STA 137 Midterm 2: Take Home

Janice Luong Section A02

Febuary 23, 2016

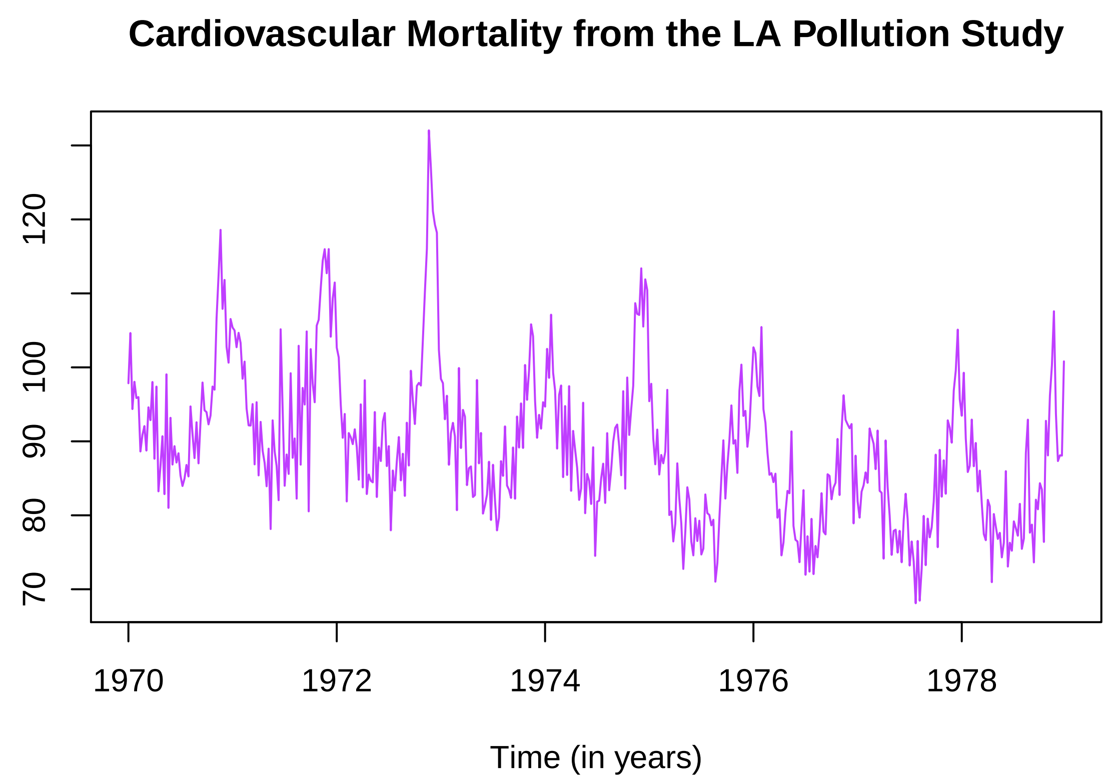
```r
#get the data
require(astsa)
```

```
## Loading required package: astsa
```

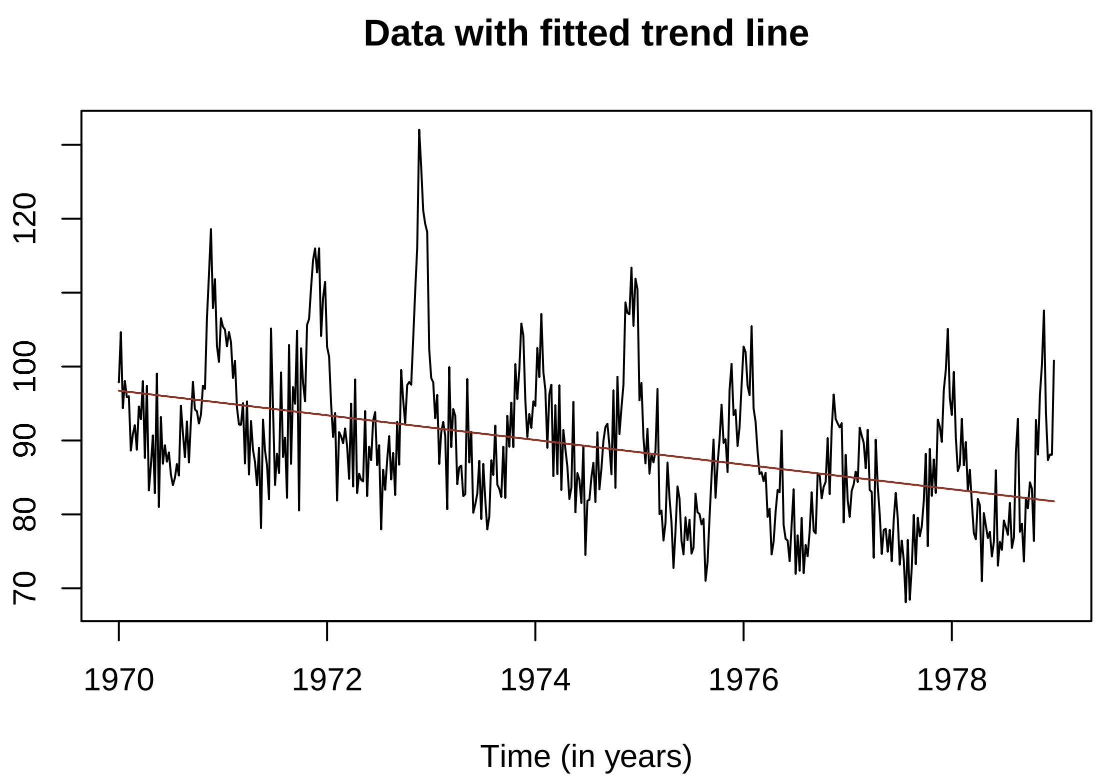
```r
data("cmort")
cmort.part = window(cmort, start = c(1970, 1), end = c(1978, 52))
```

Question 1

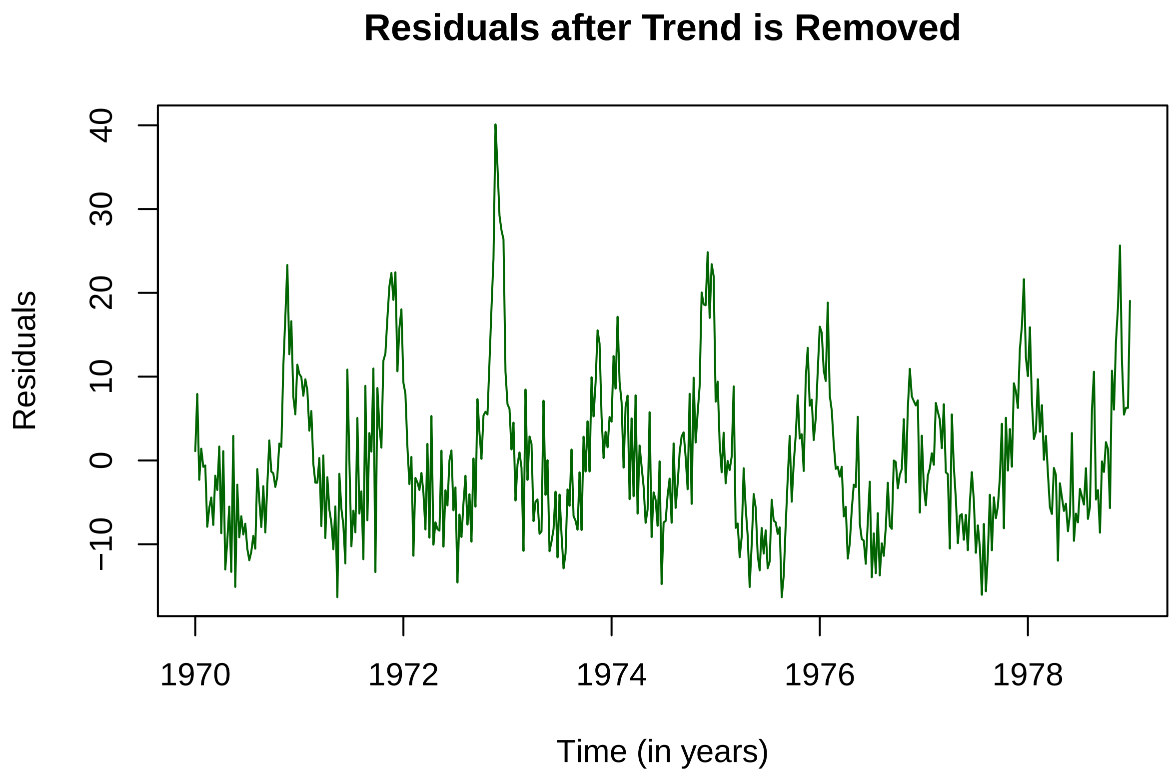
```r
dataCmort = as.vector(cmort.part)
timeCmort = as.vector(time(cmort.part))

plot(timeCmort, dataCmort, type = "l",
     main = "Cardiovascular Mortality from the LA Pollution Study",
     xlab = "Time (in years)", ylab = "", col = "darkorchid1")
```



```r
#remove trend
trend.fit = lm(dataCmort~timeCmort)

plot(timeCmort, dataCmort, type = "l", main = "Data with fitted trend line",
     xlab = "Time (in years)", ylab = "")
lines(timeCmort, fitted(trend.fit), col = "tomato4")
```




```r
# plot residuals after trend is removed
cmortResid = residuals(trend.fit)
plot(timeCmort, cmortResid, type="l", main = "Residuals after Trend is Removed",
     xlab = "Time (in years)", ylab = "Residuals", col = "darkgreen")
```



After removing the trend, I notice a seaonality every year, something it spikes up (increases). The increase beings at every year, it is the lowest in the middle of the year and then it slowly rises again near the end of year. Since the study takes an average weekly of the cardiovascular mortality in Los Angeles County, the d = 52.

Question 2

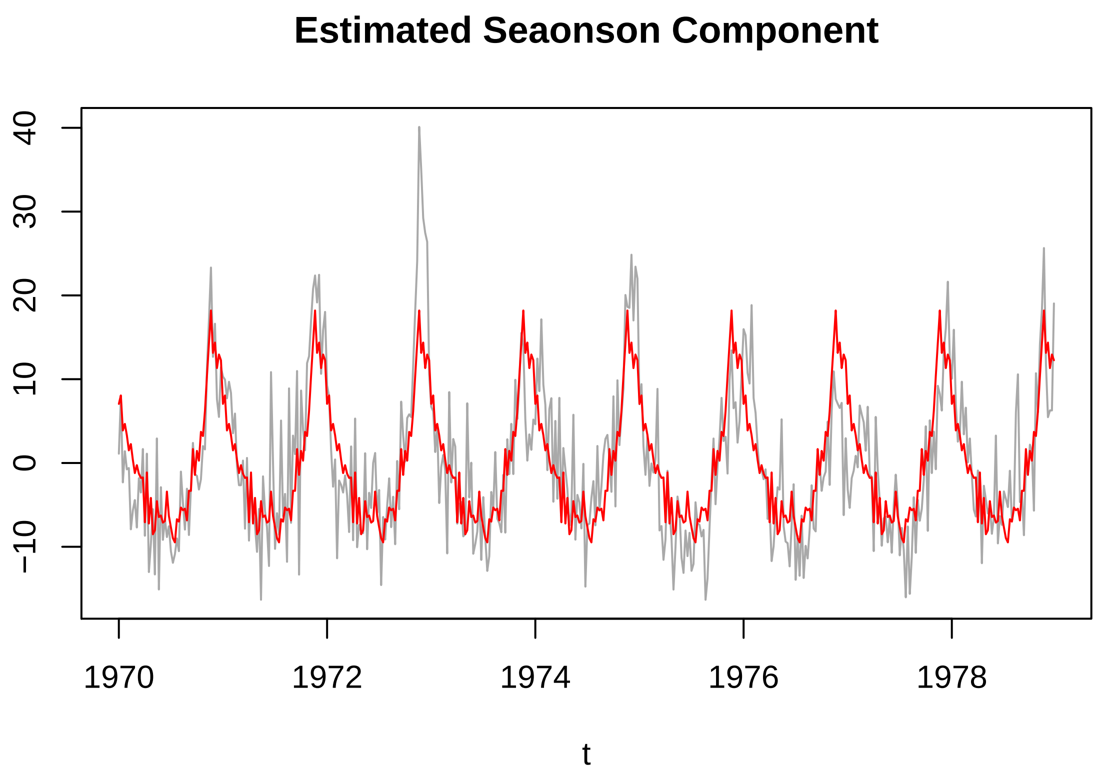
```
#remove the seasonal component by using a sum of harmonics

#use t that is in the interval [0,1]
intervalT = 1:length(cmortResid)
n = length(timeCmort)
intervalT2 = (intervalT) / n

n.harm = 26 #set to [d/2]
d = 52 #number of time pionts in each season
harm = matrix(nrow = length(intervalT2), ncol = 2*n.harm)
for(i in 1:n.harm){
  harm[,i*2-1] = sin(n/d * i *2*pi*intervalT2)
  harm[,i*2] = cos(n/d * i *2*pi*intervalT2)
}
colnames(harm)=
    paste0(c("sin", "cos"), rep(1:n.harm, each = 2))
```

```r
#fit on all of the sines and cosines
dat = data.frame(cmortResid, harm)
fit = lm(cmortResid~., data = dat)

# setup the full model and the model with only an intercept
full = lm(cmortResid~.,data = dat)
reduced = lm(cmortResid~1, data = dat)

#stepwise regression starting with the full model
#full with all sin and cos
fit.back = step(full, scope = formula(reduced), direction = "both")

#get back the original t so that we can plot over this range
t = as.vector(time(cmort.part))
```

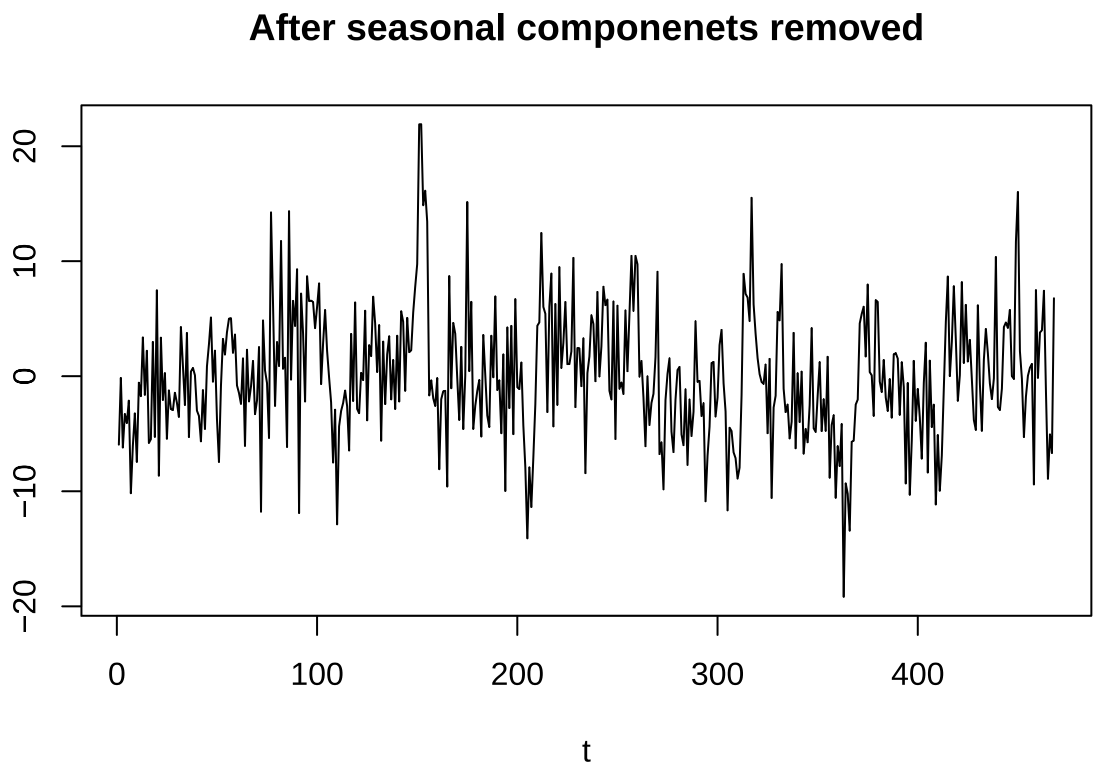
```r
summary(fit.back)
```

```
##
## Call:
## lm(formula = cmortResid ~ sin1 + cos1 + sin2 + cos2 + sin3 +
##     sin4 + cos5 + cos6 + sin7 + cos8 + cos10 + sin12 + sin20 +
##     cos21 + sin23 + sin24 + sin25, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.1734  -3.4404  -0.3471   3.6107  21.9147
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.663e-15  2.616e-01   0.000 1.000000
## sin1        -2.265e+00  3.699e-01  -6.124 2.00e-09 ***
## cos1         8.966e+00  3.699e-01  24.239  < 2e-16 ***
## sin2        -2.434e+00  3.699e-01  -6.579 1.32e-10 ***
## cos2         1.998e+00  3.699e-01   5.401 1.07e-07 ***
## sin3        -1.293e+00  3.699e-01  -3.496 0.000520 ***
## sin4        -1.277e+00  3.699e-01  -3.453 0.000606 ***
## cos5        -5.227e-01  3.699e-01  -1.413 0.158338
## cos6        -5.161e-01  3.699e-01  -1.395 0.163679
## sin7         7.146e-01  3.699e-01   1.932 0.054021 .
## cos8         6.960e-01  3.699e-01   1.882 0.060548 .
## cos10        7.877e-01  3.699e-01   2.130 0.033754 *
## sin12       -7.207e-01  3.699e-01  -1.948 0.051997 .
## sin20       -7.841e-01  3.699e-01  -2.120 0.034574 *
## cos21        8.503e-01  3.699e-01   2.299 0.021981 *
## sin23       -8.357e-01  3.699e-01  -2.259 0.024344 *
## sin24       -5.424e-01  3.699e-01  -1.466 0.143263
## sin25        7.023e-01  3.699e-01   1.898 0.058273 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.659 on 450 degrees of freedom
## Multiple R-squared:  0.6287, Adjusted R-squared:  0.6146
## F-statistic: 44.81 on 17 and 450 DF,  p-value: < 2.2e-16
```

```r
#plot the estimated seasonal components
plot(t,cmortResid, type = "l", main = "Estimated Seaonson Component",
     col = "darkgrey", ylab="")
#fitted seasonal component
lines(t, fitted(fit.back), col = "red")
```



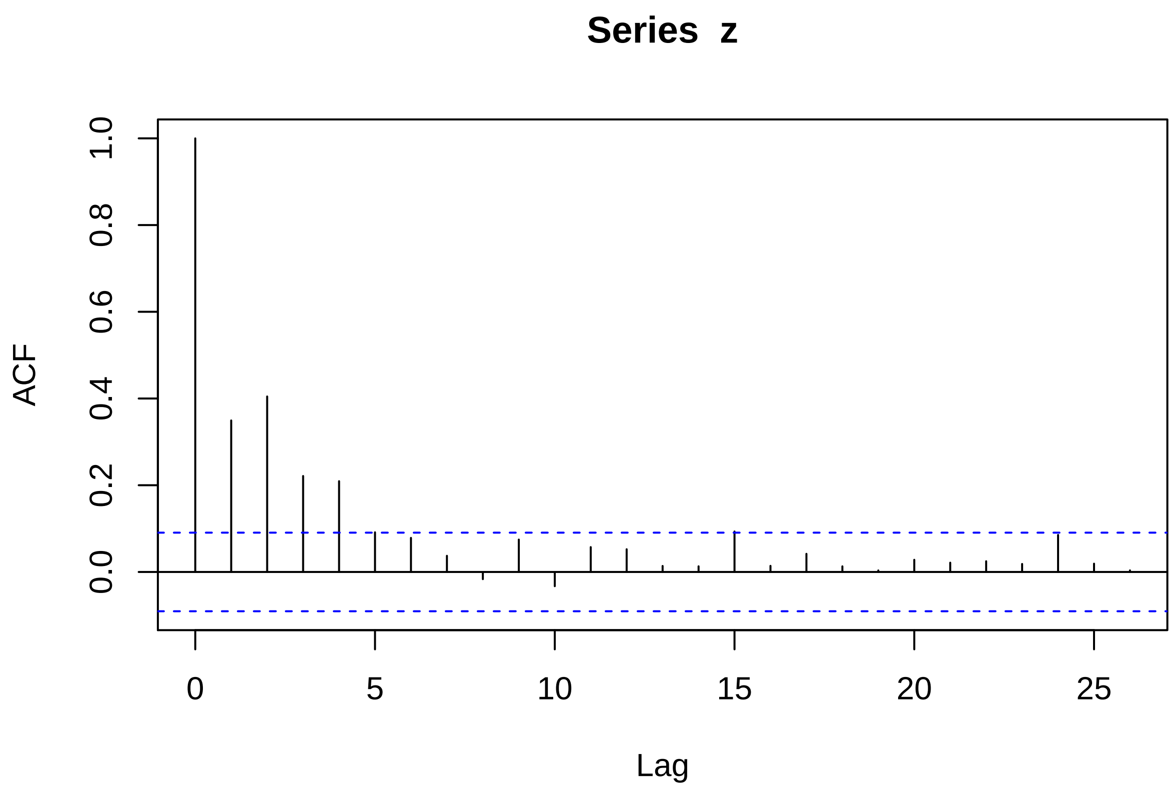

```r
#plot the residuals after seasonal component is removed
ts.plot(residuals(fit.back),
        main = "After seasonal componenets removed", ylab = "", xlab = "t")
```



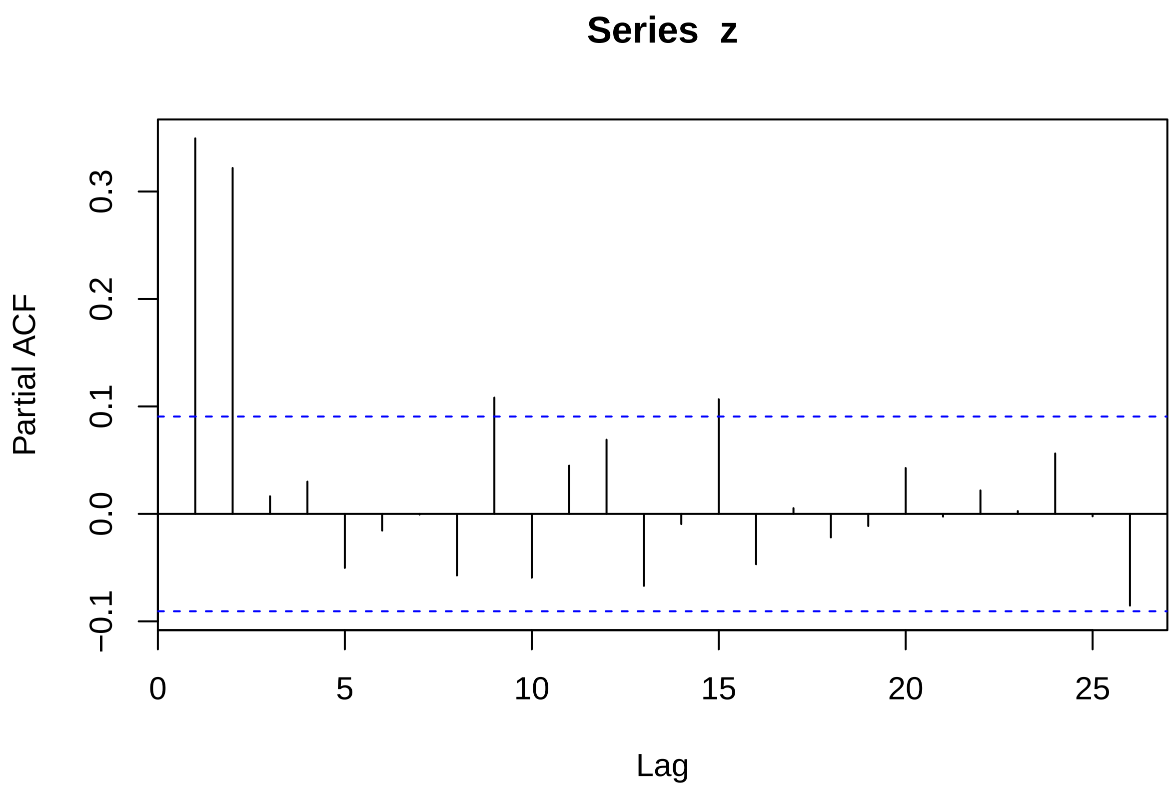
The residuals look mostly stationary because it is centered around mean 0.

Question 3

```
#plot the acf and pacf of the residuals
z = residuals(fit.back)
acf(z)
```




```
pacf(z)
```



Since the ACF trails off to 0 and the PACF drops off (close) to zero after lag p, where p = 2, The time series model I believe that is appropriate for these residuals is AR(2).

Question 4

```
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 6.2
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:astsa':
##
##     gas
```

```r
ar.fit = auto.arima(residuals(fit.back), stepwise = FALSE)
ar.fit
```

```
## Series: residuals(fit.back)
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1     ar2
##       0.2360  0.3228
## s.e.  0.0439  0.0439
##
## sigma^2 estimated as 24.2:  log likelihood=-1409.87
## AIC=2825.75   AICc=2825.8   BIC=2838.19
```

Using the function auto.arima function, with parameters stepwise = FALSE, the best fit model is AR(2).

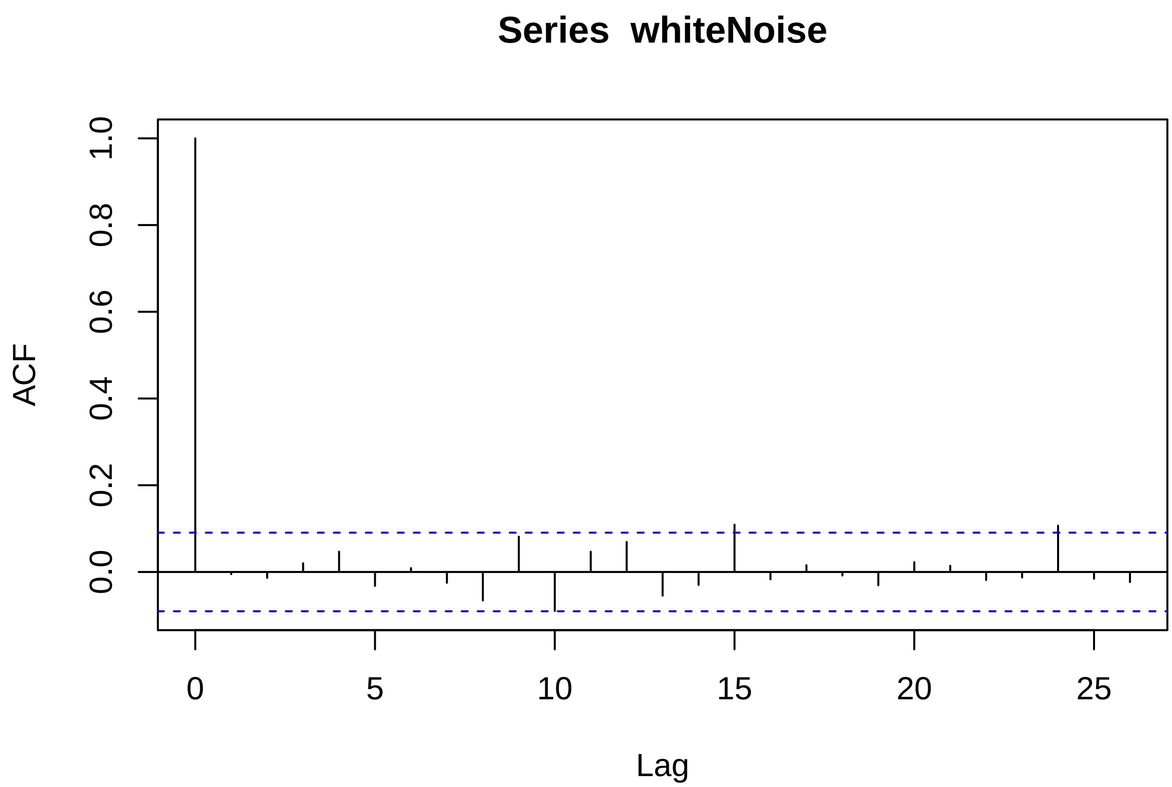
```r
auto.arima(residuals(fit.back), stepwise = TRUE)
```

```
## Series: residuals(fit.back)
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1     ar2
##       0.2360  0.3228
## s.e.  0.0439  0.0439
##
## sigma^2 estimated as 24.2:  log likelihood=-1409.87
## AIC=2825.75   AICc=2825.8   BIC=2838.19
```

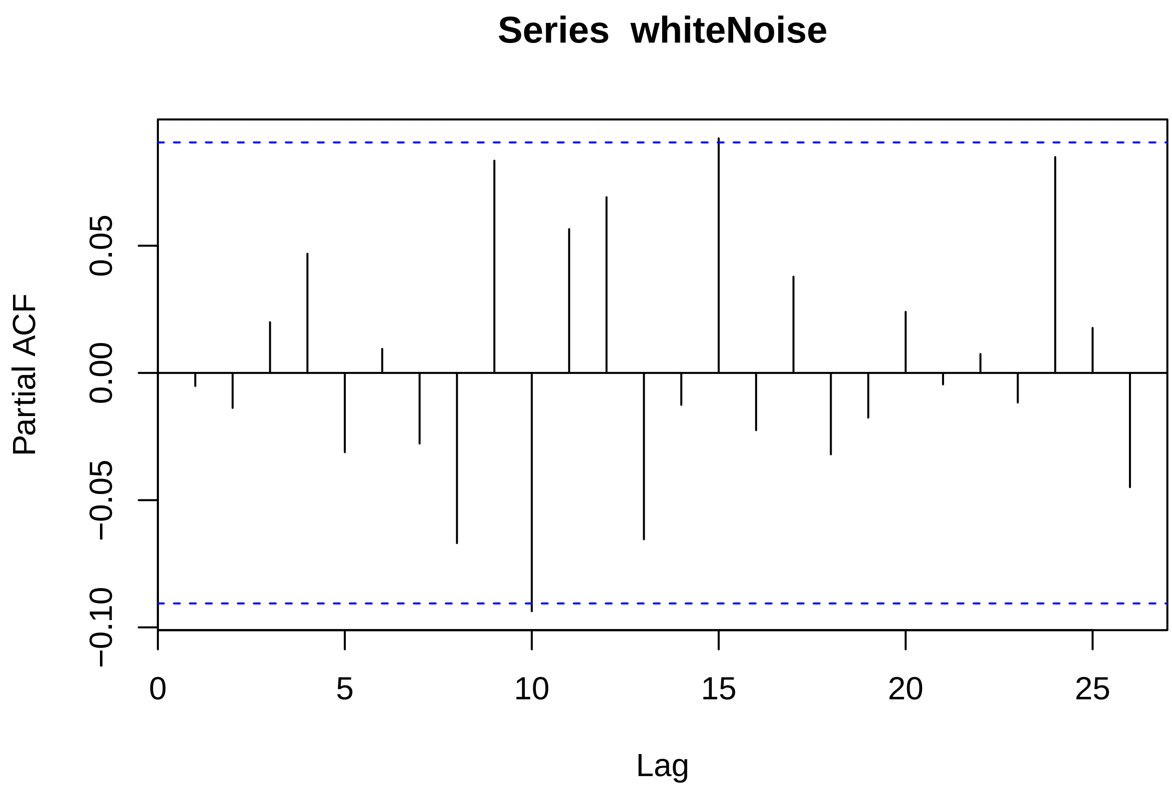
Using the function auto.arima function, with parameters stepwise = TRUE, the best fit model is AR(2).

Question 5

```r
#white noise
whiteNoise = ar.fit$residuals
#plot the acf and pacf of the remaining noise
acf(whiteNoise, na.action = na.pass)
```




```
pacf(whiteNoise, na.action = na.pass)
```



Based onf ACF and PACF functions, the remaining residuals are white noise because the ACF drops off after lag 1 and the PACF have no level of signifigance.

Question 6

```
#fit ARMA(3,1) model
armaFit = arima(z, order = c(2, 0, 0), include.mean = FALSE)
Box.test(armaFit$residuals, type = "Ljung-Box", lag = min(2*d, floor(n/5)))
```

```
##
##  Box-Ljung test
##
## data:  armaFit$residuals
## X-squared = 97.202, df = 93, p-value = 0.3623
```

H_0: residuals (denoted as $\{Y_t\}_{t=1}^{n}$) are independent up to some time lag h (no dependence structure remaining)

H_a: residuals are dependent (dependence structure remaining)

We get: $p - value = 0.3623$ and we have level of significance at $\alpha = 0.05$

Since the $p - value$ is greater than α, we fail to reject H_0. This means that the residuals, $\{Y_t\}_{t=1}^{n}$, are independent up to some time lag h (no dependence structure remaining).