Spectral Analysis

Al Nosedal
University of Toronto

Winter 2019

"essentially, all models are wrong, but some are useful"

George E. P. Box

(one of the great statistical minds of the 20th century).

Description of data

The data in **sunspots** shows yearly numbers of sunspots from 1771 to 1870.

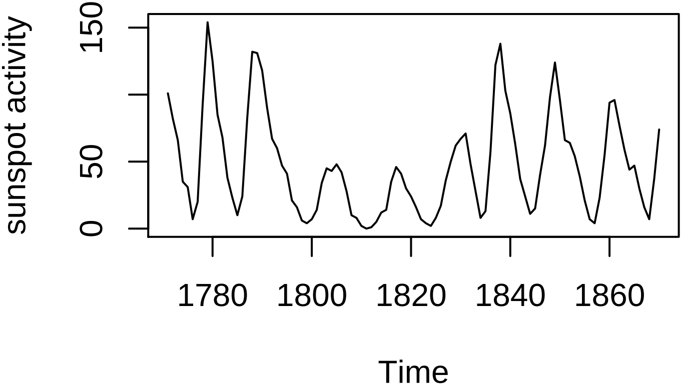
Reading data

```
sunspots<-read.table(file="sunspots.DAT",header=FALSE);

sunspots<-ts(sunspots,start=1771);
```

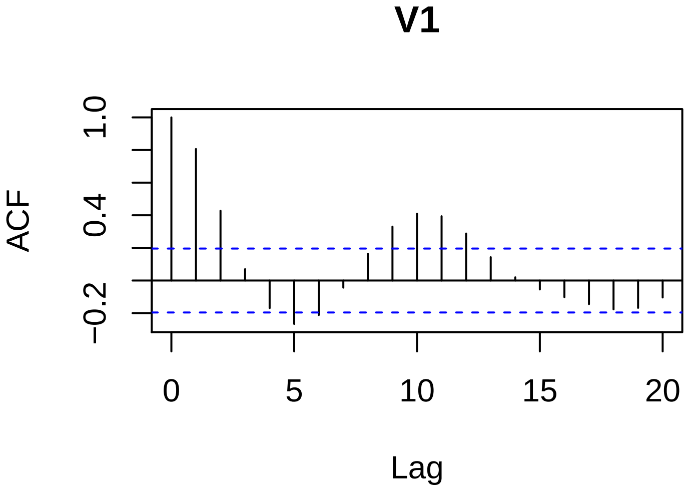
Exercise

1. Make the time series plot of the sunspots.DAT.
2. Make the correlogram (ACF) of the series.

Time Series Plot



Correlogram of 11-year differences for sunspot data



Before attempting to model the series, the cyclical component needs to be removed. A simple procedure which is useful here is to take appropriate **differences** of the series. Here, assuming an 11 year cycle, differences between points 11 years apart are used

$$Y_t = X_t - X_{t-11}$$

(The resulting series should contain no obvious periodic component).

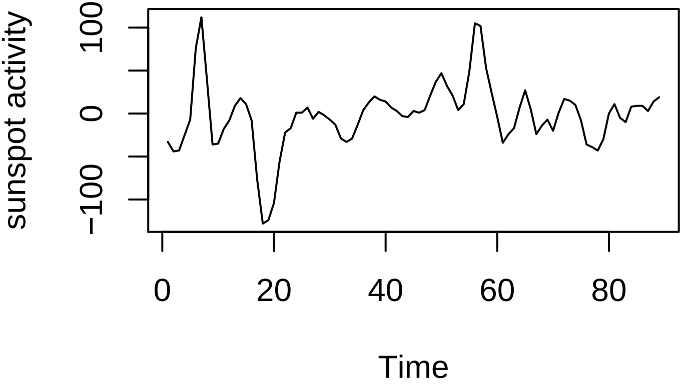
Exercise

1. Find $Y_t = X_t - X_{t-11}$ using R.
2. Make the time series plot of Y_t.
3. Make the correlogram (ACF) of Y_t.
4. Fit AR model using **ar.yw**.
5. Plot residuals from your model.

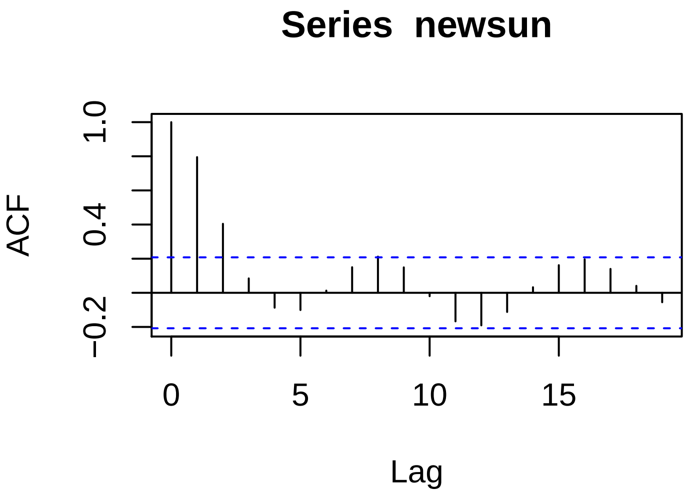
New Series (sunspots)

```
newsun<-sunspots[12:100] - sunspots[1:89]
```

Time Series Plot (newsun)



Correlogram



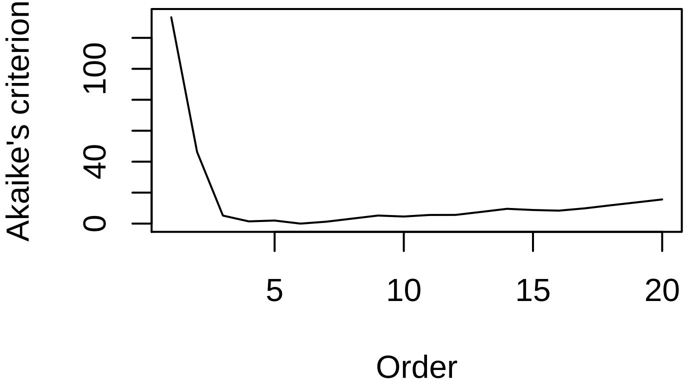
Fitting Model

```
sun.ar<-ar.yw(newsun);

sun.ar

##
## Call:
## ar.yw.default(x = newsun)
##
## Coefficients:
##       1        2        3        4        5
##  1.5009  -1.1516   0.6555  -0.4361   0.2092
##
## Order selected 5  sigma^2 estimated as  311.5
```

Plot of Akaike's criterion

```
plot.ts(sun.ar$aic, xlab="Order",
ylab="Akaike's criterion");
```

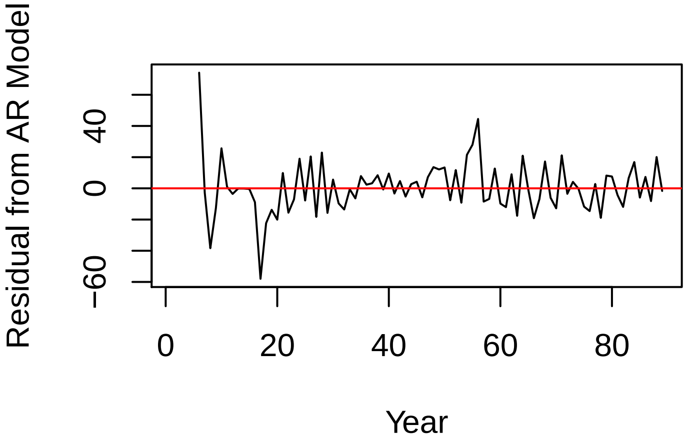
Plot of Akaike's criterion



Plot of Residuals

```
plot.ts(sun.ar$resid, xlab="Year",
ylab="Residual from AR Model");

abline(h=0, col="red");
```

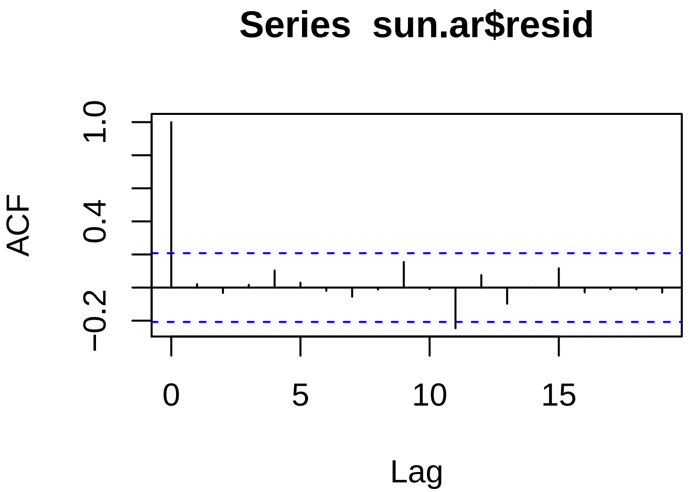
Plot of Residuals



ACF of Residuals

```r
acf(sun.ar$resid);
```

ACF of Residuals



All the correlations (apart from the one corresponding to lag 11) are small and lie within the horizontal bands, indicating that they do not differ significantly from zero. This suggests that the fitted autoregressive model is a reasonable fit for the data.

Spectral Density

If a time series $\{Y_t\}$ has autocovariance $\gamma(k)$ satisfying $\sum_{k=-\infty}^{\infty} |\gamma(k)| < \infty$, then we define its **spectral density** as

$$f(w) = \sum_{k=-\infty}^{\infty} \gamma(k) e^{-2\pi iwk} \quad \text{for} \quad -\infty < w < \infty.$$

We define its **normalized spectral density** as

$$f^*(w) = \sum_{k=-\infty}^{\infty} \rho(k) e^{-2\pi iwk} \quad \text{for} \quad -\infty < w < \infty$$

(where $\rho(k)$ represents its autocorrelation function).

Periodogram

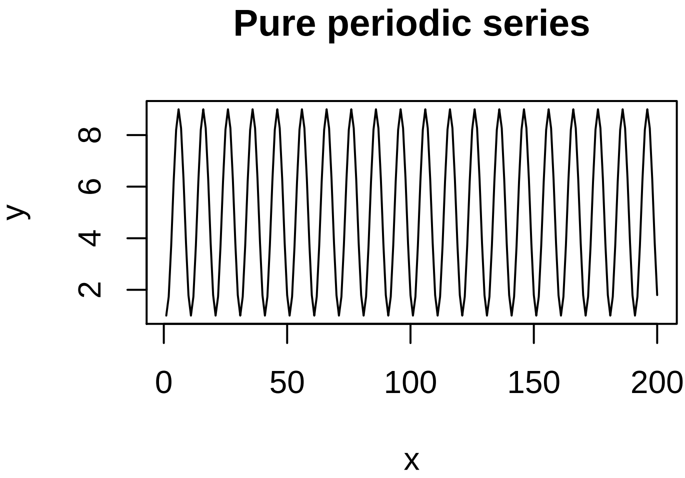
The periodogram is the sample estimate of the power spectrum (or spectral density). It is given by

$$\hat{f}(w) = \hat{C}(0) + 2\sum_{k=1}^{n-1} \hat{C}(k) cos(2\pi kw).$$

Why is the periodogram useful?

The periodogram is ideal for identifying periodicity in data and estimating the frequency of the period. Consider a very simple periodic pattern with no noise producing the Figure shown below (see R code on next slide).

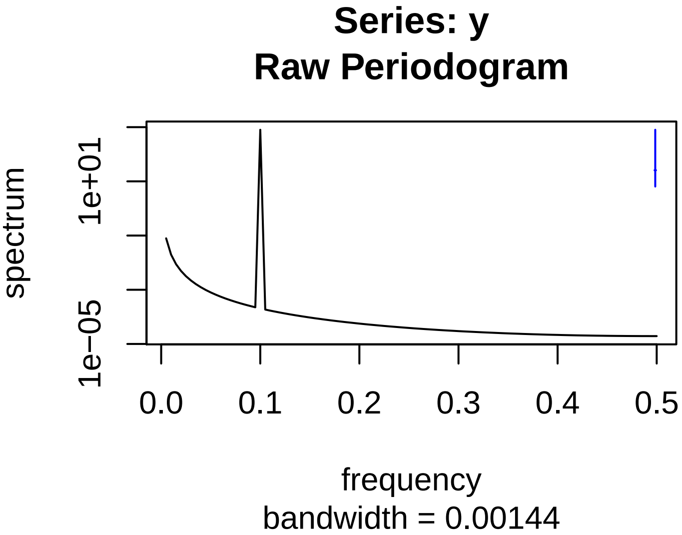
```r
n<-200;
x<-c(1:n);
y<-5+4*cos(2*pi*x/10 + 2.5);
plot(x,y,type="l");
title("Pure periodic series");
```


Pure periodic series
y
2
4
6
8
0
50
100
150
200
x

Periodogram (R code)

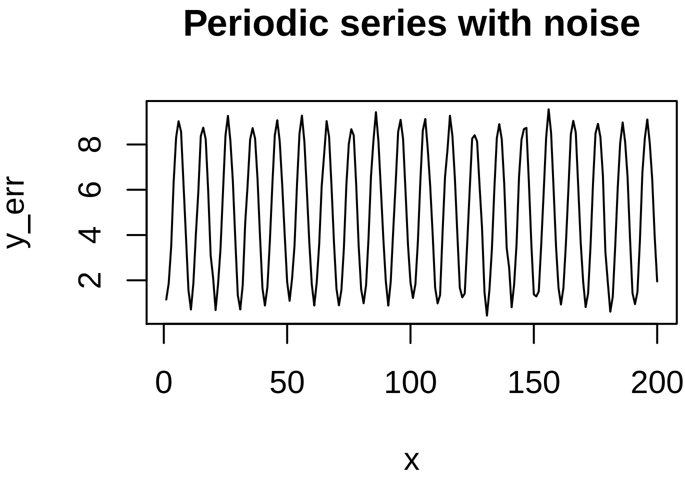
```
z<-spec.pgram(y,fast=FALSE, taper =0.0);
```

Periodogram (Graph)



Noisy version

```
n<-200;
x<-c(1:n);
y<-5+4*cos(2*pi*x/10 + 2.5);
err<-rnorm(n,0,0.25);
y_err<-y + err;
plot(x,y_err,type="l");
title("Periodic series with noise");
```

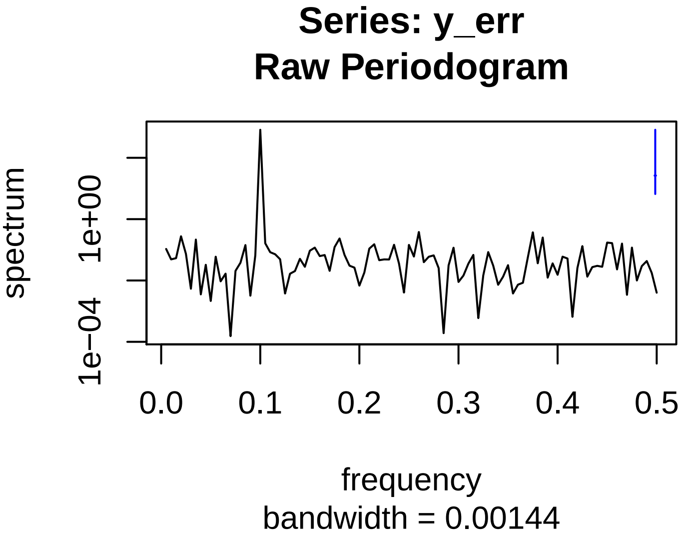


Periodic series with noise
y_err
2
4
6
8
0
50
100
150
200
x

Periodogram (R code)

```
z_err<-spec.pgram(y_err,fast=FALSE, taper =0.0);
```

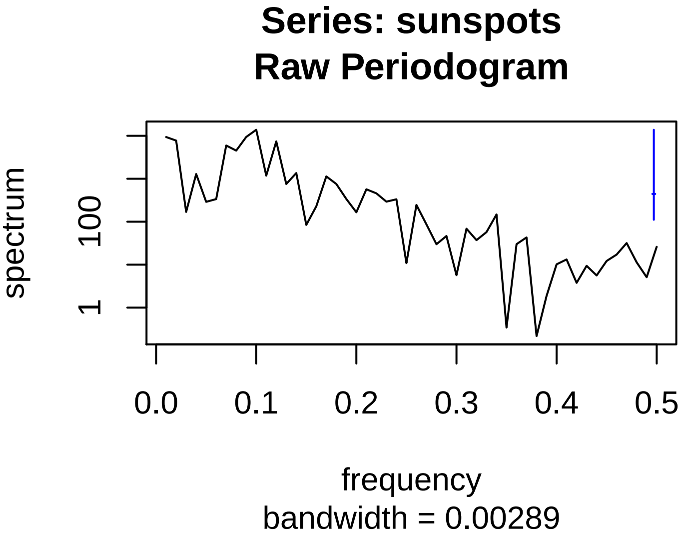
Periodogram (Graph)



The Sunspot Data (R Code)

```
sunspots<-read.table(file="sunspots.DAT",header=FALSE);

sunspots<-ts(sunspots,start=1771);

z_sun<-spec.pgram(sunspots,fast=FALSE, taper =0.0);
```

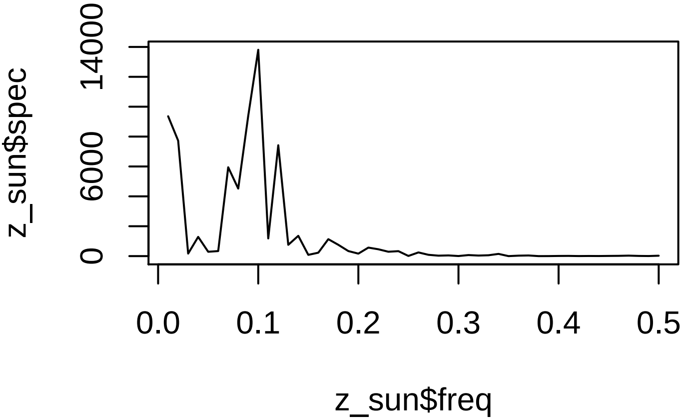
The Sunspot Data (Graph)



The Sunspot Data (R Code...again)

```
names(z_sun);

plot(z_sun$freq,z_sun$spec,type="l");
```

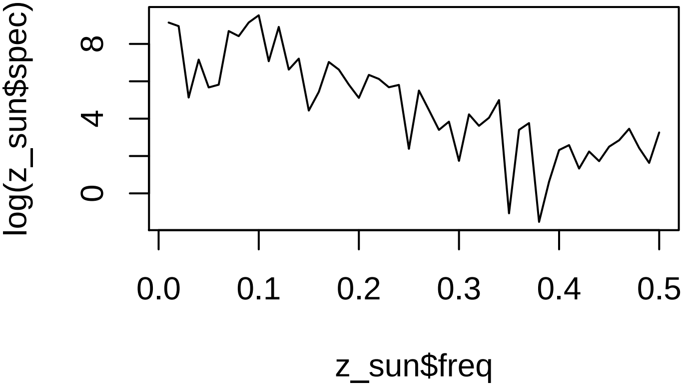
The Sunspot Data (Graph)



The Sunspot Data (R Code...again)

```
names(z_sun);

plot(z_sun$freq,log(z_sun$spec),type="l");
```

The Sunspot Data (Graph)



A General Approach to Time Series Modeling

- Plot the series and examine the main features of the graph, checking in particular whether there is
  a) a trend,
  b) a seasonal component,
  c) any apparent sharp changes in behavior,
  d) any outlying observations.

A General Approach to Time Series Modeling

- Remove the trend and seasonal components to get **stationary** residuals. To achieve this goal it may sometimes be necessary to apply a preliminary transformation to the data. For example, if the magnitude of the fluctuations appears to grow roughly linearly with the level of the series, then the transformed series $\ln(X_t)$ will have fluctuations of more constant magnitude. There are several ways in which trend and seasonality can be removed, some involving estimating components and subtracting them from the data, and others depending on **differencing** the data, i.e., replacing the original series X_t by $Y_t = X_t - X_{t-d}$ for some positive integer d. Whichever method is used, the aim is to produce a stationary series, whose values we shall refer to as **residuals**.

A General Approach to Time Series Modeling

- Choose a model to fit the residuals.
- Forecasting will be achieved by forecasting the residuals and then inverting the transformations described above to arrive at forecasts of the original series X_t.

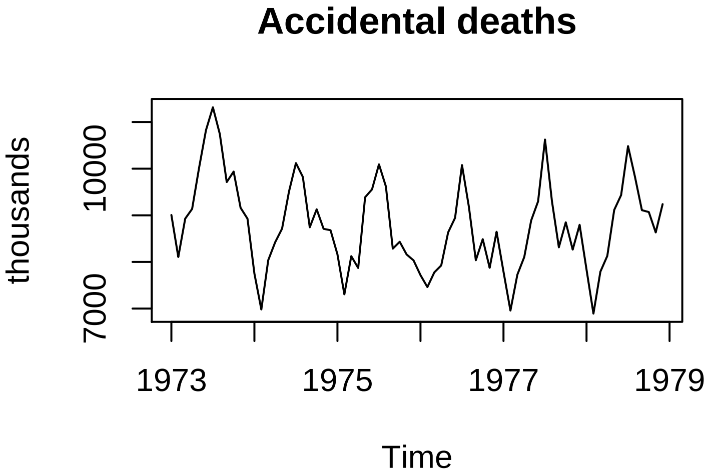
Example: Accidental deaths

The file deaths.DAT contains monthly accidental death figures in the U. S. A. from 1973 to 1978.

Reading data

```
deaths<-read.table(file="deaths.DAT",header=FALSE);

deaths<-unlist(deaths);

deaths.ts<-ts(deaths,start=1973, freq=12);
```

Time Series Plot

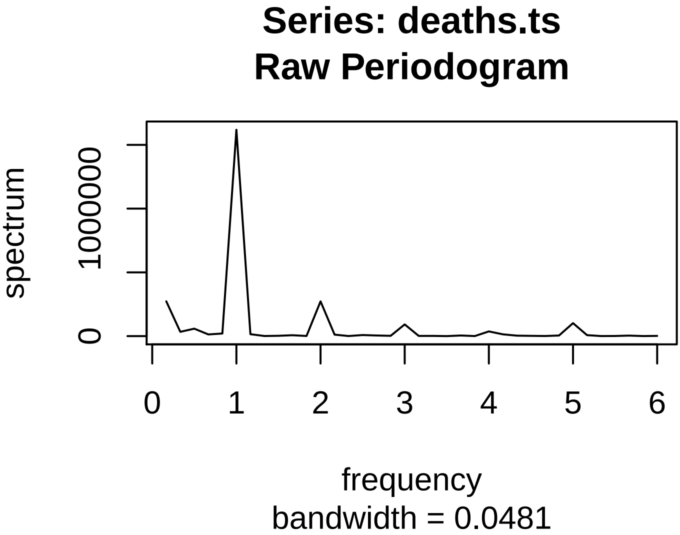


This time series plot shows a strong seasonal pattern. We shall consider the problem of representing the data as the sum of a trend, a seasonal component, and a residual term.

Periodogram (R Code)

```
z_deaths<-spec.pgram(deaths.ts,fast=FALSE,
taper =0.0,log="no");
```

Periodogram (Graph)



Fitting Seasonal Component

```
deaths_mean<-deaths-mean(deaths.ts);
# mean corrected series;
n<-length(deaths.ts);
time<-c(1:n);
col_1<-rep(1,n);
col_c<-cos(2*pi*time/12);
col_s<-sin(2*pi*time/12);
X<-cbind(col_1,col_c,col_s);
fit_deaths<-lm(deaths_mean~ -1 + X);
# no intercept needed;
# it is included in X;
```

Fitting Seasonal Component

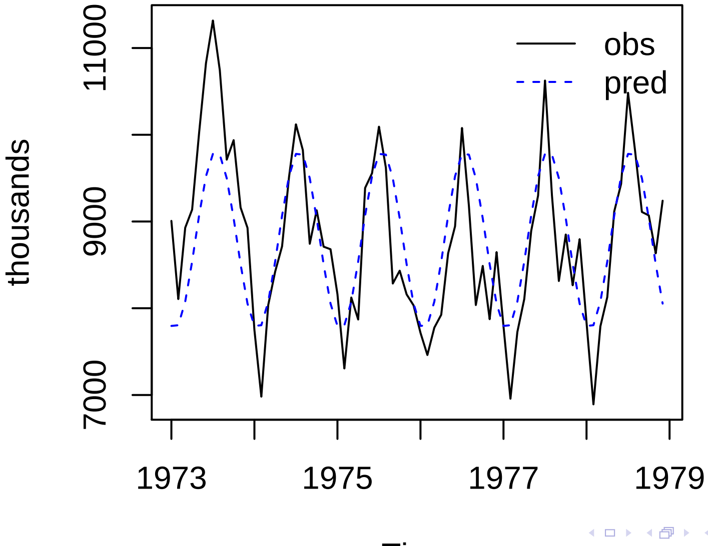
```
summary(fit_deaths)[4]
```

Fitting Seasonal Component

```
## $coefficients
##              Estimate Std. Error       t value     Pr(>|t|)
## Xcol_1 -1.594540e-13   74.50627 -2.140142e-15 1.000000e+00
## Xcol_c -7.340397e+02  105.36778 -6.966453e+00 1.529930e-09
## Xcol_s -7.116420e+02  105.36778 -6.753886e+00 3.703885e-09
```

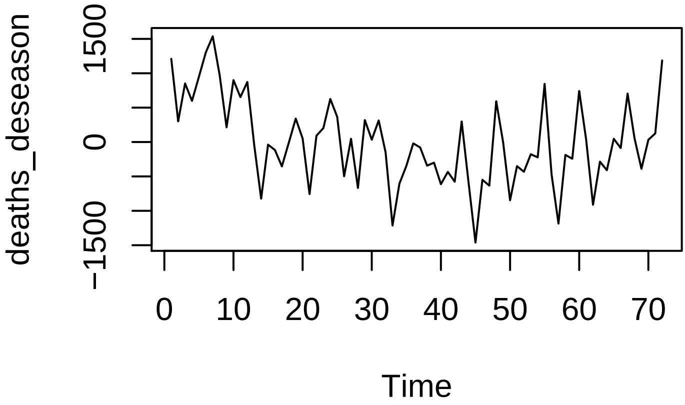
Predictions vs Observations

```
preds_ts<-ts(fit_deaths$fit,start=1973,freq=12);

# turning predictions from linear
#model into a time series object;

plot.ts(deaths.ts,ylab="thousands");

lines(preds_ts+mean(deaths.ts),lty=2, col="blue");

legend("topright",c("obs", "pred"),
lty=c(1,2), col=c("black","blue"),bty="n" );
```


obs
pred
thousands
7000
9000
11000
1973
1975
1977
1979

Graph of deseasonalized data

```
deaths_deseason<-fit_deaths$res;

plot.ts(deaths_deseason);
```


deaths_deseason
−1500
0
1500
0
10
20
30
40
50
60
70
Time

Some comments

The graph of deseasonalized data suggests the presence of an additional quadratic trend function.

Fitting Quadratic Trend

```
n<-length(deaths_deseason);
time<-c(1:n);
col_1<-rep(1,n);
time2<-time^2;
T<-cbind(col_1,time,time2);
fit_trend<-lm(deaths_deseason~ -1+T);
# no intercept needed;
# it is included in T;
```

Fitting Quadratic Trend

```
summary(fit_trend)[4]
```

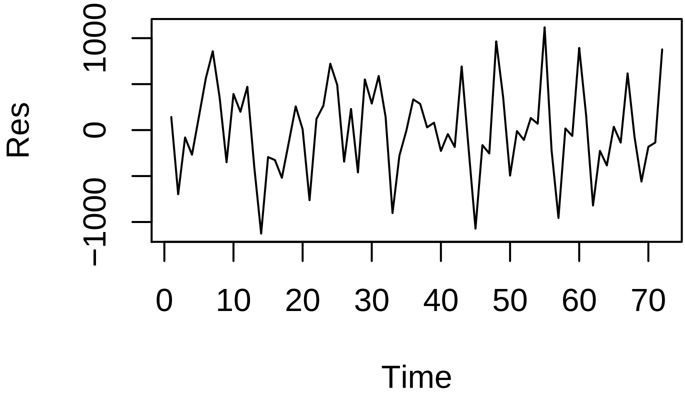
Fitting Quadratic Trend

```
## $coefficients
##              Estimate  Std. Error   t value     Pr(>|t|)
## Tcol_1 1138.8686110 178.3411517  6.385899 1.692203e-08
## Ttime   -71.3667762  11.2743491 -6.330013 2.128145e-08
## Ttime2    0.8309979   0.1496698  5.552210 4.892800e-07
```

Plot of Residuals

```
Res<-fit_trend$res

plot.ts(Res);
```

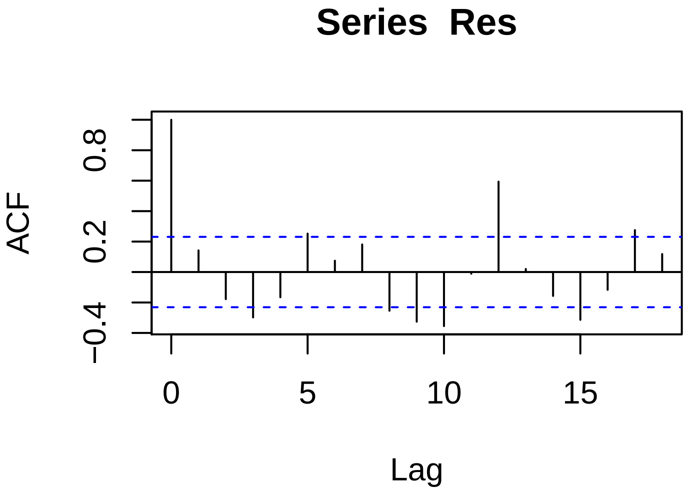
Plot of Residuals



ACF of Residuals

```
acf(Res);
```

ACF of Residuals



Some comments

From our plots, it is clear that there is substantial dependence in the series of residuals.

Modeling Residuals

```
library(forecast);

auto.arima(Res);
```

Modeling Residuals

```
## Series: Res
## ARIMA(1,0,1) with zero mean
##
## Coefficients:
##           ar1     ma1
##       -0.6626  0.9264
## s.e.   0.1164  0.0518
##
## sigma^2 estimated as 210888:  log likelihood=-542.79
## AIC=1091.58   AICc=1091.93   BIC=1098.41
```

The classical decomposition model

$$X_t = m_t + s_t + Y_t$$

where m_t is a slowly changing function known as a **trend component**, s_t is a function with known period d referred to as the **seasonal component**, and Y_t is a **random noise component** that is stationary.

The classical decomposition model

For our example, we have that

$$X_t \approx 8787.7361111 - 734.0397113 cos(\frac{2\pi t}{12}) - 711.6419859 sin(\frac{2\pi t}{12})$$
$$+1138.868611 + -71.3667762t + 0.8309979t^2$$
$$-0.6625909 Y_{t-1} - 0.6625909 W_{t-1} + W_t$$

Another way

```
auto.arima(deaths.ts);
```

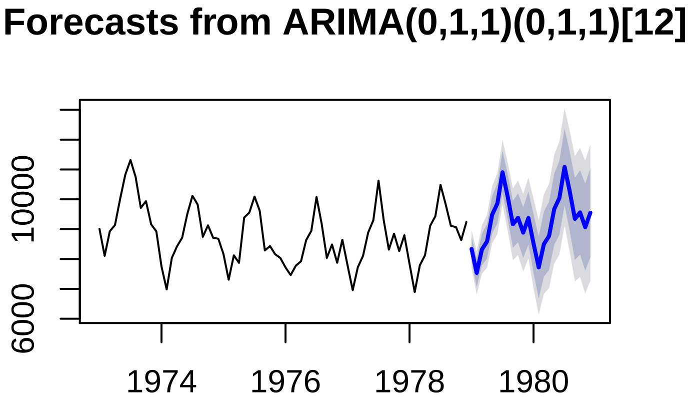
Another way

```
## Series: deaths.ts
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.4264  -0.5584
## s.e.   0.1226   0.1787
##
## sigma^2 estimated as 102999:  log likelihood=-425.53
## AIC=857.06   AICc=857.5   BIC=863.3
```

Forecasts

```r
sarima<-auto.arima(deaths.ts);

fcast<-forecast(sarima);

plot(fcast);
```

Forecasts



Forecasts from ARIMA(0,1,1)(0,1,1)[12]
6000
10000
1974
1976
1978
1980

Some final comments

The corresponding fitted model for $\{X_t\}$ is thus the **SARIMA** $(0,1,1) \times (0,1,1)_{12}$ process

$$(1 - 0.4264B)(1 - 0.5584B^{12})W_t,$$

where $\{W_t\} \sim WN(0, \sigma_W^2 = 99480)$.