pvalue <- pval
   r <- list(stat=bartlett,p.value=pvalue)
   r
}
```

```r
bartlett(dollar_d)
Box.test(dollar_d, lag=5,type="Ljung")
Box.test(dollar_d, lag=10, type="Ljung")
Box.test(dollar_d, lag=15, type="Ljung")

#2(a)
yield = scan("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 3/yield.txt")
yield_d = diff(ts(yield))
adf.test(yield, k=5)
adf.test(yield, k=10)
adf.test(yield, k=15)
bartlett(yield_d)
Box.test(yield_d, lag=10, type="Ljung")

#(b)
arima1=arima(yield, order=c(1,1,1))
arima2=arima(yield, order=c(2,1,2))
arima3=arima(yield, order=c(3,1,3))
arima1$aic
arima2$aic
arima3$aic

#(d)
tsdiag(arima3)
qqnorm(arima3$residuals)
qqline(arima3$residuals,col="purple", pch=21)


#3(a)
fatalities <- scan("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 3/fatalities.txt")
fatalities <- ts(log(fatalities))
fata_d <- diff(diff(fatalities),12)
par(mfrow=c(1,2))
acf(fata_d)
pacf(fata_d)
f1=arima(fatalities,order=c(0,1,1), seasonal=list(order=c(0,1,1),period=12))
bartlett(f1$residuals)


#(b)
P_f1 <- predict(f1,n.ahead=12)
plot(exp(P_f1$pred),ylim=c(45, 325), ylab="number of traffic fatalities", main="Forcast the number of tr
lines(exp(P_f1$pred + 1.96*P_f1$se),lty=2)
lines(exp(P_f1$pred - 1.96*P_f1$se),lty=2)

#(c)
fatalities2 <- fatalities[1:168]
f2=arima(fatalities2,order=c(0,1,1), seasonal=list(order=c(0,1,1),period=12))
P_f2 <- predict(f2,n.ahead=12)
plot(exp(P_f2$pred),ylim=c(45, 325), ylab="number of traffic fatalities", main="Forcast the number of tr
lines(exp(P_f2$pred + 1.96*P_f2$se),lty=2)
lines(exp(P_f2$pred - 1.96*P_f2$se),lty=2)
points(c(169:180),exp(fatalities[169:180]))
```

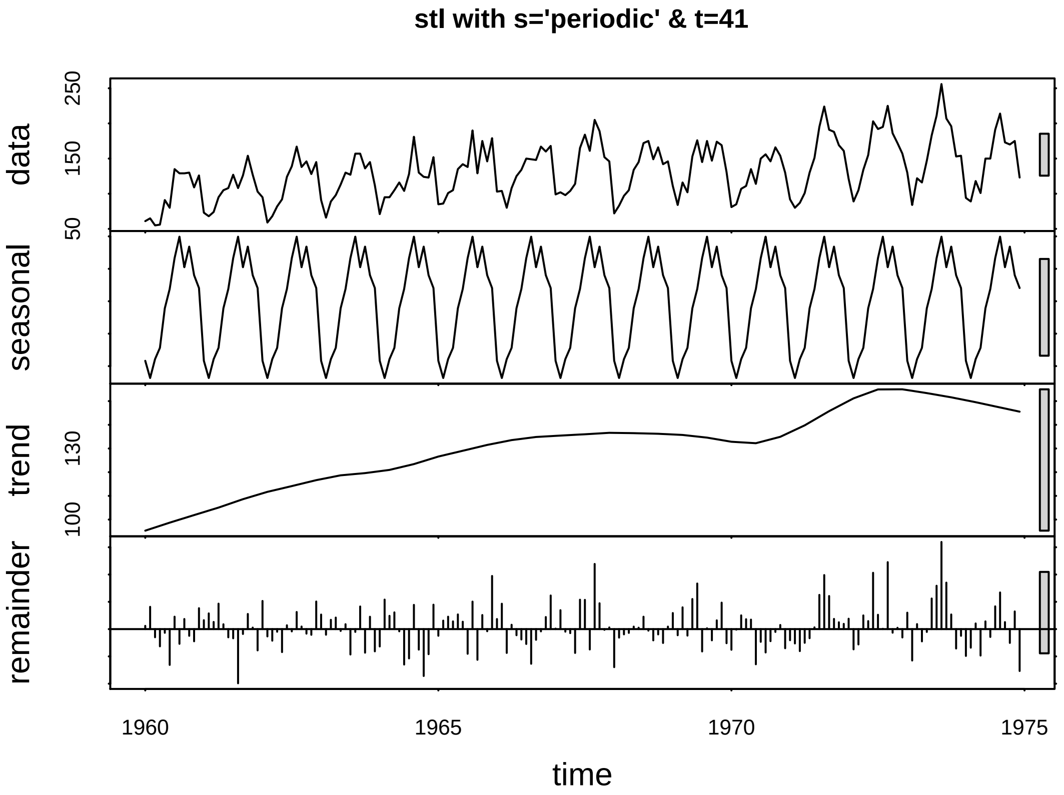
STA 457 Assignment 4

Last name: Deng
First name: Qi (Christina)
Student ID: 1001142408
data: STA457H1S

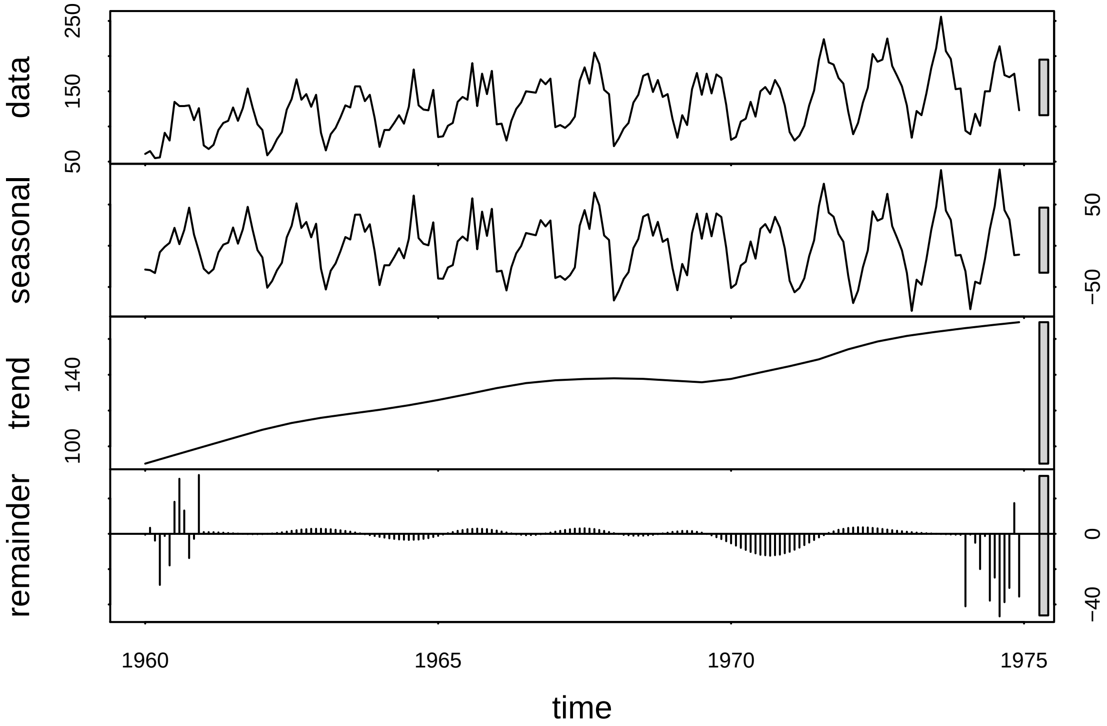
Mar 27, 2017

Q1

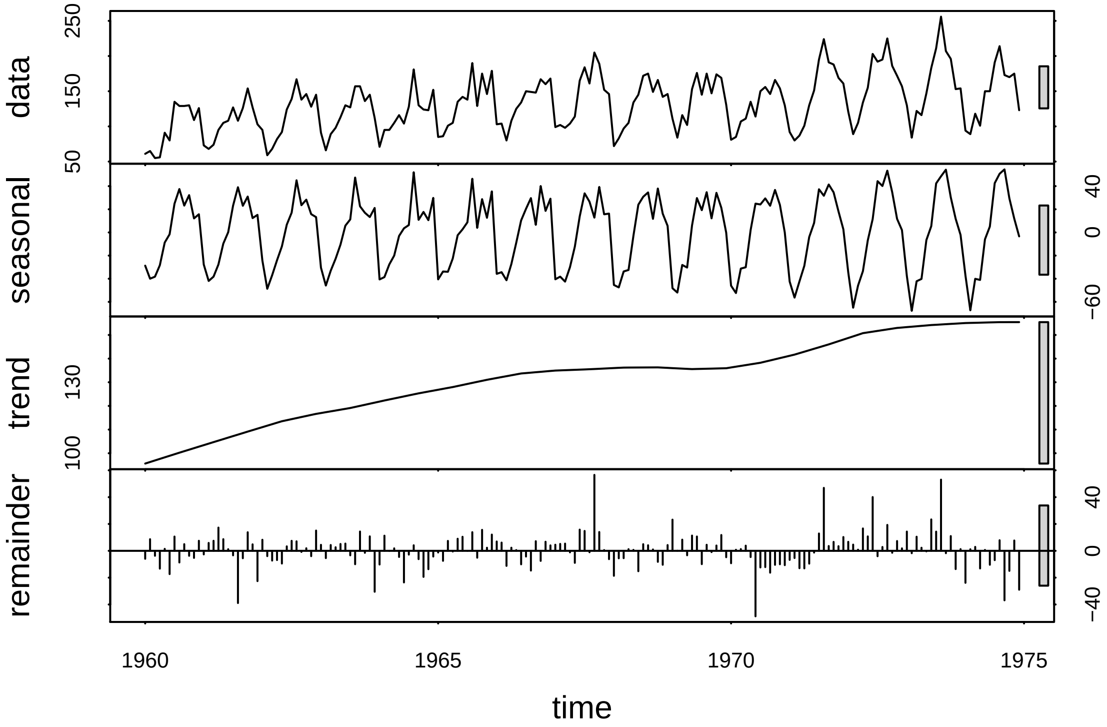
(a) Use the function stl to seasonally adjust the data. The two key tuning parameters in stl are s.window and t.window, which control the number of observations used by loess in the estimation of the seasonal and trend components, respectively.

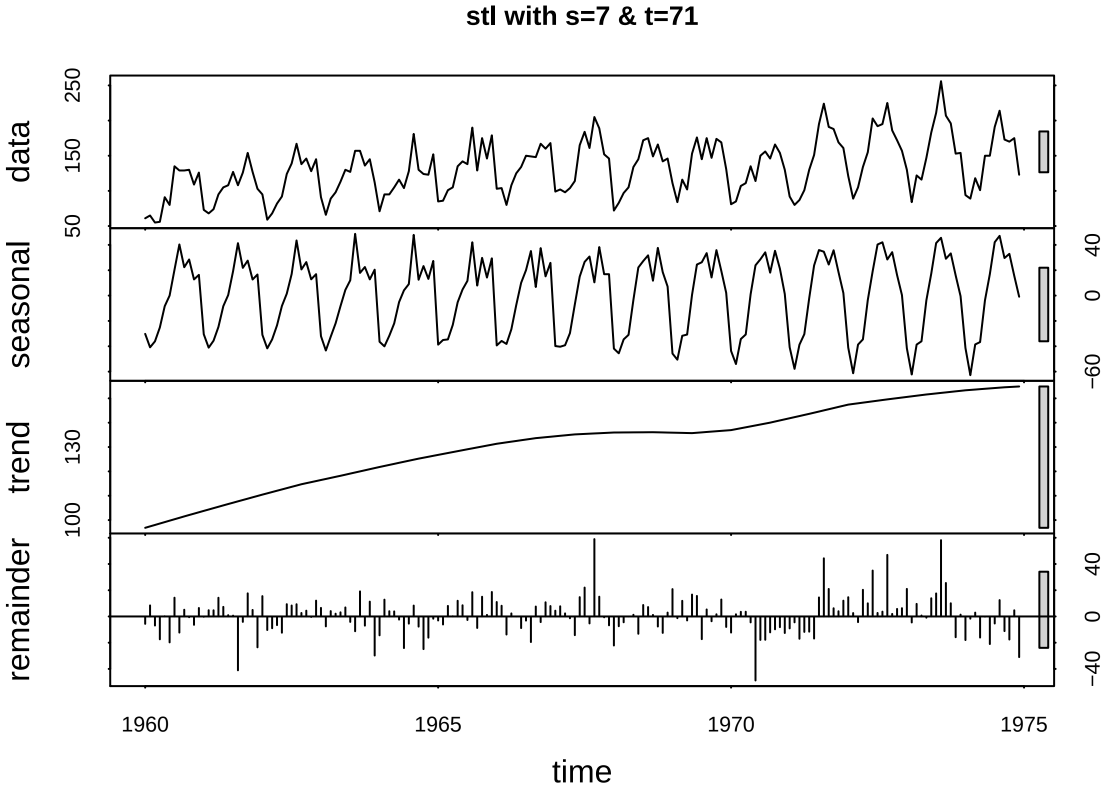


stl with s=3 & t=51



stl with s=5 & t=61

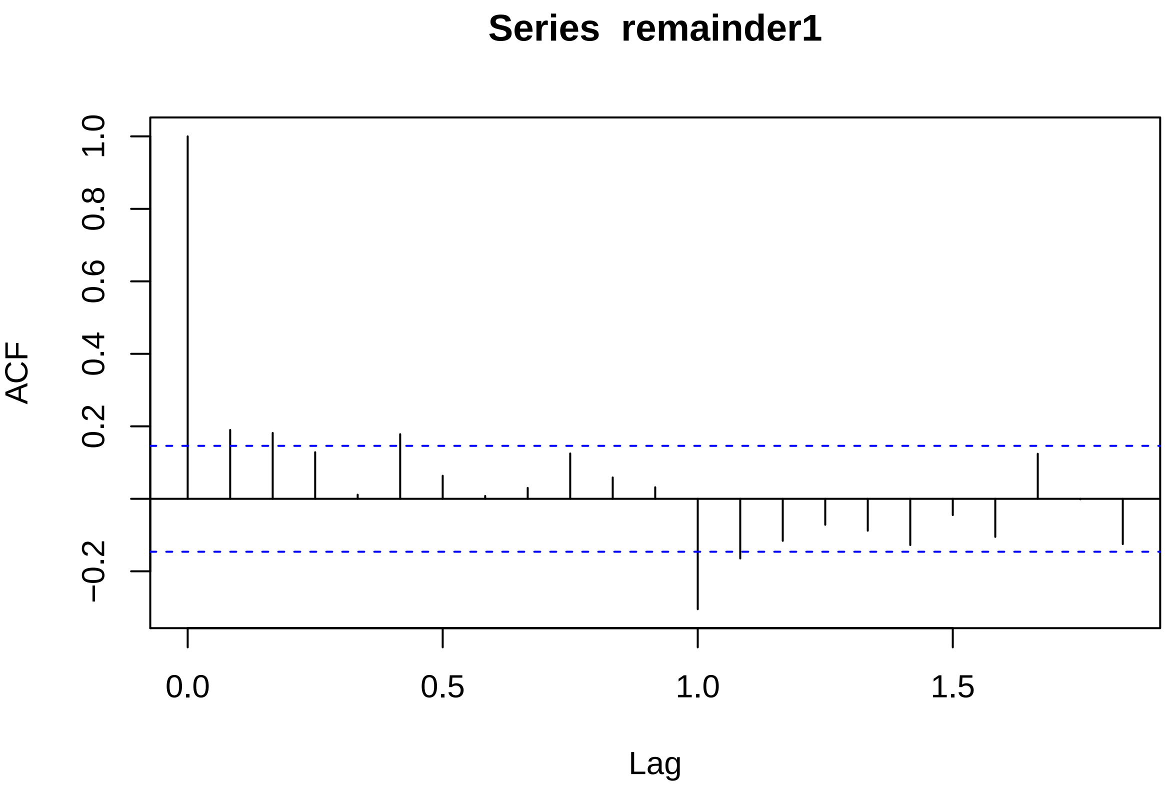


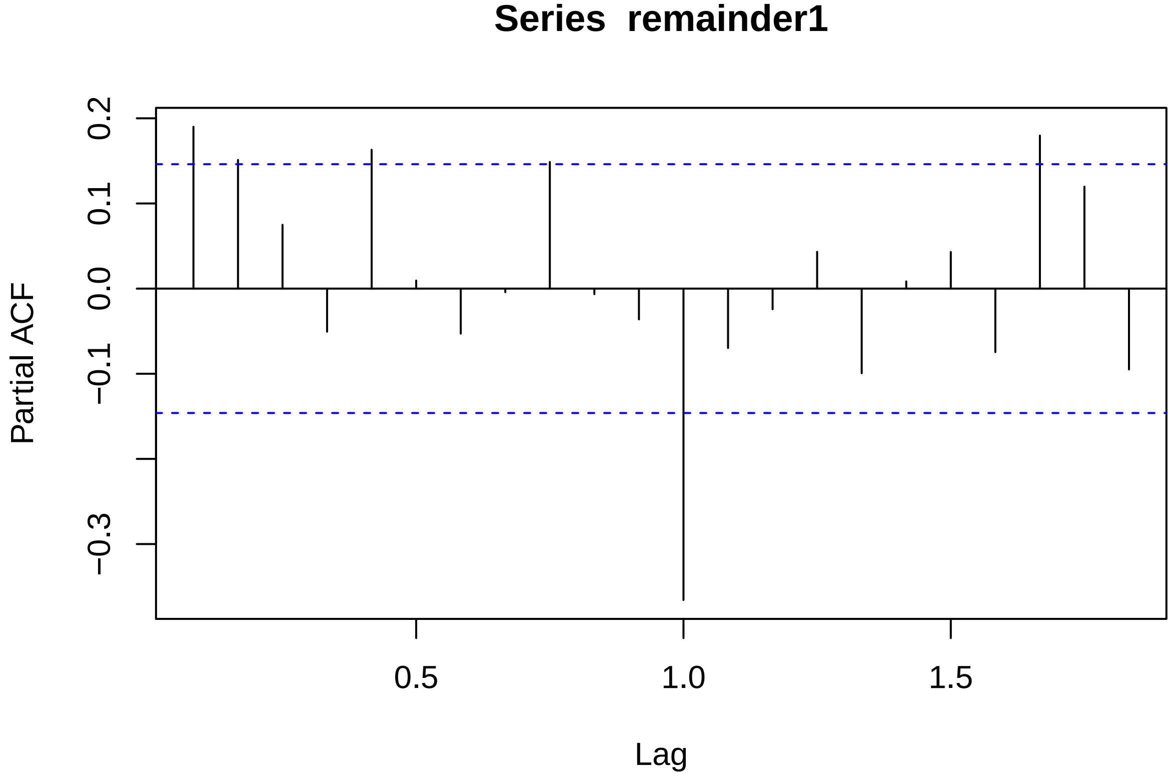


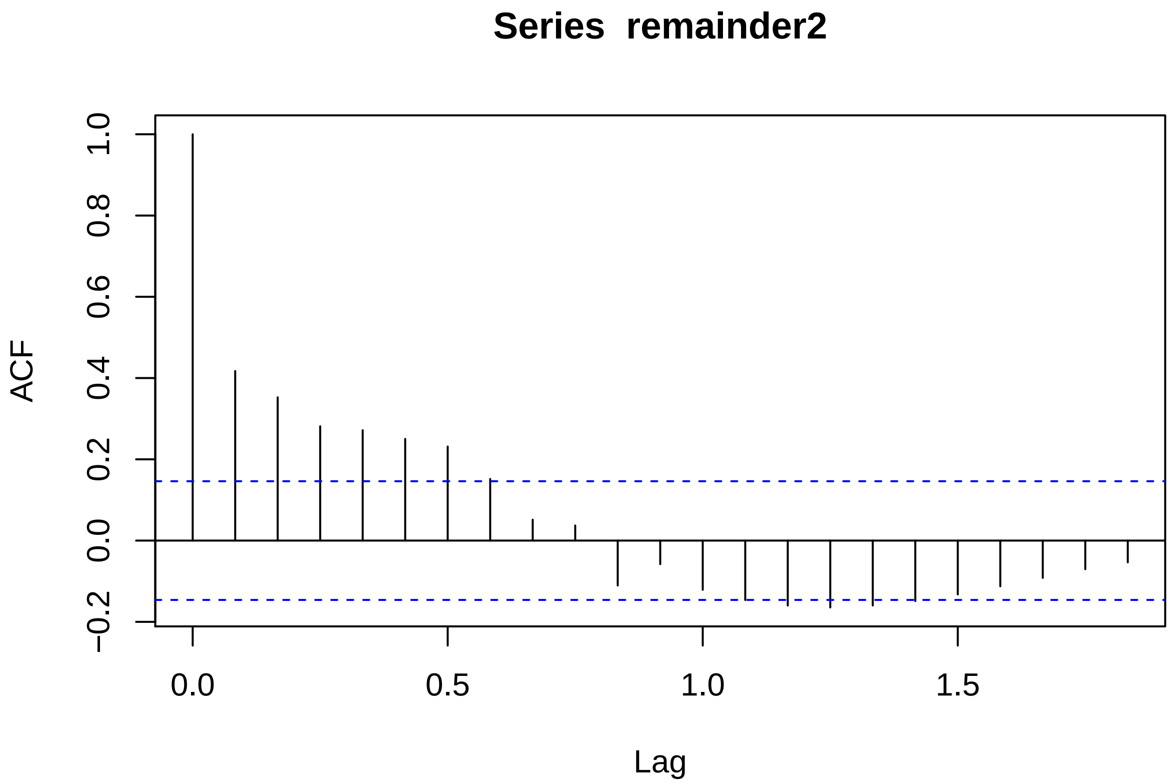
Comment:

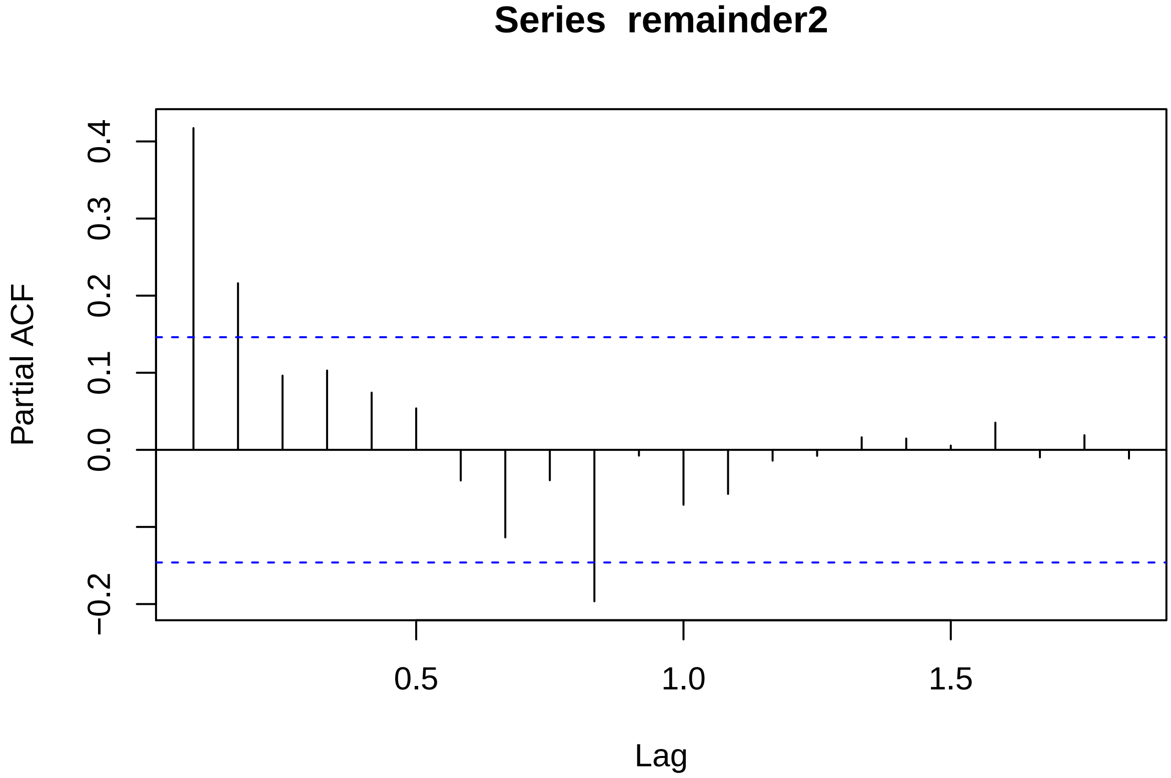
- By knowledge, the s.window argument is either a “span” for the seasonal bit or if periodic just computes the same average as before. While the t.window argument is the overall span of the trend smoothing.
- By taking value of t.window as 41, 51, 61 and 71, we observed that the trend is getting more and more smooth. Thus, we conclude that **as t increases, the trend will be more smooth**.
- Let’s look at the remainder part, when s.window is 3, the time series still have seasonal component, but while s.window taking value of 5 or 7, the irregular graphs do not have any patterns. Hence we can say that the **data do not have seasonal component for large s.window value** (here for s.window > 5, the time series does not have seasonal component).

(b) For one of set of parameter values used in part (a), look at the estimated irregular component. Does it look like white noise? Would you expect it to look like white noise?






Series remainder2
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.5
1.0
1.5
Lag


Series remainder2
Partial ACF
0.4
0.3
0.2
0.1
0.0
−0.2
0.5
1.0
1.5
Lag

Comment:

- From part(a), we know that the greater t and s we choose, the less trend and seasonal the data will be.
- Thus, we choose the stl with t.window=71 and s.window=7, which will give us greatest probability to be white noise. Then, by looking at its acf and pacf graph of the remainder part, we notice that it **looks like a white noise process**.
- However, if we choose stl with t.window=41 and s.window=3, we **will not** have the same result.
- To summary, we cannot expect it to look like white noise. For every case here, the most ideal case will give us the result we want, but that does not mean that all the cases will give us the same result. Therefore, **the irregular component may be white noise, but we would not expect it is for sure a white noise**.

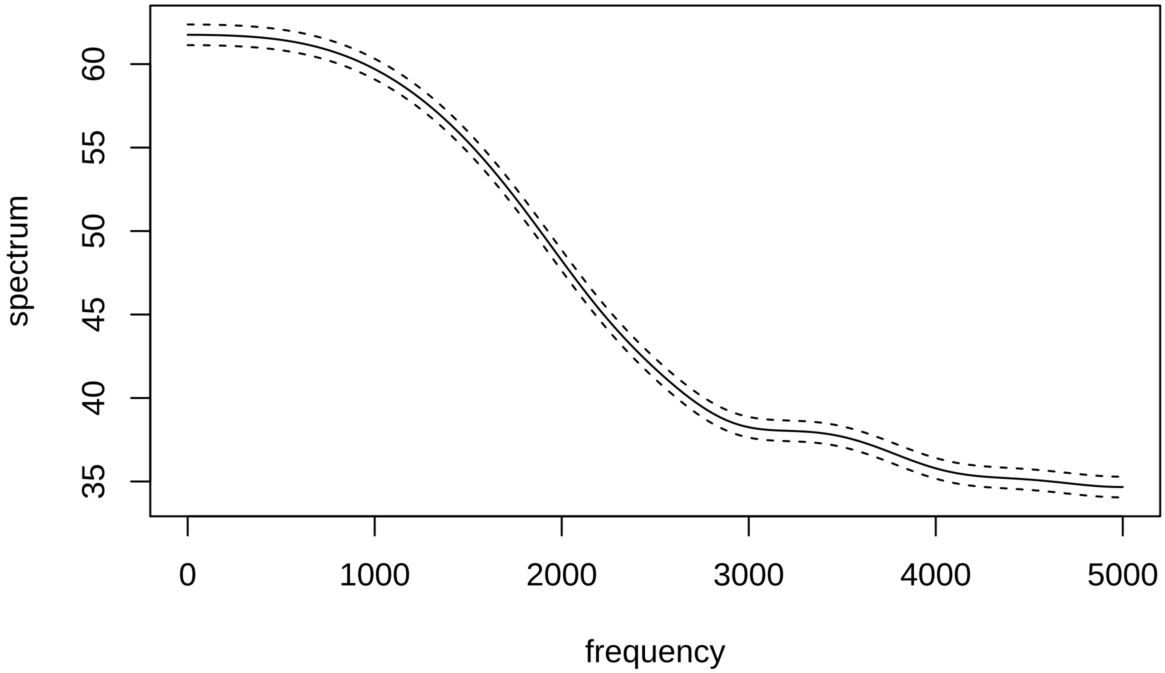
(c) The function stl estimates trend, seasonal, and irregular components. Other seasonal adjustment procedures can also estimate a calendar component in order to reflect variation due to the number of weekend days etc. For these data, do you think a calendar component would be useful?

Comment:

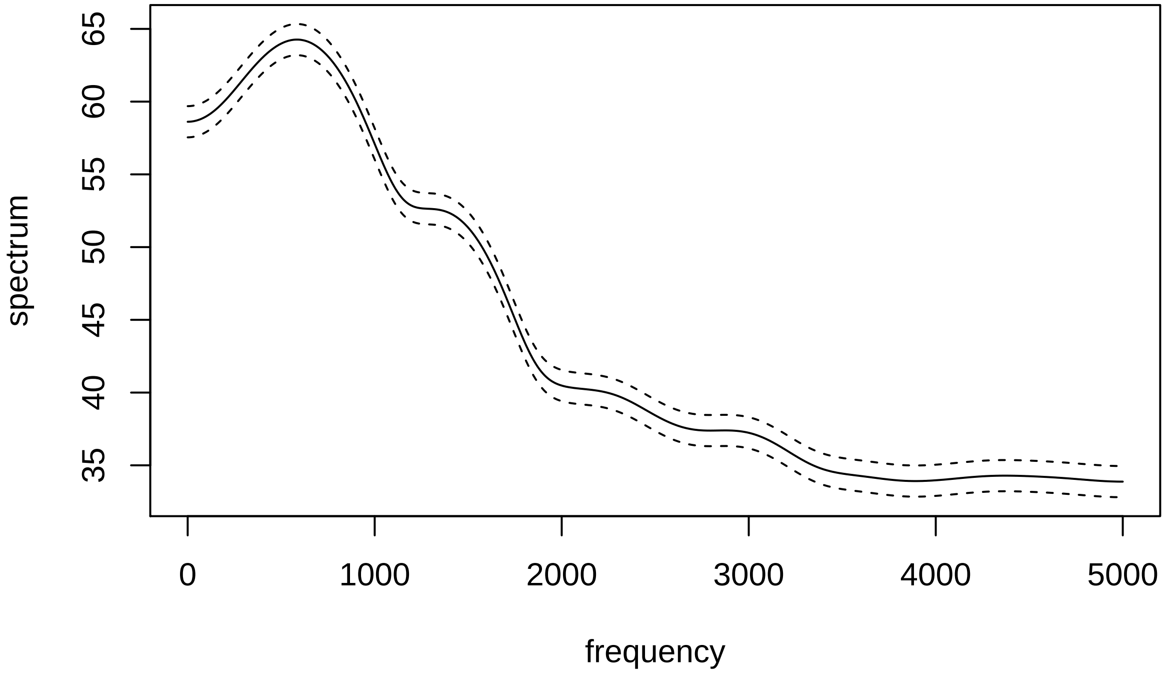
- By research, the calendar causes fluctuations in our everyday activities, which is likely to affect the short-term movements in the time series. The calendar effects are such as the working days effect, the leap year effect and the moving holiday effect.
- For these data, the calendar component seems to be useful, but it is really depending on the situation we are encountering. We cannot be 100% sure if it is useful or not in our case, but we know that it is somehow related to our procedures.

Q2

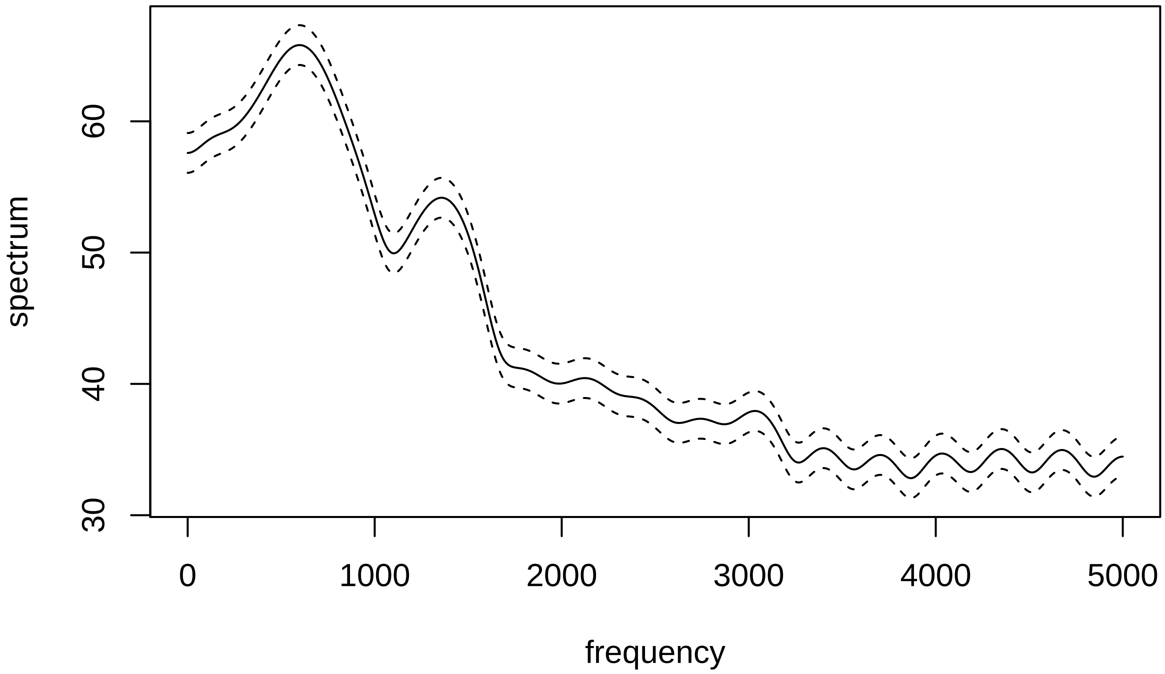
(a) Play around with different values of M to see how the estimates change with M.



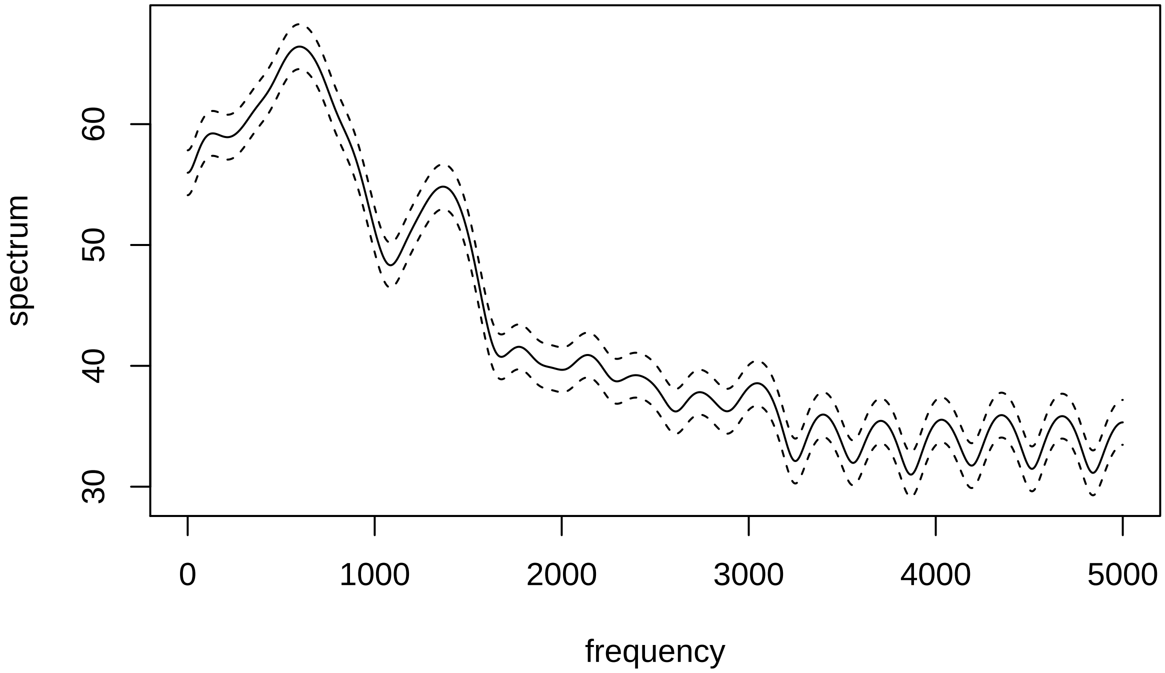
Parzen's window with M= 10



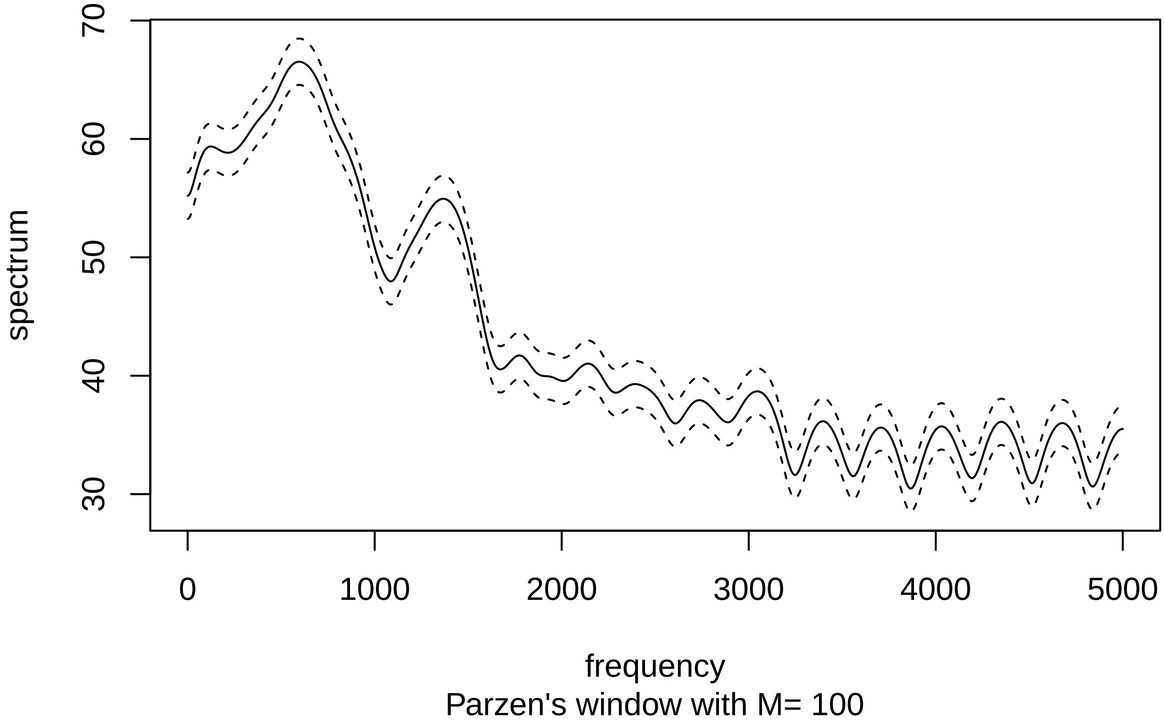
Parzen's window with M= 30



Parzen's window with M= 60



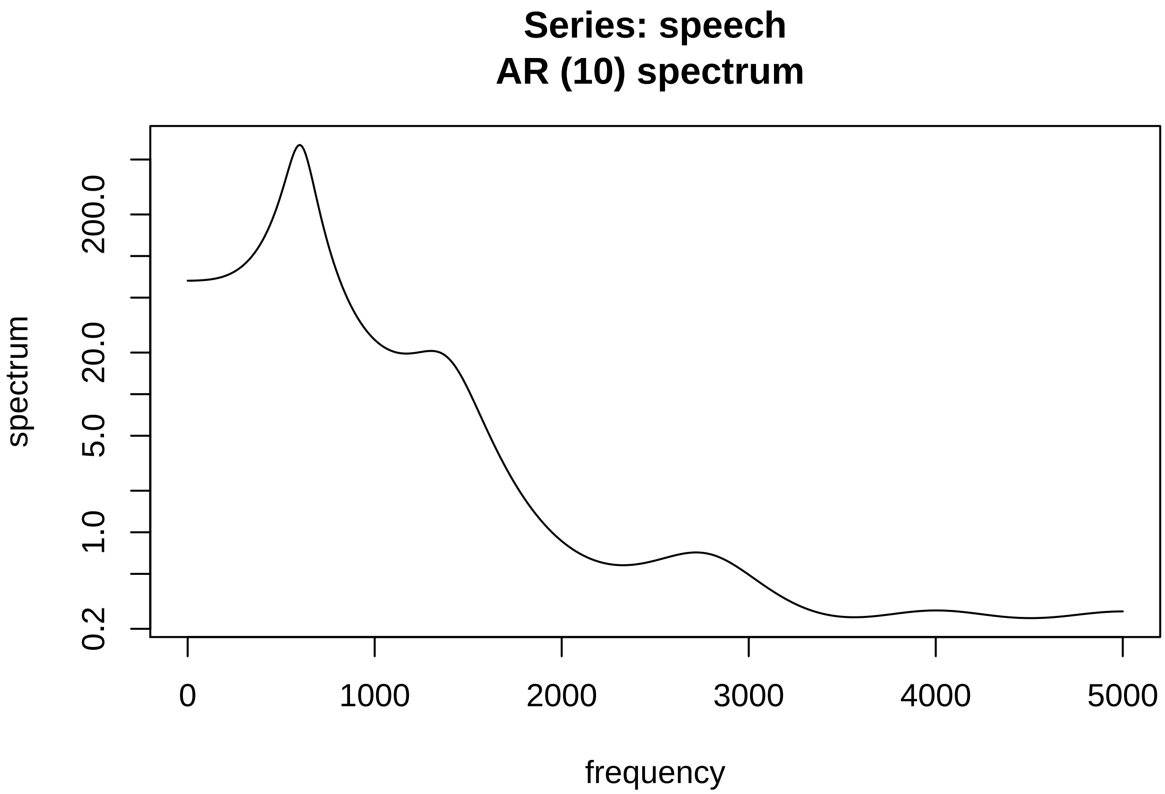
Parzen's window with M= 90

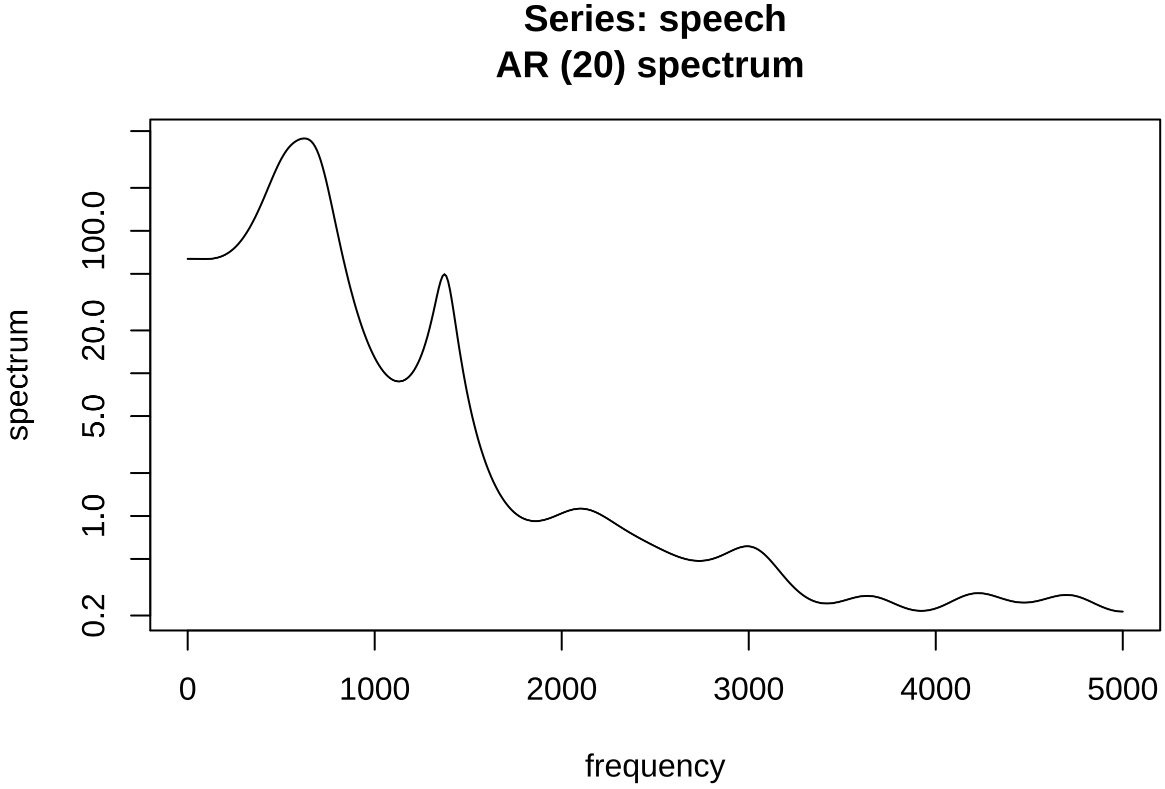


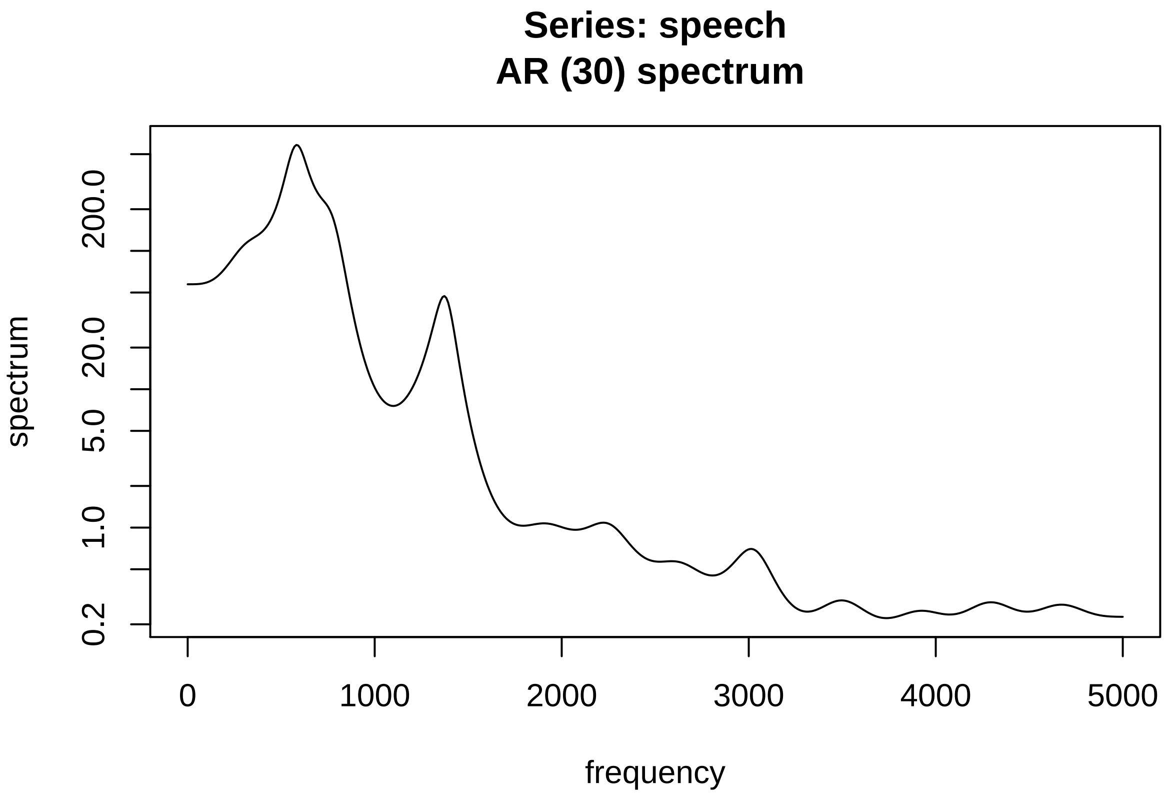
Comment:

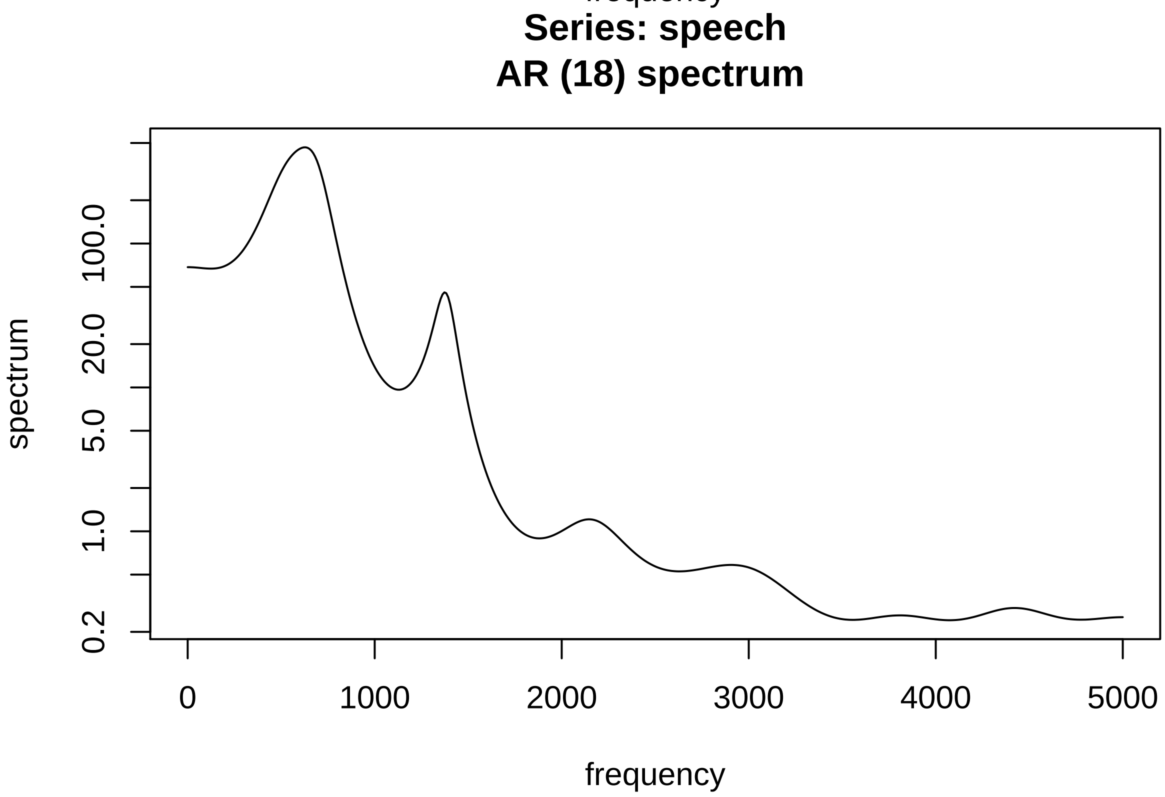
- By above graphs, we notice that as M increases, the plot tends to fluctuate more intense. And more peaks are shown on the plot.
- When M is small (ie. M=10), there is no obvious peaks since the curve is too smooth.
- For M large enough, (we tested M = 60, 90, 100), there are two obvious peaks value one is 700 and the other is 1200.
- When maxlag=100, we observed the least smooth plot.

(b) Compare these to the estimates this estimate with the estimate obtained in part (a).









Comment:

By comparing the graphs of (a) and (b), we observe that both (a) and (b) have **similar graphs**. The estimates obtained here have **alike trend and peaks** as the estimates we observed in part (a).

(c)

Comment:

The frequencies as **1200** and **700** seem to be most dominant. (those two are obvious peaks of the plots)

Q3

(a)

```
## Warning: package 'tseries' was built under R version 3.3.2
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts(barrick)
## Dickey-Fuller = -2.4496, Lag order = 17, p-value = 0.3879
## alternative hypothesis: stationary
```

```
## [1] -24438.09
```

```
## [1] -24450.7
```

```
##
##  Box-Ljung test
##
## data:  m1$residuals
## X-squared = 32.698, df = 10, p-value = 0.0003061
```

```
##
##  Box-Ljung test
##
## data:  m2$residuals
## X-squared = 16.089, df = 10, p-value = 0.09712
```

Comment:

- The null hypothesis here is H0: the Time Series is non-stationary. The p-value is 0.3879, which is greater than 0.05. Thus, we conclude that we are unable to reject the null hypothesis. Clearly, the time series is **non-stationary**, so it needs to be differenced to make it stationary.
- For ARIMA(p,d,q), we take d=1 in order to take difference of the time series once. To choose between ARIMA(0,1,1) and ARIMA(0,1,2), we apply ljung-box test to check White Noise. The result has p-value = 0.0003061 (<0.05) for mod1 and 0.09712 (>0.05) for mod2. Thus, we reject H0 for mod1 and fail to reject H0 for mod2. That means that residual of mod2 is **white noise**, while residual for mod1 is **not**. Therefore, we **prefer ARIMA(0,1,2)**.

(b) Using the residuals from your preferred model from part (a), fit ARCH(m) models for m = 1, 2, 3, 4, 5. Which model seems to be the best?

Comment:

- By calculation, we obtain that AIC for ARCH(1) is -4.503860, AIC for ARCH(2) is -4.533235, AIC for ARCH(3) is -4.562443, AIC for ARCH(4) is -4.576758 and AIC for ARCH(5) is -4.588344. We prefer the model with smallest AIC. Thus, **ARCH(5)** seems the best here.

(c) Repeat part (b), using GARCH(1, s) models for s = 1, 2, 3, 4, 5. Are any of these models an improvement over the best ARCH model from part (b)?

Comment:

- By calculation, we obtain that AIC for GARCH(1,1) is -4.664868, AIC for GARCH(1,2) is -4.665924, AIC for GARCH(1,3) is -4.665919, AIC for GARCH(1,4) is -4.665603 and AIC for GARCH(1,5) is -4.665689. All the GARCH models have smaller AIC than ARCH(5), which has aic value as -4.588344. Hence, we say that **all GARCH models are improvement over the best ARCH model from part (b)**.

```
# ---------> complete and run the following code for this assignment  <-------
#
# R code for STA457 assignment 4
# copyright by Christina (Qi) Deng
# date: March 27, 2017
#

#1(a)
fatal <- scan("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 4/fatalities.txt")
fatal <- ts(fatal,start=c(1960,1),end=c(1974,12),freq=12)
r <- stl(fatal,s.window = "periodic", robust = T, t.window = 41)
plot(r, main="stl with s='periodic' & t=41")
r <- stl(fatal,s.window = 3, robust = T, t.window = 51)
plot(r, main="stl with s=3 & t=51")
r <- stl(fatal,s.window = 5, robust = T, t.window = 61)
plot(r,main="stl with s=5 & t=61")
r <- stl(fatal,s.window = 7, robust = T, t.window = 71)
plot(r,main="stl with s=7 & t=71")

#(b)
remainder1=stl(fatal,s.window = 7,t.window = 71)$time.series[,"remainder"]
acf(remainder1)
pacf(remainder1)
remainder2=stl(fatal,s.window = 3,t.window = 41)$time.series[,"remainder"]
acf(remainder2)
pacf(remainder2)

#2(a)
spec.parzen <- function(x,maxlag,nfreq,plot=T) {
       f <- 1
       if (is.ts(x)) f <- frequency(x)
       x <- as.vector(x[!is.na(x)])
       n <- length(x)
       x <- x - mean(x)
       if (missing(maxlag)) maxlag <- n - 1
       if (maxlag >= n) maxlag <- n - 1
       if (maxlag <= 1) maxlag <- 2
       k <- ceiling(log(n+maxlag)/log(2))
       nn <- 2^k
       m <- nn/2
       if (missing(nfreq)) nfreq <- m
       if (nfreq < m) nfreq <- m
       if (nfreq > m) {
               k <- 1 + ceiling(log(nfreq)/log(2))
               nn <- 2^k
               nfreq <- nn/2
               }
       x <- c(x,rep(0,nn-n))
       if (missing(maxlag)) maxlag <- n - 1
       if (maxlag >= n) maxlag <- n - 1
       if (maxlag <= 1) maxlag <- 2
       k1 <- floor(maxlag/2)
       k2 <- maxlag
       k <- c(0:k1)
       window <- 1 - 6*(k/maxlag)^2 + 6*(k/maxlag)^3
       k <- c((k1+1):k2)
       window <- c(window,2*(1-k/maxlag)^3)
       if (maxlag < nn) window <- c(window,rep(0,nn-maxlag-1))
       spec <- Mod(fft(x))^2/n
       ac <- Re(fft(spec,inv=T))/nn
       ac <- ac*window
       spec <- 2*Re(fft(ac))-ac[1]
       spec <- 10*log10(spec[1:(nfreq+1)])
       freq <- f*c(0:nfreq)/nn
       se <- 4.342945*sqrt(0.539*maxlag/n)
       if (plot) {
         lims <- c(min(spec - 1.96*se),max(spec + 1.96*se))
         plot(freq,spec,type="l",xlab="frequency",ylab="spectrum",ylim=lims)
         lines(freq,spec + 1.96*se,lty=2)
         lines(freq,spec - 1.96*se,lty=2)
         ttl <- paste("Parzen's window with M=",maxlag)
         title(sub=ttl)
         }
       std.err <- se
       r <- list(frequency=freq,spec=spec,M=maxlag,std.err=std.err)
       r
       }
speech <- ts(scan("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 4/speech.txt"),frequency=
r <- spec.parzen(speech,maxlag=10,plot=T)
r <- spec.parzen(speech,maxlag=30,plot=T)
r <- spec.parzen(speech,maxlag=60,plot=T)
r <- spec.parzen(speech,maxlag=90,plot=T)
r <- spec.parzen(speech,maxlag=100,plot=T)

#(b)
r <- spec.ar(speech,order=10,method="burg")
r <- spec.ar(speech,order=20,method="burg")
r <- spec.ar(speech,order=30,method="burg")
r <- spec.ar(speech,method="yw")

#3(a)
library(tseries)
barrick <- scan("/Users/christinadeng/Desktop/Winter 2017/STA 457/Assignment 4/barrick.txt")
adf.test(ts(barrick))
m1<-arima(barrick,c(0,1,1))
m1$aic
m2<-arima(barrick,c(0,1,2))
m2$aic
Box.test(m1$residuals,type="Ljung", lag=10)
Box.test(m2$residuals,type="Ljung", lag=10)

#(b)
library('fGarch')
R=m2$residuals
F1=garchFit(R~garch(1,0),data=R,trace=F)
summary(F1)
```

```
F2=garchFit(R~garch(2,0),data=R,trace=F)
summary(F2)
F3=garchFit(R~garch(3,0),data=R,trace=F)
summary(F3)
F4=garchFit(R~garch(4,0),data=R,trace=F)
summary(F4)
F5=garchFit(R~garch(5,0),data=R,trace=F)
summary(F5)
#(c)
f1=garchFit(R~garch(1,1),data=R,trace=F)
summary(f1)
f2=garchFit(R~garch(1,2),data=R,trace=F)
summary(f2)
f3=garchFit(R~garch(1,3),data=R,trace=F)
summary(f3)
f4=garchFit(R~garch(1,4),data=R,trace=F)
summary(f4)
f5=garchFit(R~garch(1,5),data=R,trace=F)
summary(f5)
```