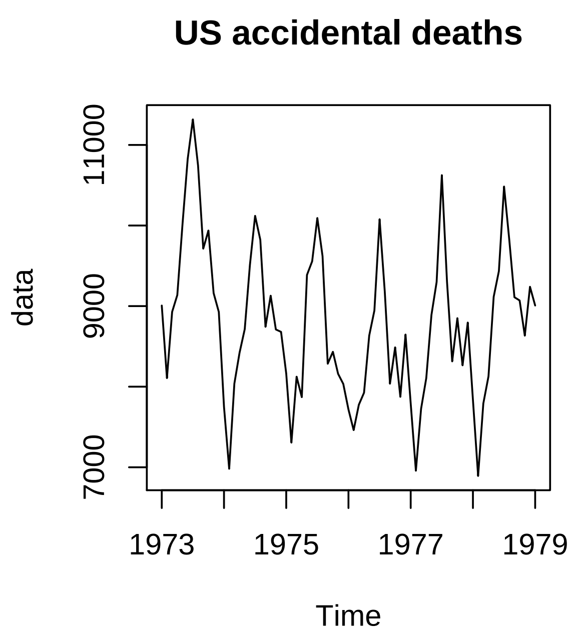
Accidental death data analysis

Some preliminary analysis gives US accidental deaths data clearly shows seasonality

```
rm(list=ls(all=TRUE))
source("TS-library.R")
data = scan("deaths.txt");
data = ts(data, start=1973, end=1979, freq=12)
n = length(data)
par(mfrow=c(1,2))
plot.ts(data);
title("US accidental deaths")
acf2(data, 35);
title("SACF")
```



Remove seasonality by Harmonic regression

To remove seasonal patterns, we decided to apply Harmonic regression as in the below.

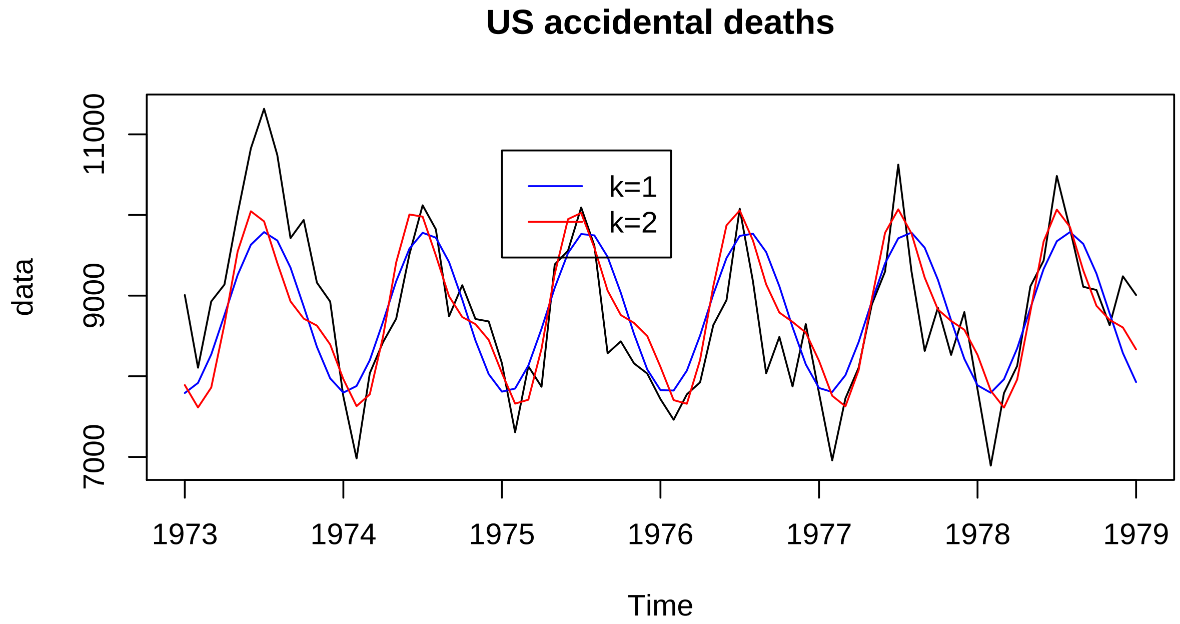
```
t = 1:n;
m1 = 6;
m2 = 12;
costerm1 = cos(m1*2*pi/n*t);
sinterm1 = sin(m1*2*pi/n*t);
costerm2 = cos(m2*2*pi/n*t);
sinterm2 = sin(m2*2*pi/n*t);
out.lm1 = lm(data ~ 1 + costerm1 + sinterm1)
summary(out.lm1)
```

```
##
## Call:
## lm(formula = data ~ 1 + costerm1 + sinterm1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1503.17  -428.60   -62.62   352.88  1529.52
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8790.74      75.19 116.913  < 2e-16 ***
## costerm1     -862.95     106.34  -8.115 1.12e-11 ***
## sinterm1     -501.61     106.34  -4.717 1.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 642.4 on 70 degrees of freedom
## Multiple R-squared:  0.5573, Adjusted R-squared:  0.5446
## F-statistic: 44.06 on 2 and 70 DF,  p-value: 4.117e-13
```

```r
out.lm2 = lm(data ~ 1 + costerm1 + sinterm1 + costerm2 + sinterm2)
summary(out.lm2)
```

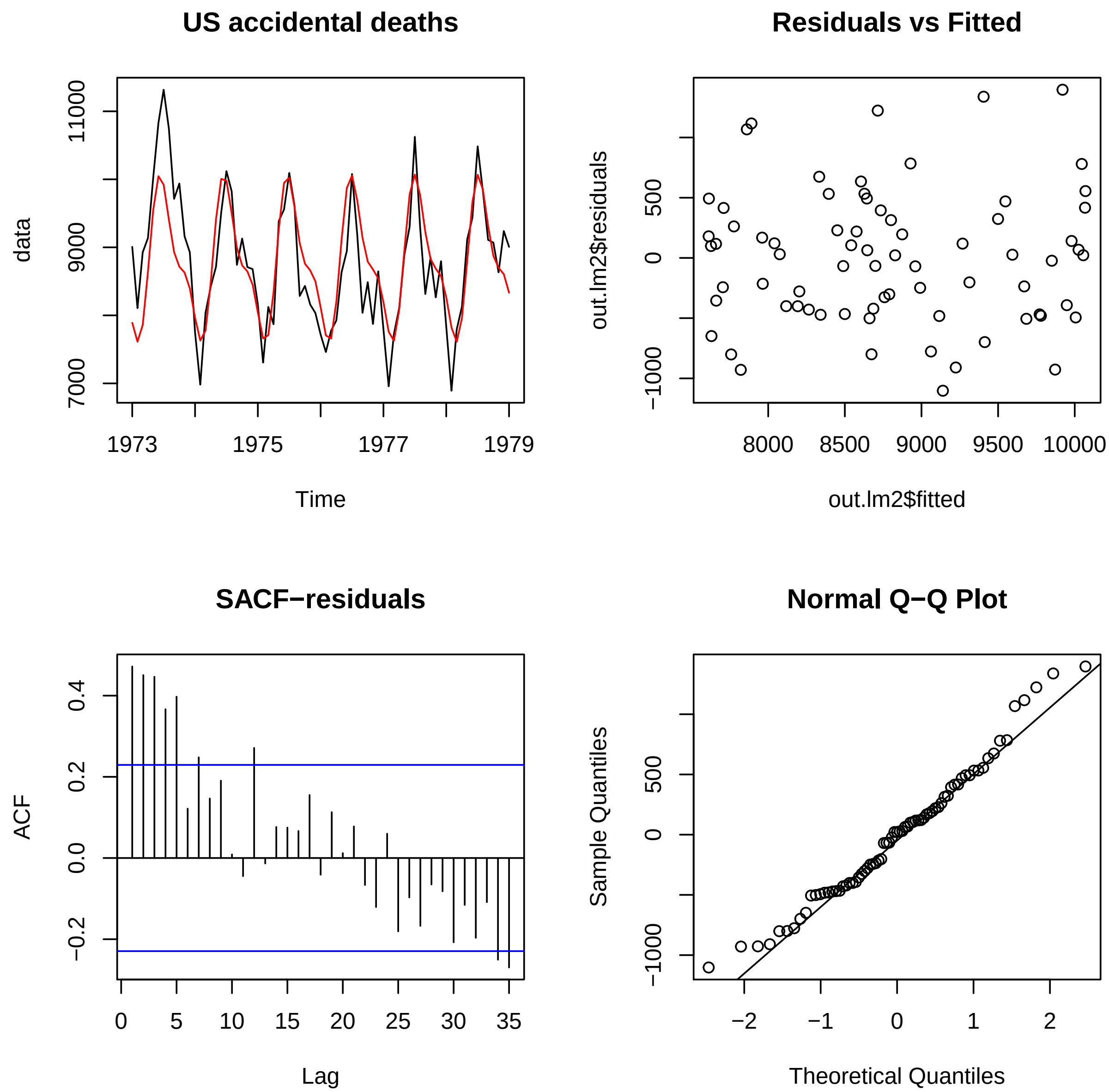
```
##
## Call:
## lm(formula = data ~ 1 + costerm1 + sinterm1 + costerm2 + sinterm2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1102.8  -421.2    22.2   323.4  1396.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8790.74      67.03 131.143  < 2e-16 ***
## costerm1     -862.95      94.80  -9.103 2.20e-13 ***
## sinterm1     -501.61      94.80  -5.291 1.40e-06 ***
## costerm2      405.11      94.80   4.273 6.14e-05 ***
## sinterm2     -127.69      94.80  -1.347    0.182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 572.7 on 68 degrees of freedom
## Multiple R-squared:  0.6582, Adjusted R-squared:  0.6381
## F-statistic: 32.74 on 4 and 68 DF,  p-value: 3.291e-15
```

```r
x = as.vector(time(data))
plot.ts(data);
title("US accidental deaths")
lines(x, out.lm1$fitted, col="blue")
lines(x,out.lm2$fitted, col="red")
legend(1975, 10800, lty=c(1,1), col=c("blue","red"), c("k=1", "k=2"))
```



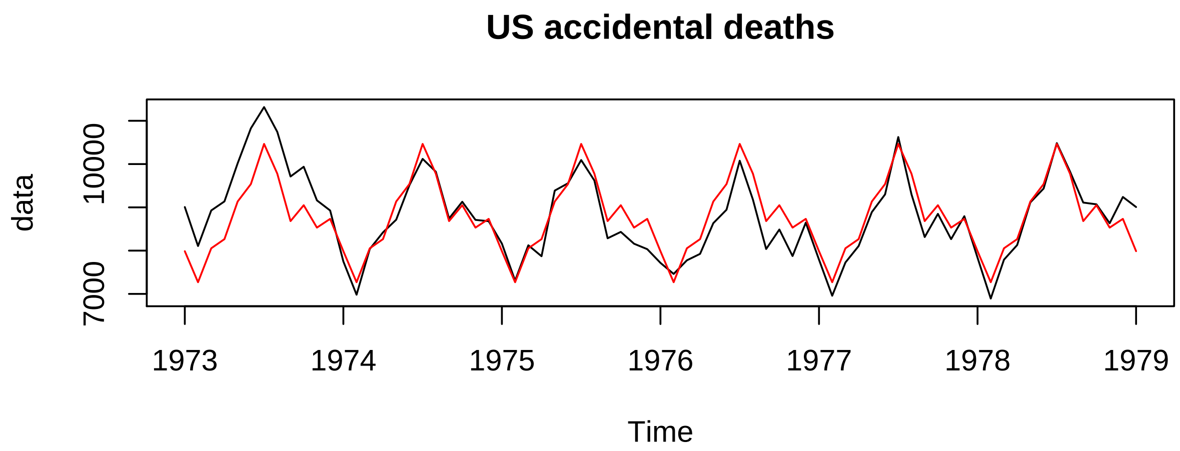
It seems that $k = 2$ is better than $k = 1$ in terms of fit. So we proceed with $k = 2$. Regression diagnostics gives the following

```
par(mfrow=c(2,2));
plot.ts(data);
title("US accidental deaths")
lines(x,out.lm2$fitted, col="red")
plot(out.lm2$fitted, out.lm2$residuals);
title("Residuals vs Fitted")
acf2(out.lm2$residuals)
title("SACF-residuals")
qqnorm(out.lm2$residuals);
qqline(out.lm2$residuals)
```



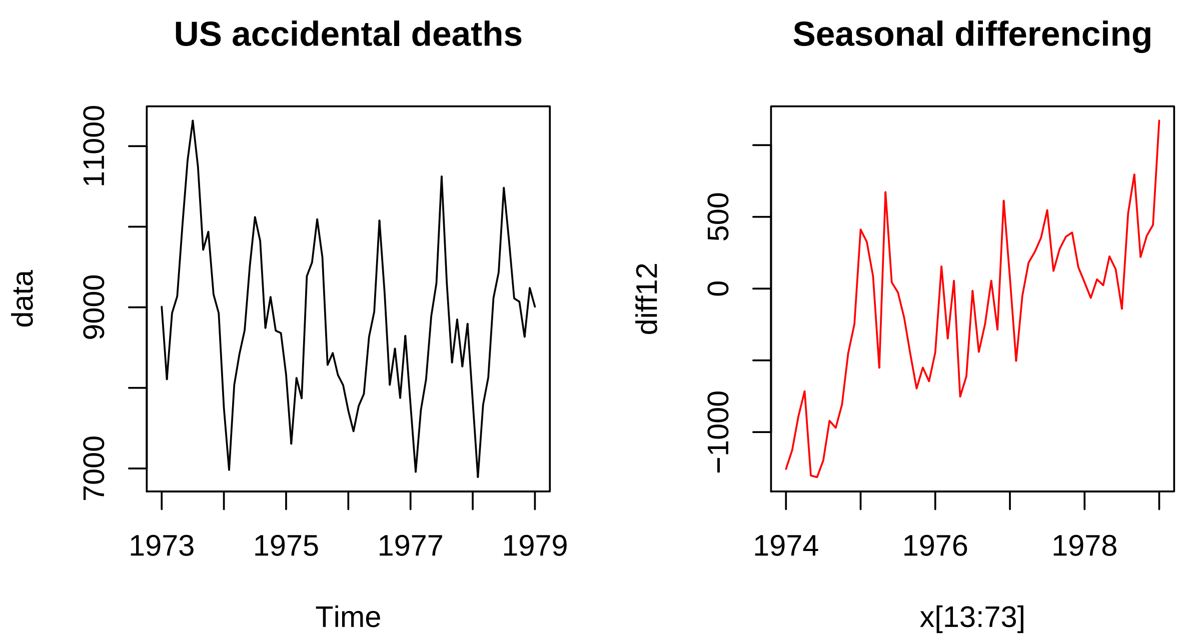
The second way to estimate seasonal component is to take seasonal average

```
# If we apply seasonal average
library(itsmr)
season.avg = season(data, d=12)
plot.ts(data);
title("US accidental deaths")
lines(x, season.avg + mean(data), col="red")
```



Another simple way is to apply seasonal differencing

```
####################
# lag-12 differencing
####################
diff12 = diff(data, lag=12);
par(mfrow=c(1,2))
plot.ts(data);
title("US accidental deaths")
plot(x[13:73], diff12, type="l", col="red")
title("Seasonal differencing")
```



If we apply classical Decomposition algorithm to remove both seasonality and trend, it gives the following

```r
classical
```

```
## function (data, d, order)
## {
##     n = length(data)
##     q = ifelse(d%%2, (d - 1)/2, d/2)
##     x = c(rep(data[1], q), data, rep(data[n], q))
##     if (d%%2 == 0) {
##         ff = c(0.5, rep(1, 2 * q - 1), 0.5)/d
##     }
##     if (d%%2 == 1) {
##         ff = rep(1, 2 * q + 1)/d
##     }
##     xx = filter(x, ff, method = c("convolution"))
##     mhat = na.omit(xx)
##     mhat = as.numeric(mhat)
##     z = data - as.numeric(mhat)
##     st = season(z, d)
##     mnew = trend(data - st, order)
##     fit = mnew + st
##     resi = data - fit
##     return(list(fit = fit, st = st, m = mnew, resi = resi, m1 = mhat))
## }
```

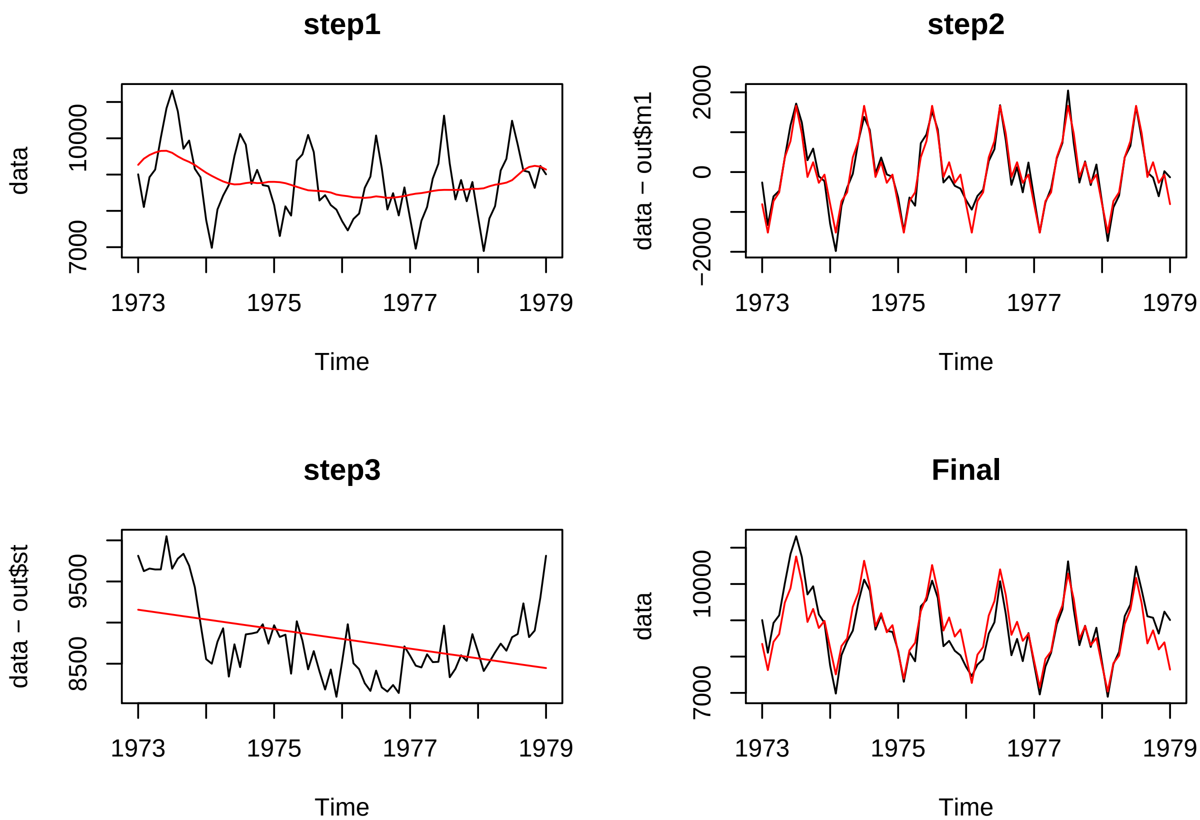
```r
out = classical(data, d=12, order=1);

par(mfrow=c(2,2))
plot.ts(data);
title("step1")
lines(x, out$m1, col="red");

plot.ts(data-out$m1);
title("step2")
lines(x, out$st, col="red");

plot.ts(data-out$st);
title("step3")
lines(x, out$m, col="red");

plot.ts(data);
lines(x, out$fit, col="red")
title("Final")
```



In-class activity: Do HW1 15, part (a)-(e).

In addition,

(f) find the best q for MA(q) by CV. Use following command:

```
hopt = optimize(f=ma.cv, interval=c(5, floor(n/2)), tol=.Machine$double.eps^0.25, Y=data, l=1)
hopt$minimum;
```

(g) Apply classical decomposition and see the result.