02 Time series with trend and seasonality components

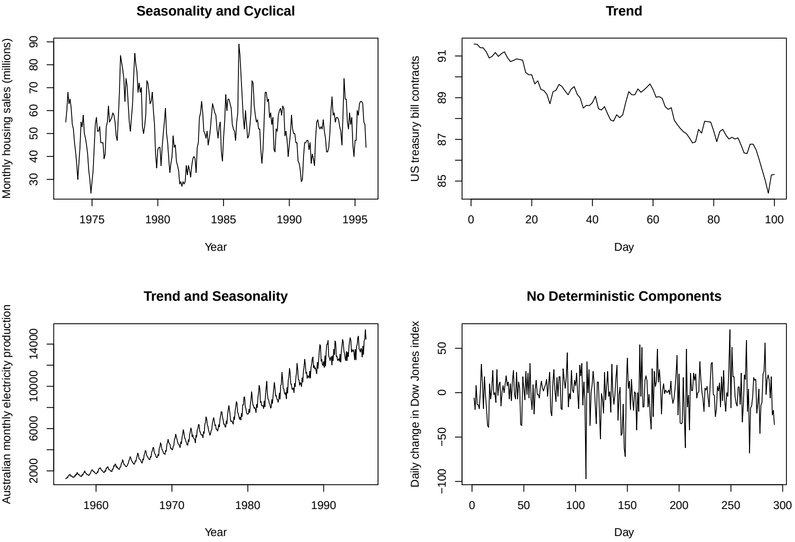
Andrius Buteikis, andrius.buteikis@mif.vu.lt
http://web.vu.lt/mif/a.buteikis/

Time series with deterministic components

Up until now we assumed our time series is generated by a *stationary process* - either a white noise, an autoregressive, a moving-average or an ARMA process.

However, this is not usually the case with real-world data - they are often governed by a (deterministic) **trend** and they might have (deterministic) **cyclical** or **seasonal** components in addition to the **irregular/remainder** (stationary process) component:

- **Trend component** - a long-term increase or decrease in the data which might not be linear. Sometimes the trend might change direction as time increases.
- **Cyclical component** - exists when data exhibit rises and falls that are not of fixed period. The average length of cycles is longer than the length of a seasonal pattern. In practice, the trend component is assumed to include also the cyclical component. Sometimes the trend and cyclical components together are called as trend-cycle.
- **Seasonal component** - exists when a series exhibits regular fluctuations based on the season (e.g. every month/quarter/year). Seasonality is always of a fixed and known period.



In order to remove the deterministic components, we can *decompose* our time series into separate stationary and deterministic components.

Time series decomposition

The general mathematical representation of the decomposition approach:

$$Y_t = f(T_t, S_t, E_t)$$

where

- Y_t is the time series value (actual data) at period t;
- T_t is a **deterministic** trend-cycle or general movement component;
- S_t is a **deterministic** seasonal component
- E_t is the **irregular** (remainder or residual) (stationary) component.

The exact functional form of $f(\cdot)$ depends on the decomposition method used.

Trend Stationary Time Series

A common approach is to assume that the equation has an **additive** form:

$$Y_t = T_t + S_t + E_t$$

Trend, seasonal and irregular components are simply added together to give the observed series.

Alternatively, the **multiplicative** decomposition has the form:

$$Y_t = T_t \cdot S_t \cdot E_t$$

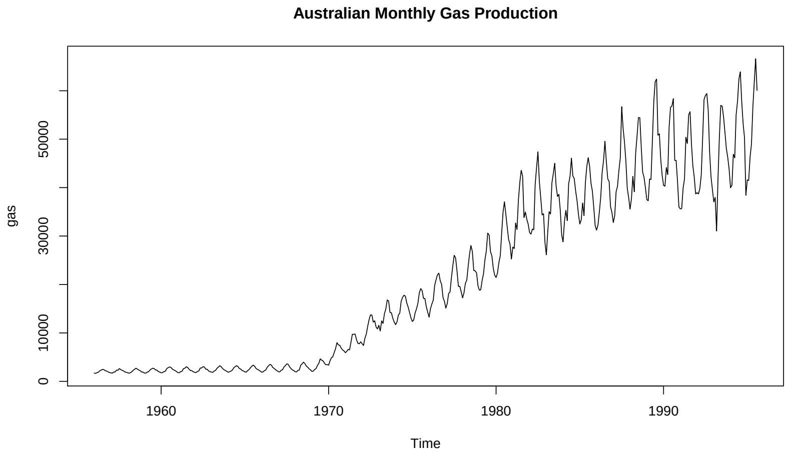
Trend, seasonal and irregular components are *multiplied* together to give the observed series.

In both additive and multiplicative cases the series Y_t is called a **trend stationary** (TS) series.

This definition means that after removing the deterministic part from a TS series, what remains is a **stationary** series.

If our historical data ends at time T and the process is *additive*, we can forecast the *deterministic* part by taking $\widehat{T}_{T+h} + \widehat{S}_{T+h}$, **provided we know the analytic expression for both trend and seasonal parts and the remainder is a WN** .

(Note: time series can also be described by another, *difference stationary* (DS) model, which will be discussed in a later topic)


Australian Monthly Gas Production
gas
0
10000
30000
50000
Time
1960
1970
1980
1990

Additive or multiplicative?

- An *additive* model is appropriate if **the magnitude of the seasonal fluctuations does not vary with the level of time series**;
- The *multiplicative* model is appropriate if **the seasonal fluctuations increase or decrease proportionally with increases and decreases in the level of the series**.

Multiplicative decomposition is more prevalent with economic series because most seasonal economic series do have seasonal variations which increase with the level of the series.

Rather than choosing either an additive or multiplicative decomposition, we could *transform* the data beforehand.

Transforming data of a multiplicative model

Very often the transformed series can be modeled additively when the original data is not additive. In particular, logarithms turn a multiplicative relationship into an additive relationship:

- if our model is

$$Y_t = T_t \cdot S_t \cdot E_t$$

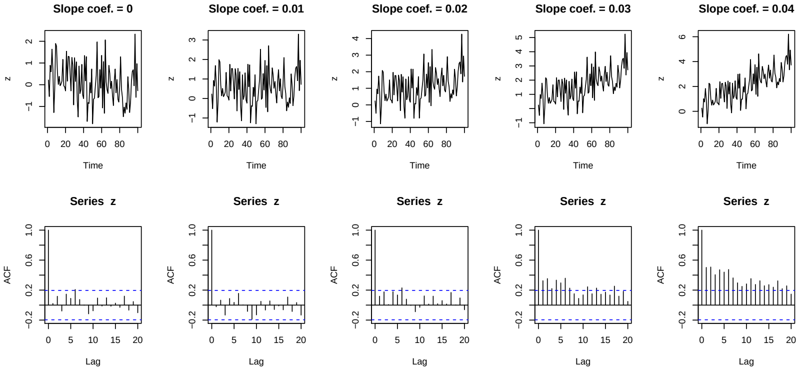
- then taking the logarithms of both sides gives us:

$$log(Y_t) = \log(T_t) + \log(S_t) + \log(E_t)$$

So, we can fit a multiplicative relationship by fitting a more convenient additive relationship to the logarithms of the data and then to move back to the original series by *exponentiating*.

Determining if a time series has a trend component

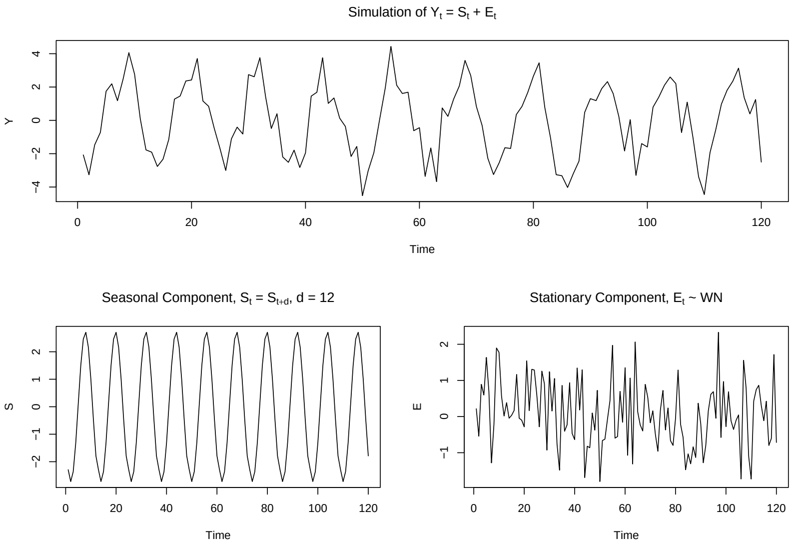
One can use ACF to determine if a time series has a a trend. Some examples by plotting time series with a larger trend (by increasing the slope coefficient): $Y_t = \alpha \cdot t + \epsilon_t$



a non-stationary series with a constant variance and a non-constant mean. The more pronounced the trend, the slower the ACF declines.

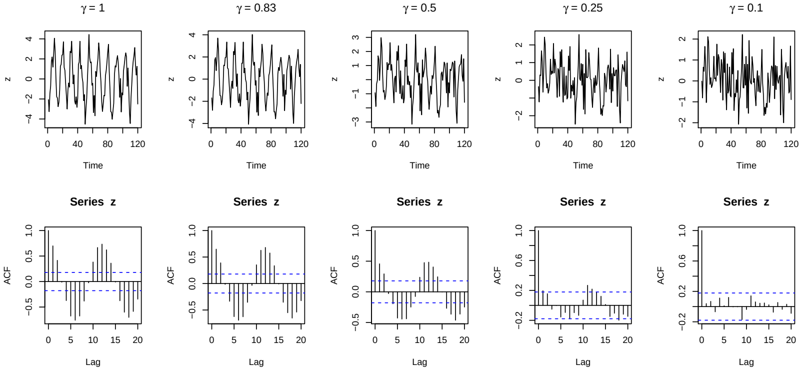
Determining if a time series has a seasonal component

We can use the ACF to determine if seasonality is present in a time series. For example, $Y_t = \gamma \cdot S_t + \epsilon_t$.



Determining if a time series has a seasonal component

Some examples of more pronounced seasonality:

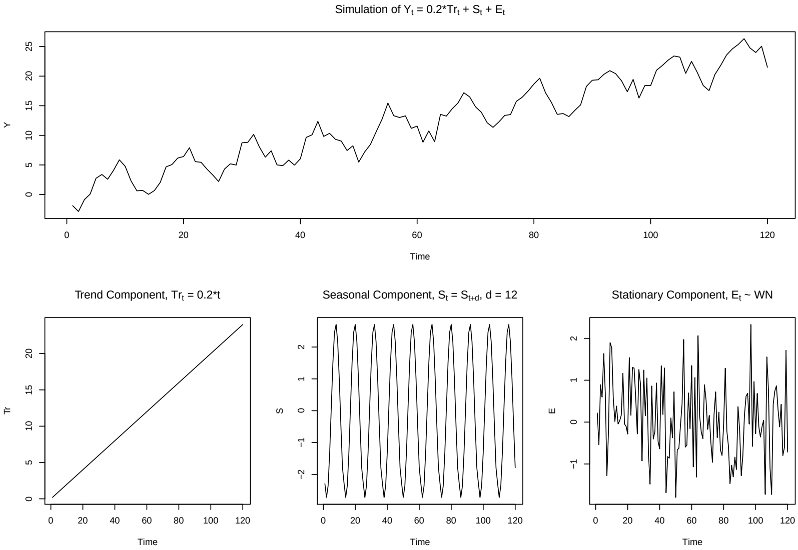


The larger the amplitude of seasonal fluctuations, the more pronounced the oscillations are in the ACF.

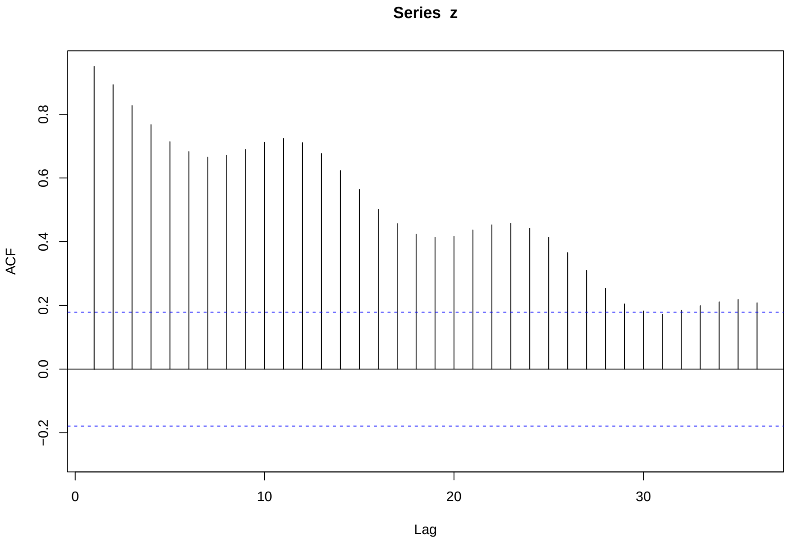
Determining if a time series has both a trend and seasonal component

For a Series with both a Trend and a Seasonal component:

$Y_t = Tr_t + S_t + \epsilon_t$



```r
forecast::Acf(z, lag.max = 36)
```



The ACF exhibits both a slow decline and oscillations.

Basic Steps in Decomposition (1)

1. Estimate the trend. Two approaches:
   - Using a smoothing procedure;
   - Specifying a regression equation for the trend;
2. De-trending the series:
   - For an additive decomposition, this is done by subtracting the trend estimates from the series;
   - For a multiplicative decomposition, this is done by dividing the series by the estimated trend values.
3. Estimating the seasonal factors from the de-trended series:
   - Calculate the mean (or median) values of the de-trended series for each specific period (for example, for monthly data - to estimate the seasonal effect of January - average the de-trended values for all January values in the series etc);
   - Alternatively, the seasonal effects could also be estimated along with the trend by specifying a regression equation.

The number of seasonal factors is equal to the frequency of the series (e.g. monthly data = 12 seasonal factors, quarterly data = 4, etc.).

Basic Steps in Decomposition (2)

4. The seasonal effects should be normalized:
   - For an additive model, seasonal effects are adjusted so that the average of d seasonal components is 0 (this is equivalent to their sum being equal to 0);
   - For a multiplicative model, the d seasonal effects are adjusted so that they average to 1 (this is equivalent to their sum being equal to d);
5. Calculate the irregular component (i.e. the residuals):
   - For an additive model $\widehat{E}_t = Y_t - \widehat{T}_t - \widehat{S}_t$
   - For a multiplicative model $\widehat{E}_t = \dfrac{Y_t}{\widehat{T}_t \cdot \widehat{S}_t}$;
6. Analyze the residual component. Whichever method was used to decompose the series, the aim is to produce **stationary** residuals.
7. Choose a model to fit the *stationary* residuals (e.g. see ARMA models).
8. Forecasting can be achieved by forecasting the residuals and combining with the forecasts of the trend and seasonal components.

Estimating the trend, T_t

There are various ways to estimate the trend T_t at time t but a relatively simple procedure which does not assume any specific form of T_t is to calculate a *moving average* centered on t.

A *moving average* is an average of a specific number of time series values **around** each value of t in the time series, with the exception of the first few and last few terms (this procedure is available in `R` with the `decompose` function). This method **smooths** the time series.

The estimation depends on the seasonality of the time series:

- If the time series has **no seasonal component**;
- If the time series contains **a seasonal component**;

Smoothing is usually done to help us better see patterns (like the trend) in the time series by smoothing out the irregular roughness to see a clearer signal. For seasonal data, we might smooth out the seasonality so that we can identify the trend.

Estimating T_t if the time series has no seasonal component

In order to estimate the trend, we can take any **odd** number, for example, if $l = 3$, we can estimate an additive model:

$$\widehat{T}_t = \frac{Y_{t-1} + Y_t + Y_{t+1}}{3}, \text{ (two-sided averaging)}$$

$$\widehat{T}_t = \frac{Y_{t-2} + Y_{t-1} + Y_t}{3}, \text{ (one-sided averaging)}$$

In this case, we are calculating the averages, either:

- centered around t - one element to the left (past) and one element to the right (future),
- or alternatively - two elements to the left of t (past values at $t-1$ and $t-2$).

Estimating T_t if the time series contains a seasonal component

If the time series contains a seasonal component and we want to average it out, the length of the moving average **must be equal to the seasonal frequency** (for monthly series, we would take $l = 12$). However, there is a slight hurdle.

Suppose, our time series begins in January ($t = 1$) and we average up to December ($t = 12$). This averages corresponds to a time $t = 6.5$ (time between June and July).

When we come to estimate seasonal effects, we need a moving average at integer times. This can be achieved by averaging the average of January to December and the average of February ($t = 2$) up to January ($t = 13$). This average of the two moving averages corresponds to $t = 7$ and the process is called **centering**.

Thus, the trend at time t can be estimated by the centered moving average:

$$\widehat{T}_t = \frac{(Y_{t-6} + ... + Y_{t+5})/12 + (Y_{t-5} + ... + Y_{t+6})/12}{2}$$
$$= \frac{(1/2)Y_{t-6} + Y_{t-5} ... + Y_{t+5} + (1/2)Y_{t+6}}{12}$$

where $t = 7, ..., T - 6$.

By using the seasonal frequency for the coefficients in the moving average, the procedure generalizes for any seasonal frequency (i.e. quarterly, weekly, etc. series), provided the condition that the coefficients sum up to unity is still met.

Estimating the seasonal component, S_t

An estimate of S_t at time t can be obtained by subtracting $\widehat{T}_t$:

$$\widehat{S}_t = Y_t - \widehat{T}_t$$

*By **averaging** these estimates of the monthly effects for each month (January, February etc.), we obtain a single estimate of the effect for each month*. That is, if the seasonality period is d, then:

$$S_t = S_{t+d}$$

Seasonal factors can be thought of as expected variations from trend throughout a seasonal period, so we would expect them to cancel each other out over that period - i.e., they should add up to zero.

$$\sum_{t=1}^{d} S_t = 0$$

Note that this applies to the *additive decomposition*.

Estimating the seasonal component, S_t

If the estimated (average) seasonal factors $\widetilde{S}_t$ do not add up to zero, then we can correct them by dividing the sum of the seasonal estimates by the seasonality period and adjusting each seasonal factor. For example, if the seasonal period is d, then:

1. Calculate the total sum: $\sum_{t=1}^{d} \widetilde{S}_t$
2. Calculate the value $w = \dfrac{\sum_{t=1}^{d} \widetilde{S}_t}{d}$
3. Adjust each period $\widehat{S}_t = \widetilde{S}_t - w$

Now, the seasonal components add up to zero: $\sum_{t=1}^{d} \widehat{S}_t = 0$.

It is common to present economic indicators such as unemployment percentages as seasonally adjusted series.

This highlights any trend that might otherwise be masked by seasonal variation (for example, to the end of the academic year, when schools and university graduates are seeking work).

If the seasonal effect is additive, a seasonally adjusted series is given by $Y_t - \widehat{S}_t$.

The described moving-average procedure usually quite successfully describes the time series in question, however **it does not allow to forecast it**.

Remark

To decide upon the mathematical form of a trend, one must first draw the plot of the time series.

If the behavior of the series is rather ‘regular’, one can choose a parametric trend - usually it is a low order polynomial in t, exponential, inverse or similar functions.

The most popular method to estimate the coefficients of the chosen function is OLS, however, the form could also be described by certain computational algorithms (one of which will be presented later on).

In any case, the smoothing method is **acceptable** if the residuals $\widehat{\epsilon}_t = Y_t - \widehat{T}_t - \widehat{S}_t$ constitute a *stationary* process.

If we have a few competing trend specifications, the *best* one can be chose by **AIC**, **BIC** or similar criterions.

An alternative approach is to create models for all but some T_0 **end points** and then choose the model whose forecast fits the original data best. To select the model, one can use such characteristics as:

- Root Mean Square Error:

$$RMSE = \sqrt{\frac{1}{T_0} \sum_{t=T-T_0}^{T} \widehat{\epsilon}_t^2}$$

- Mean Absolute Percentage Error:

$$MAPE = \frac{100}{T_0} \sum_{t=T-T_0}^{T} \left| \frac{\widehat{\epsilon}_t}{Y_t} \right|$$

and similar statistics.

The Global Methods of Decomposition and Forecasting - OLS

The OLS method estimates the coefficients of, say, quadratic trend:

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t$$

by minimizing:

$$RSS(\beta_0, \beta_1, \beta_2) = \sum_{t=1}^{T} (Y_t - (\beta_0 + \beta_1 t + \beta_2 t^2))^2$$

Note that if the value of the last Y_T for whatever reason deviates much from the trend - this may considerably change the estimates $\widehat{\beta}_0, \widehat{\beta}_1$ and $\widehat{\beta}_2$ and, therefore, the fitted values of the first $\widehat{Y}_1$.

This is why we term the method `global`. One local method which little alters the estimate of Y_1, following a change in a remote Y_T, will be examined in the next section.

Example

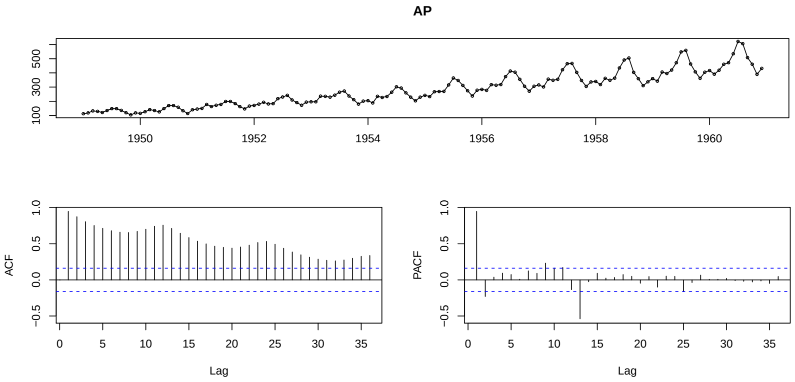
We shall examine the number of international passenger bookings (in thousands) per month on an airline in the US, 1949:1 - 1960:12. We shall create three models:

- Model 1: $AP_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t$;
- Model 2: $AP_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \gamma_1 dm1_t + ... + \gamma_{11} dm11_t + \epsilon_t$;
- Model 3: $logAP_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \gamma_1 dm1_t + ... + \gamma_{11} dm11_t + \epsilon_t$;

where $t = 1, ..., 144$ is for the *trend*, $dm1$ is the dummy variable for the *1st* month, $dm2$ - second month etc.

Recall that in order to avoid the dummy variable trap, we have to exclude one dummy variable (in this case, we exclude $dm12$) from our regression models.

```r
suppressPackageStartupMessages({
  library(forecast)
  library(fma)
})
data(airpass)
AP <- airpass
AP <- ts(AP, start = c(1949, 1), freq = 12)
tsdisplay(AP)
```



We need to create the additional variables:

```
t = time(AP)
AP <- data.frame(AP, t)
for(j in 1:11){
  val <- j + 12 *(0:(nrow(AP)/12))
  val <- val[val < nrow(AP)]
  tmp <- rep(0, times = nrow(AP))
  tmp[val] <- 1
  AP <- cbind(AP, tmp)
}
colnames(AP) <- c("AP", "t", paste0("dm", 1:11))
AP <- ts(AP, start = c(1949, 1), freq = 12)
```

Note: alternatively, when dealing with time series data, we can use `seasonaldummy()` function to generate the seasonal dummies of our data.

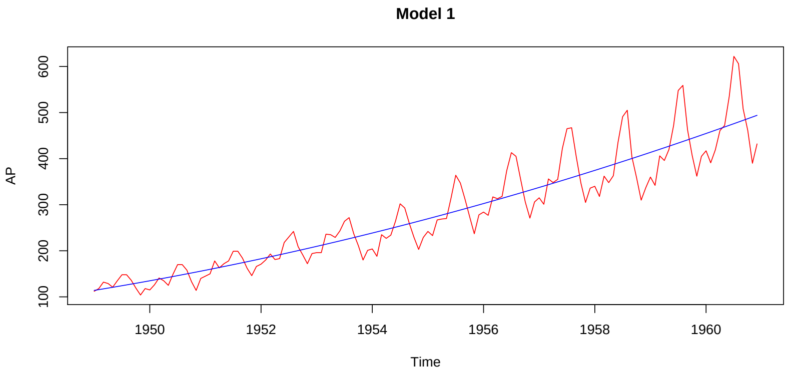
We will now estimate the separate models:

```
AP.lm1 = lm(AP ~ t + I(t^2), data = AP)
AP.lm2 = lm(AP ~ t + I(t^2) +., data = AP)
AP.lm3 = lm(log(AP) ~ t + I(t^2) +., data = AP)
```

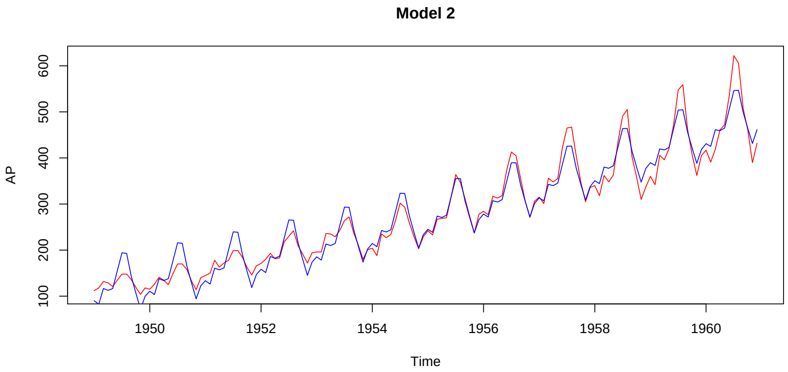
You can view the summary statistics of each model with the `summary` function.

We can now View the resulting models using the `fitted` function:

```
plot(AP[,"AP"], main = "Model 1", type = "l", ylab = "AP",
     col = "red")
lines(ts(fitted(AP.lm1), start = c(1949, 1), freq = 12),
      col = "blue")
```



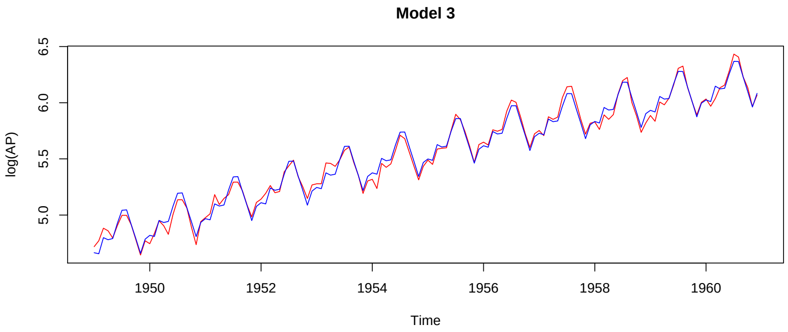
While the first model does capture the trend quite well, it does not capture the seasonal fluctuations.



The second model attempts to capture the seasonal effect, however, it is captured in the wrong way - in the historic data, the seasonal fluctuations increase together with the level, but in the fitted values they don't. It appears that the actual data might be better captured via a multiplicative model.

To correct for multiplicativity, we created the last model for logarithms.

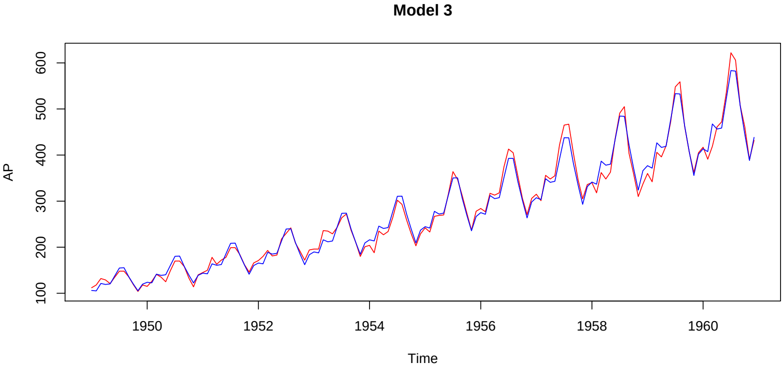
```
plot(log(AP[,"AP"]), main = "Model 3", type = "l",
     ylab = "log(AP)", col = "red")
lines(ts(fitted(AP.lm3), start = c(1949, 1), freq = 12),
      col = "blue")
```



Note: we also need to check if the residuals are *WN*. Otherwise, we need to specify a different model or a separate model for the *stationary* residuals.

To get the fitted values for the original time series instead of the logarithm, we can take the exponent of the fitted values:

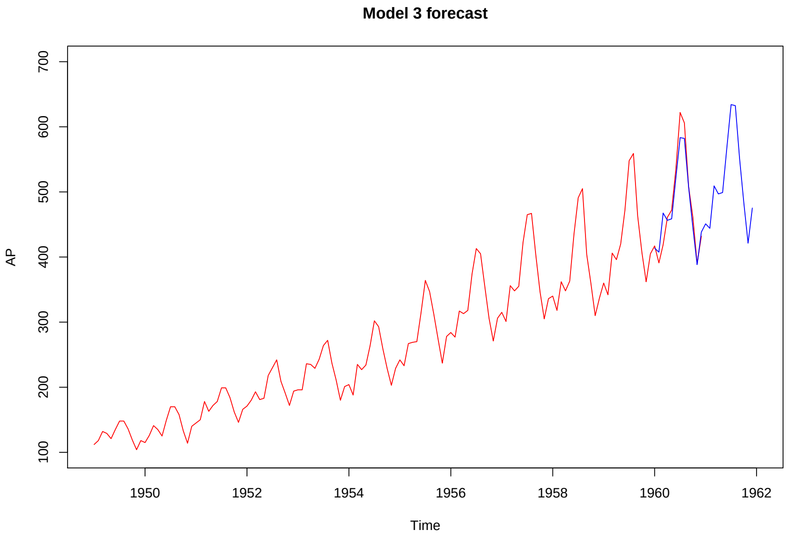
```
plot(AP[,"AP"], main = "Model 3", type = "l",
     ylab = "AP", col = "red")
lines(ts(exp(fitted(AP.lm3)), start = c(1949, 1), freq = 12),
      col = "blue")
```



If we want, we can forecast the data:

```
new.dt <- data.frame(t = seq(1960, by = 1/12,
                              length.out = 24))
for(j in 1:11){
  val <- j + 12 *(0:(nrow(new.dt)/12))
  val <- val[val < nrow(new.dt)]
  tmp <- rep(0, times = nrow(new.dt))
  tmp[val] <- 1
  new.dt <- cbind(new.dt, tmp)
}
colnames(new.dt) <- c("t", paste0("dm", 1:11))
AP.lm3.forc <- predict(AP.lm3, new.dt)
AP.lm3.forc <- ts(AP.lm3.forc,
                  start = 1960, freq = 12)
```

```r
plot(AP[,"AP"], main = "Model 3 forecast", type = "l",
     ylab = "AP", col = "red",
     xlim = c(1949, 1962), ylim = c(100, 700))
lines(exp(AP.lm3.forc), col = "blue")
```



One Local Method of Decomposition and Forecasting

We will present a short introduction to exponential smoothing.

Exponential smoothing is a technique that can be applied to times series data, either to produce smoothed data for presentation, or **to make forecasts**.

Exponential smoothing and *ARIMA* models are the two most widely-used approaches to time series forecasting, and provide complementary approaches to the problem.

While ARIMA models aim to describe the autocorrelations in the data, exponential smoothing models are based on a description of the trend and seasonality in the data.

We will examine three types of Exponential Smoothing:

- Single (i.e. simple)
- Double
- Triple

Simple Exponential Smoothing

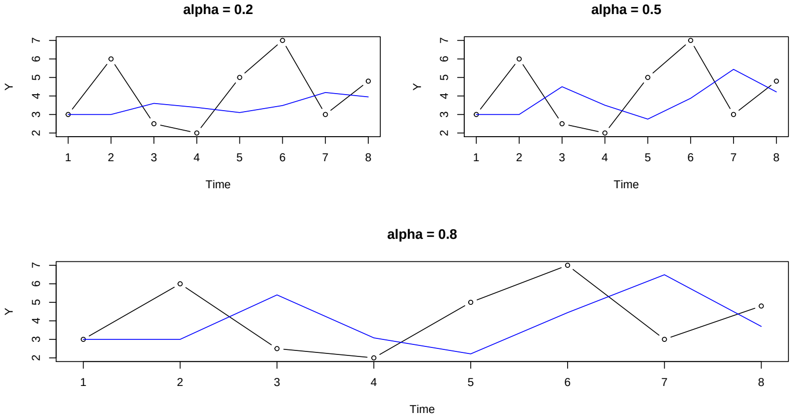
This method is suitable for forecasting data with no clear trend or seasonal pattern.

We state the exponential smoothing procedure as an algorithm for converting the observed series Y_t into a smoothed series $\widehat{Y}_t$, $t = 1, ..., T$ and forecasts $\widehat{Y}_{T+h,T}$:

1. Initialize at $t = 1$: $\widehat{Y}_1 = Y_1$;
2. Update: $\widehat{Y}_t = \alpha Y_{t-1} + (1 - \alpha)\widehat{Y}_{t-1}$, $t = 2, ..., T$;
3. Forecast: $\widehat{Y}_{T+h,T} = \widehat{Y}_T$, $h = 1, 2, ...$.

We call $\widehat{Y}_t$ the estimate of the *level* at time t. The smoothing parameter α is in the unit interval, $\alpha \in [0, 1]$.

The smaller α is, the smoother the estimated level. As α approaches 0, the smoothed series approaches constancy, and as α approaches 1, the smoothed series approaches point-by-point interpolation.



Typically, the more observations we have per unit of calendar time, the more smoothing we need - we would smooth weekly data more than quarterly data. There is no substitute, however, for a trial-and-error approach involving a variety of values of the smoothing parameter.

Double Exponential Smoothing - Holt's Linear Method

Now imagine that we have not only a slowly evolving local level, but also a *trend* with a slowly evolving local slope. Then the optimal smoothing algorithm is as follows:

1. Initialize at $t = 2$: $\widehat{Y}_2 = Y_2$, $F_2 = Y_2 - Y_1$;
2. Update:

$$\widehat{Y}_t = \alpha Y_t + (1-\alpha)(\widehat{Y}_{t-1} + F_{t-1}), \quad 0 < \alpha < 1;$$
$$F_t = \beta(\widehat{Y}_t - \widehat{Y}_{t-1}) + (1-\beta)F_{t-1}, \quad 0 < \beta < 1, \quad t = 3, ..., T;$$

3. Forecast: $\widehat{Y}_{T+h,T} = \widehat{Y}_T + hF_T$.

where $\widehat{Y}_t$ is the estimated, or smoothed, level at time t and F_t is the estimated slope at time t.

The parameter α controls smoothing of the level and β controls smoothing of the slope.

The h-step ahead forecast simply takes the estimated level at time T and augments it with h times the estimated slope at time T.

Triple Exponential Smoothing - Holt-Winters' Method

- If the data has no trend or seasonal patterns, then the simple exponential smoothing is appropriate;
- If the data exhibits a linear trend, then Holt's linear method is appropriate;
- However, if the data is *seasonal*, these methods on their own cannot handle the problem well.

A method known as *Holt-Winters method* is based on three smoothing equations:

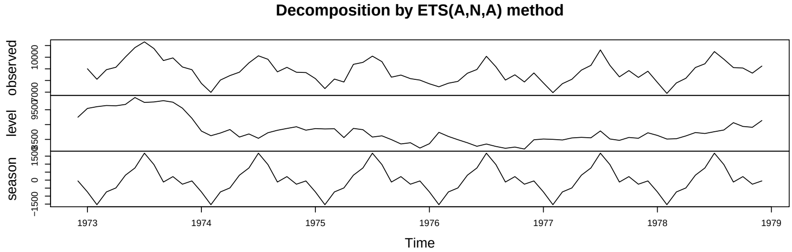
- Level (overall) smoothing;
- Trend smoothing;
- Seasonality smoothing.

It is similar to Holt's linear method, with one additional equation dealing with seasonality.

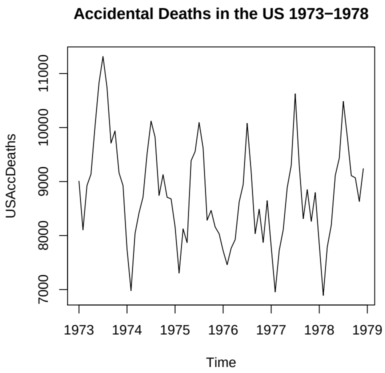
Example

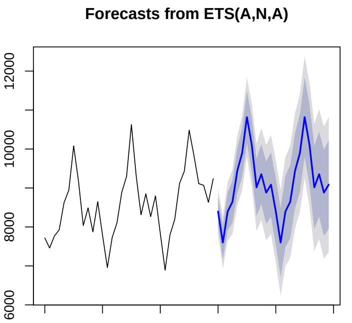
The `ets` function from the `forecast` package represents a fully automated procedure (the best model is elected according to its AIC) based on the exponential moving average filter.
As an example, we shall smooth the data of accidental deaths in the US in 1973-1978:

```
data(USAccDeaths)
US.ad <- ets(USAccDeaths)
plot(US.ad)
```

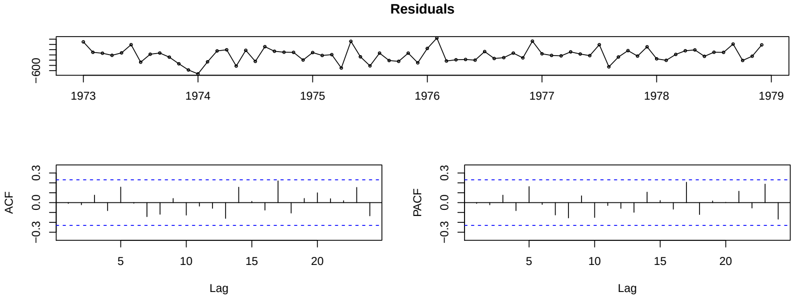


```r
par(mfrow = c(1,2))
plot(USAccDeaths,
     main = "Accidental Deaths in the US 1973-1978")
plot(forecast(US.ad), include = 36)
```





```r
tsdisplay(US.ad$residuals, main = "Residuals")
```



Recall that this decomposition is valid **only if** the irregular part (residuals) of our model make a *stationary* process. In this case, the residuals seem to form a stationary process.

Remark

The h-step ahead forecast of an additive TS time series
$Y_t = T_t + S_t + E_t$, $t = 1, ..., T$ is given by:
$Y_{T+h,T,T} = \widehat{T}_{T+h} + \widehat{S}_{T+h}$, provided E_t is a *WN* process.
If the residuals $\widehat{E}_t$ constitute a more complicated stationary process (AR, MA, ARMA etc.), the forecast should take into account their structure.

There are many more `R` functions for decomposition and/or smoothing: `StructTS`, `decompose`, `stl`, `tsSmooth`, `ts`, `ma`, `ses`, `lowess`, etc.
However, most of them do not allow to forecast the series under consideration.

Combining Different Decomposition Methods

We can also combine the moving average with these methods:

1. Evaluate the trend, $\widehat{T}_t$ via moving average smoothing method;
2. Estimate and normalize the seasonal factors, $\widehat{S}_t$, from the de-trended series;
3. Deseasonalize the data by removing the seasonal component from the series only (i.e. do **not** remove the trend component from the series): $\widetilde{Y}_t = Y_t - \widehat{S}_t$;
4. Re-estimate the trend, $\widehat{T}_t^{(2)}$, from the deseasonalized data using either a (polynomial) regression, exponential smoothing, or any other method, which allows forecasting the trend;
5. Analyse the residuals $\widehat{E}_t = Y_t - \widehat{S}_t - \widehat{T}_t^{(2)}$ - verify that they are stationary and specify their model (if needed).
6. Forecast the series $\widehat{Y}_{T+h}$. Remember that $\widehat{S}_t = \widehat{S}_{t+d}$ means that we can always forecast the seasonal component.

Differencing to de-trend a series

Instead of attempting to remove the noise by smoothing the series or estimating an OLS regression, we can attempt to eliminate the trend by differencing:

$$\nabla X_t = X_t - X_{t-1} = (1 - L)X_t$$
$$\nabla^k X_t = \nabla^{k-1}(X_t - X_{t-1}) = \nabla^{k-1}X_t - \nabla^{k-1}X_{t-1} = ...$$

If our time series is a linear function: $Y_t = \beta_0 + \beta_1 \cdot t + \epsilon_t$
Then the differenced series does not have the trend:

$$\nabla Y_t = \beta_1 \cdot t - \beta_1 \cdot (t-1) + \epsilon_t - \epsilon_{t-1} = \beta_1 + \nabla\epsilon_t$$

In the same way, any polynomial trend of degree k can be removed by applying the operator ∇^k.
In practice, the order k to remove the trend is often quite small $k = 1, 2$.
It should be noted that by differencing the data, we are reducing our sample size. The interpretation also changes, since we are now working with differences, rather than levels of Y_t.

Differencing to de-seasonalize a series

If our time series contains a seasonal component (and a trend):

$$Y_t = \beta_0 + \beta_1 \cdot t + S_t + \epsilon_t, \quad S_t = S_{t+d}$$

Then, if we define our difference operator as:

$$\nabla_d X_t = X_t - X_{t-d} = (1 - L^d) X_t$$
$$\nabla_d^k X_t = \nabla_d^{k-1}(X_t - X_{t-d}) = \nabla_d^{k-1} X_t - \nabla_d^{k-1} X_{t-d} = ...$$

Then the differenced series does not have a seasonal component:

$$\nabla_d Y_t = \beta_1 \cdot t - \beta_1 \cdot (t - d) + S_t - S_{t-d} + \epsilon_t - \epsilon_{t-d} = \beta_1 \cdot d + \nabla_d \epsilon_t$$

Usually $k = 1$ is sufficient to remove seasonality. Note that we have also removed the trend and instead have a constant $\beta_1 \cdot d$, although we may need to apply both a non-seasonal first difference and a seasonal difference if we want to completely remove the trend and seasonality.
Our data interpretation is also different since we are now working with period-differences of the series, $\nabla_d Y_t$, instead of the levels Y_t.

Seasonal ARMA models

The seasonal ARIMA model incorporates both non-seasonal and seasonal factors in a multiplicative model: $SARIMA(p, d, q)(P, D, Q)_S$.

For now, we will restrict our analysis to non-differenced data $SARMA$ models (i.e. $d = 0$ and $D = 0$), where p, q are the $ARMA$ orders of the non-seasonal components and P, Q are the $ARMA$ orders of the seasonal components.

For example, our series could be described as a seasonal (e.g. quarterly) process:

$$Y_t = \Phi Y_{t-1} + w_t + \Theta w_{t-4}$$

while our shocks w_t could also be a non-seasonal MA process:

$$w_t = \epsilon_t + \theta \epsilon_{t-1}$$

So, while the seasonal term is **additive**, the combined model is **multiplicative**:

$$\begin{aligned} Y_t &= \Phi Y_{t-1} + w_t + \Theta w_{t-4} \\ &= \Phi Y_{t-1} + \epsilon_t + \theta \epsilon_{t-1} + \Theta \epsilon_{t-4} + \theta \Theta \epsilon_{t-5} \end{aligned}$$

We can write the general model formally as:

$$\Phi(L^S)\phi(L)(Y_t - \mu) = \Theta(L^S)\theta(L)\epsilon_t$$

where $\phi(z) = 0, \forall |z_i| > 1$ and $\Phi(z) = 0, \forall |z_j| > 1$, and:

- The non-seasonal components are:

$$\text{AR: } \phi(L) = 1 - \phi_1 L - ... - \phi_p L^p$$
$$\text{MA: } \theta(L) = 1 + \theta_1 L + ... + \theta_q L^q$$

- The seasonal components are:

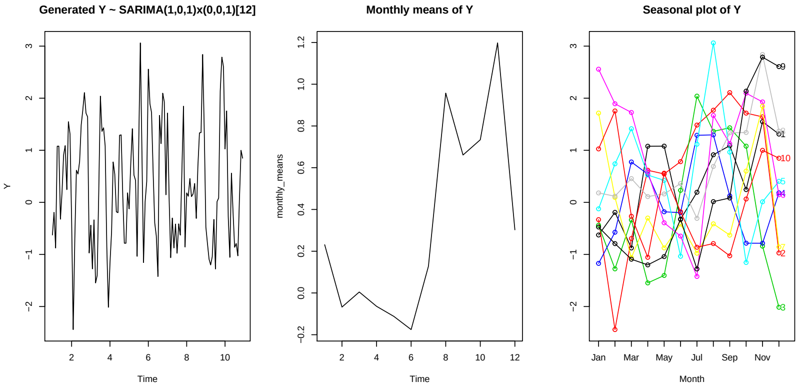
$$\text{Seasonal AR: } \Phi(L^S) = 1 - \Phi_1 L^S - ... - \Phi_p L^{S\cdot P}$$
$$\text{Seasonal MA: } \Theta(L^S) = 1 + \Theta_1 L^S + ... + \Theta_q L^{S\cdot Q}$$

Note that on the left side of equation the seasonal and non-seasonal AR components multiply each other, and on the right side of equation the seasonal and non-seasonal MA components multiply each other.

For example, a $SARIMA(1,0,1)(0,0,1)_{12}$ model could be written:

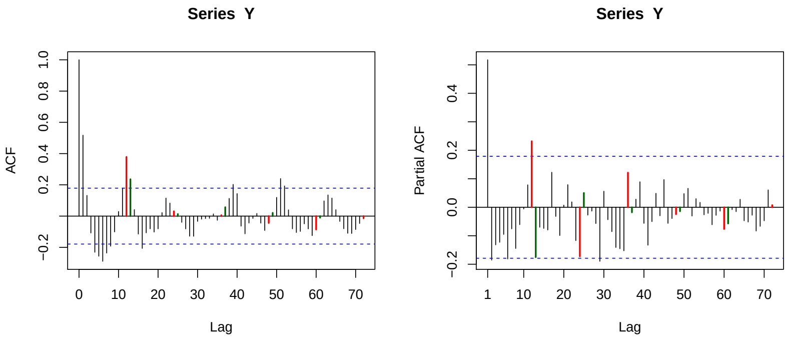
$$(1-\phi L)Y_t = (1+\theta L)\cdot(1+\Theta L^{12})\epsilon_t$$
$$(1-\phi L)Y_t = (1+\theta L+\Theta L^{12}+\theta\Theta L^{12+1})\epsilon_t$$
$$Y_t = \phi Y_{t-1}+\epsilon_t+\theta\epsilon_{t-1}+\Theta\epsilon_{t-12}+\theta\Theta\epsilon_{t-13}$$

where $\phi = 0.4$, $\theta = 0.2$ and $\Theta = 0.5$.



There is seasonality, but no trend.

Examine the ACF and PACF of the data:



Overall, both ACF and PACF plots seem to be declining - a possible $ARMA(1,1)$ model for the non-seasonal model component.

From the ACF plot - the first 12th lag is significant and every other 12th lag (24, 36, etc.) is not (i.e. seasonal cut-off after the first period lag). From the PACF plot - the $12th$, $24th$, $36th$, etc. lags are declining. Also note the $13th$ lag, ϵ_{t-13}. This means that the seasonality could be a $MA(1)$ model.

```r
seas_mdl <- Arima(Y,
      order = c(1, 0, 1),
      seasonal = list(order = c(0, 0, 1), period = 12),
      include.mean = FALSE)
seas_mdl
```

```
## Series: Y
## ARIMA(1,0,1)(0,0,1)[12] with zero mean
##
## Coefficients:
##          ar1     ma1    sma1
##       0.4148  0.1870  0.4802
## s.e.  0.1369  0.1432  0.0902
##
## sigma^2 estimated as 0.7888:  log likelihood=-156.28
## AIC=320.56   AICc=320.91   BIC=331.71
```

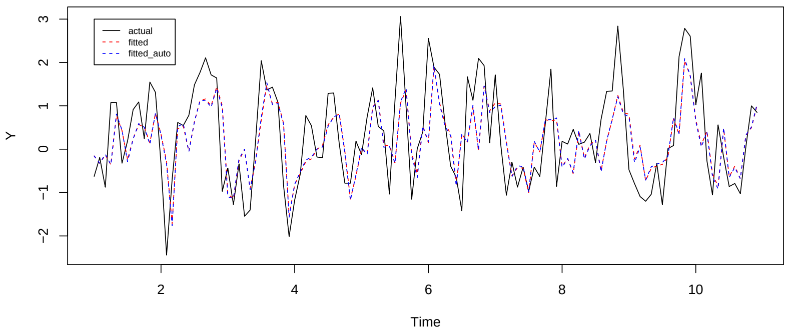
Our estimated model coefficients are: $\hat{\phi} = 0.4919$, $\hat{\theta} = 0.2058$ and $\hat{\Theta} = 0.4788$. Note `Y` is a `ts()` object, i.e. `Y <- ts(Y, freq = 12)`.

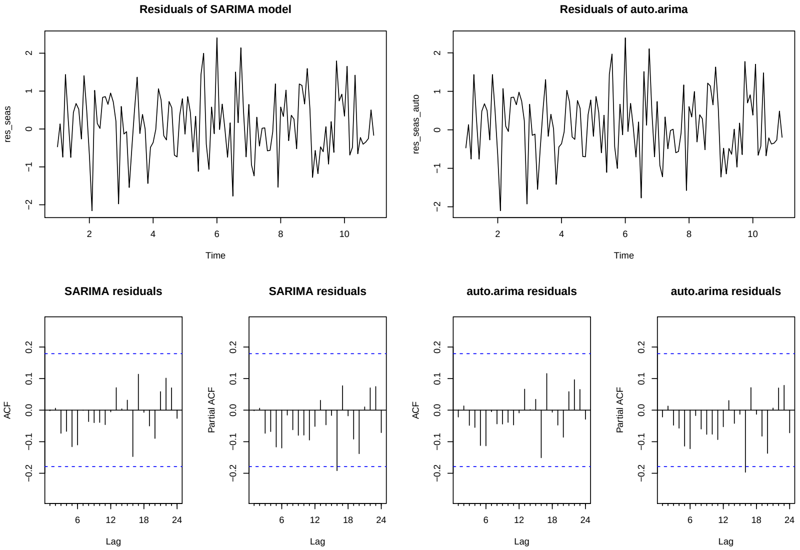
In comparison, the `auto.arima` suggests a slightly different ARMA model:

```
capture.output(summary(seas_mdl_auto <- auto.arima(Y)))[2]
```

```
## [1] "ARIMA(2,0,0)(0,0,1)[12] with zero mean "
```

```
plot.ts(Y, lwd = 1)
lines(fitted(seas_mdl), col = "red", lty = 2)
lines(fitted(seas_mdl_auto), col = "blue", lty = 2)
legend(x = 1, y = 3, c("actual", "fitted", "fitted_auto"),
  col = c("black", "red", "blue"), lty = c(1, 2, 2), cex = 0.7)
```





From the ACF and PACF plots the manually specified $SARIMA(1,0,1)(0,0,1)_{12}$ model residuals are very close to the $SARIMA(2,0,0)(0,0,1)_{12}$ residuals from the `auto.arima` function.

Example: $SARIMA(0,0,1)(0,0,1)_{12}$ model

$$Y_t = (1+\theta L)\cdot(1+\Theta L^{12})\epsilon_t \iff Y_t = (1+\theta L + \Theta L^{12} + \theta\Theta L^{12+1})\epsilon_t$$
$$\iff Y_t = \epsilon_t + \theta\epsilon_{t-1} + \Theta\epsilon_{t-12} + \theta\Theta\epsilon_{t-13}$$

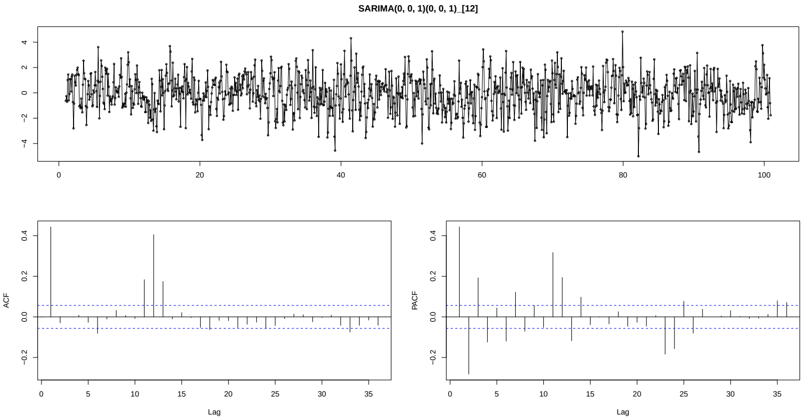
where $\theta = 0.7$ and $\Theta = 0.6$.
Note that:

$$\begin{aligned}
Y_{t-11} &= \epsilon_{t-11} + \theta\epsilon_{t-12} + \Theta\epsilon_{t-23} + \theta\Theta\epsilon_{t-24} \\
Y_{t-12} &= \epsilon_{t-12} + \theta\epsilon_{t-13} + \Theta\epsilon_{t-24} + \theta\Theta\epsilon_{t-25} \\
Y_{t-13} &= \epsilon_{t-13} + \theta\epsilon_{t-14} + \Theta\epsilon_{t-25} + \theta\Theta\epsilon_{t-26}
\end{aligned}$$

which means that the covariance between Y_t and Y_{t-11} is non-zero:

$$\begin{aligned}
&\mathbb{C}\mathrm{ov}(Y_t, Y_{t-11}) \\
&\quad = \mathbb{C}\mathrm{ov}\left(\epsilon_t + \theta\epsilon_{t-1} + \Theta\epsilon_{t-12} + \theta\Theta\epsilon_{t-13},\ \epsilon_{t-11} + \theta\epsilon_{t-12} + \Theta\epsilon_{t-23} + \theta\Theta\epsilon_{t-24}\right) \\
&\quad = \mathbb{C}\mathrm{ov}\left(\Theta\epsilon_{t-12},\ \theta\epsilon_{t-12}\right) = \Theta\theta\cdot\sigma^2 \neq 0
\end{aligned}$$

It can also be shown that $\mathbb{C}\mathrm{ov}(Y_t, Y_{t-12}) \neq 0$, $\mathbb{C}\mathrm{ov}(Y_t, Y_{t-13}) \neq 0$ and the remaining terms, like $\mathbb{C}\mathrm{ov}(Y_t, Y_{t-10}) = \mathbb{C}\mathrm{ov}(Y_t, Y_{t-14}) = 0$.

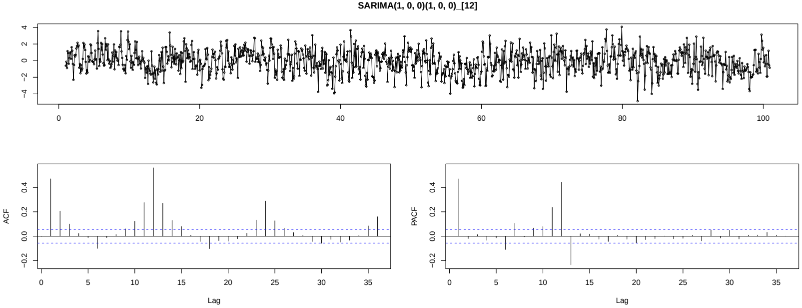


- the spikes at `lags` 1, 11, 12 and 13 in the **ACF**. Remember that the feature of a moving-average process - **ACF** has a sharp cutoff;
- this model has non-seasonal and seasonal *MA* terms, so the PACF tapers nonseasonally, following `lag` 1, and tapers seasonally that is near `lag = 12`, and again near `lag  = 2*12=24` and so on.

Example: $SARIMA(1,0,0)(1,0,0)_{12}$ model

$$(1-\Phi L^{12})(1-\phi L)Y_t = \epsilon_t \iff (1-\phi L - \Phi L^{12} + \phi\Phi L^{13})Y_t = \epsilon_t$$
$$\iff Y_t = \phi Y_{t-1} + \Phi Y_{t-12} - \phi\Phi Y_{t-13} + \epsilon_t$$

where $\phi = 0.6$ and $\Phi = 0.5$.



- there are *distinct* spikes at `lags` 1, 12 and 13 in the **PACF**. Remember that the feature of an autoregressive process - **PACF** has a sharp cutoff.

Revisiting Time Series Smoothing

In some cases, linear regression cannot clarify relationships between variables and cannot detect the trend of a data series. For this reason, we can apply other regression methods in statistics.

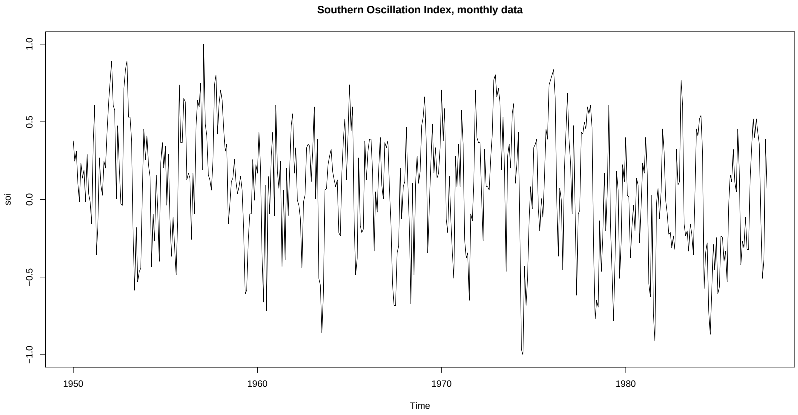
A smoother is a function or procedure for drawing a smooth curve through a scatter diagram. Similarly to linear regression (in which the "curve" is a straight line), the smooth curve is drawn in such a way as to have some desirable properties.

In general, the properties are that:

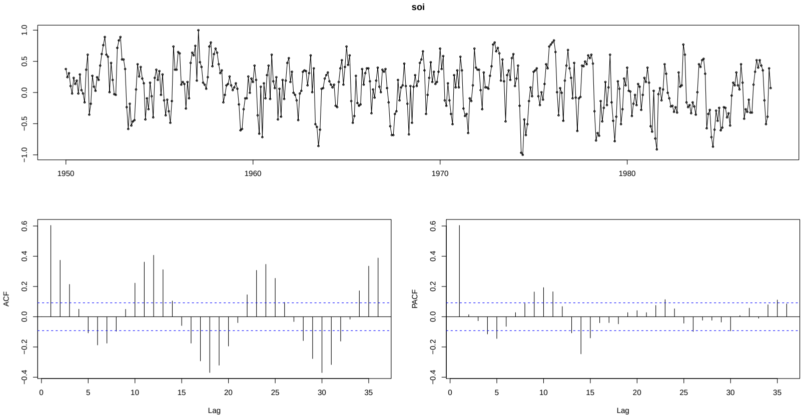
- the curve is indeed smooth;
- locally, the curve minimizes the variance of the residuals, or prediction error.

Depending on the method, we can either attempt to specify the function ourselves to control the degree of smoothing (e.g. moving average, single or double exponential smoothing), or we can estimate the optimal parameters (e.g. Holt-Winters exponential smoothing).

- The SOI measures changes in air pressure, related to sea surface temperatures in the central Pacific Ocean.
- The central Pacific warms every three to seven years due to the El *Niño* effect, which has been blamed for various global extreme weather events.
- Periodic behavior is of interest because underlying processes of interest may be regular and the rate or frequency of oscillation characterizing the behavior of the underlying series would help to identify them.



- Do you notice any trends in the data?
- Do you notice any seasonalities in the data?

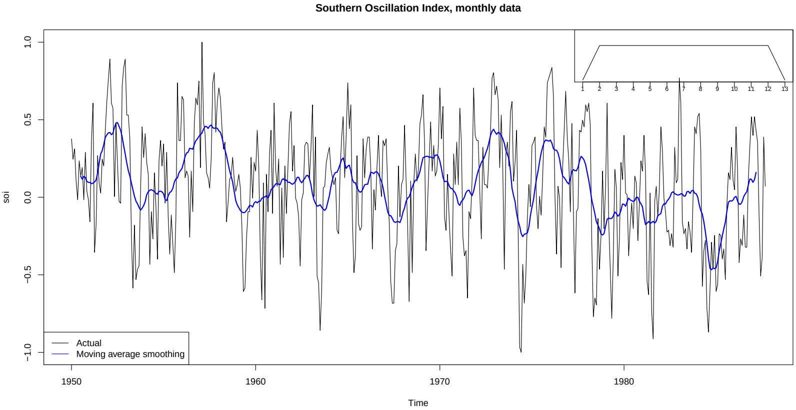


Looking at the above plots, think about the differences between:

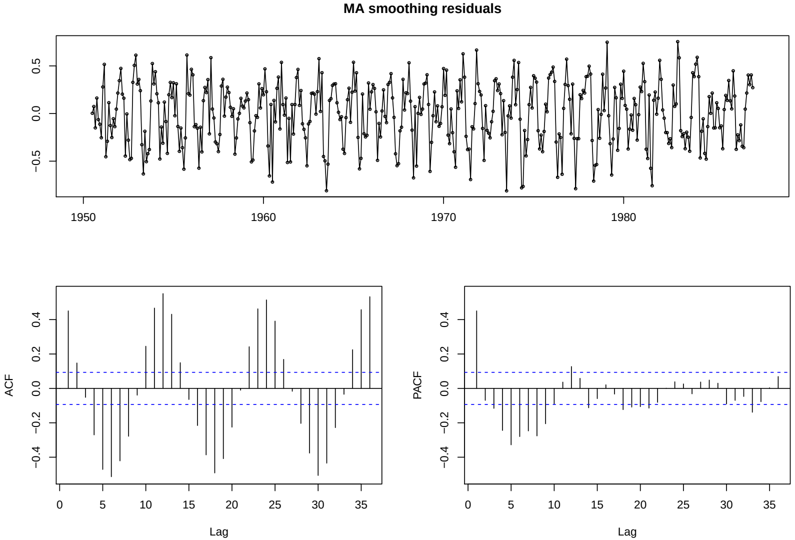
- An autoregressive process with oscillating ACF
  (e.g. $Y_t = 1.5 \cdot Y_{t-1} - 0.9 \cdot Y_{t-2} + \epsilon_t$);
- An ACF of a SARMA model;
- An ACF of a process with a seasonal component;
- An ACF of a process with a Trend component;

Additional Time Series Smoothing Methods

Previously we discussed using a moving average smoothing method, which is useful for **discovering** certain traits in a time series, such as trend and/or seasonal components.



While the moving average seems to work quite well, it appears to be too *choppy*. **We can obtain a smoother fit using the normal distribution for the weights**



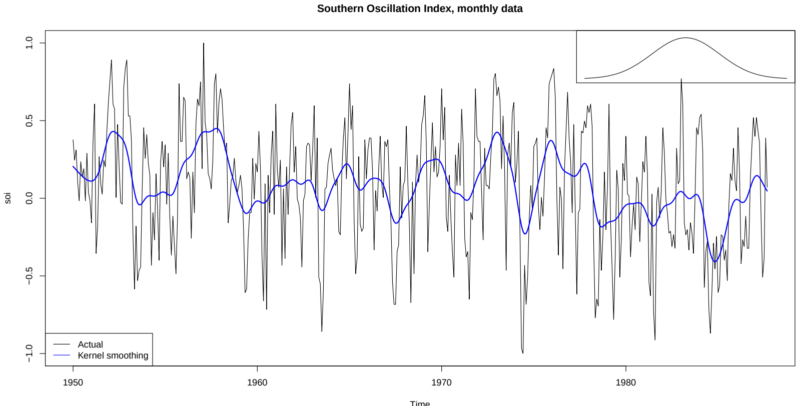
Since we have eliminated the trend component - the seasonality is more pronounced in the residuals.

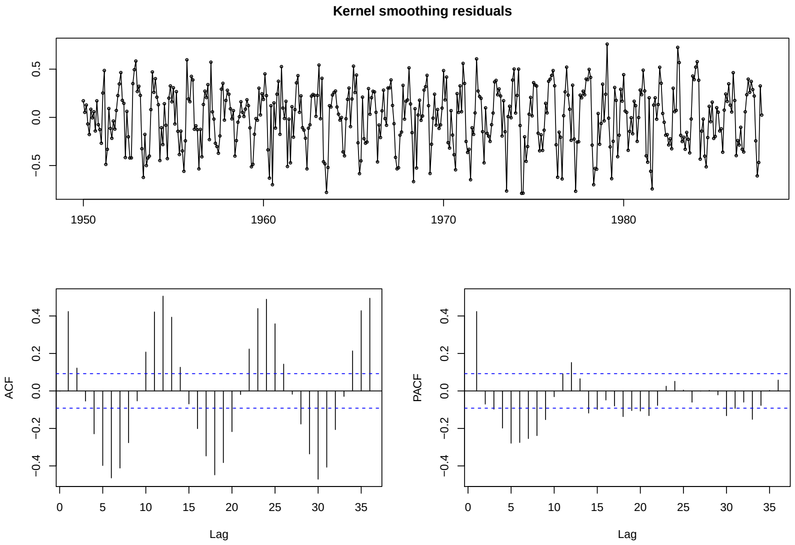
Kernel smoothing

Kernel smoothing - a moving average smoothing method that uses a kernel as the weight function to average the observations.

$$\widehat{T}_t = \sum_{i=1}^{T} w_i(t) Y_i, \quad w_i(t) = K\left(\frac{t-i}{b}\right) \Big/ \sum_{j=1}^{T} K\left(\frac{t-j}{b}\right)$$

where $K(\cdot)$ is a **kernel** function. This estimator is often called the **Nadaraya-Watson estimator**.





Since we have eliminated the trend component - the seasonality is more pronounced in the residuals.

Lowess Smoothing

Another approach to smoothing a time plot is the so called nearest neighbor regression.

- The technique is based on k-nearest neighbors ($k - NN$) regression, where we use only data points in the vicinity of Y_t – namely $Y_{t-k/2}, ..., Y_t, ..., Y_{t+k/2}$ – and predict Y_t via regression, then set $\widehat{T}_t = \widehat{Y}_t$.
- The **lo**cally **we**ighted **s**catterplot **s**moothing (**LOWESS**) method makes no assumptions about the form of the relationship, and allows the form to be discovered using the data itself.

(Note: the next slide will have a general outline of the process - try to get the basic idea behind it.)

The basic idea of LOWESS is close to the nearest neighbor regression (see here and here):

- start with a local polynomial (i.e. $k - NN$) least squares fit and then to use robust methods to obtain the final fit. Specifically, one can first fit a polynomial regression in a neighborhood of Y_t:

$$\frac{1}{T}\sum_{i=1}^{T} W_{k,i}(Y_t)\left(Y_i - \sum_{j=0}^{p}\beta_j Y_t^j\right)^2 \longrightarrow \min_{\beta_0,\dots,\beta_p}$$

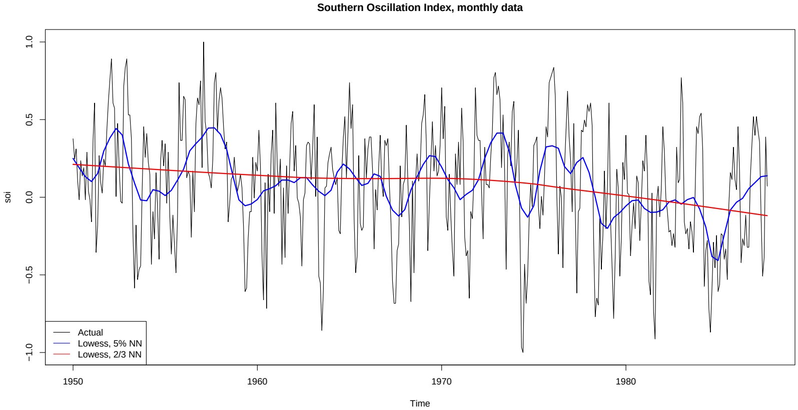
  where $W_{k,i}(Y_t)$ are the $k - NN$ weights:

$$W_{k,i}(Y_t) = Tri\left(\frac{Y_i - Y_t}{h}\right)$$

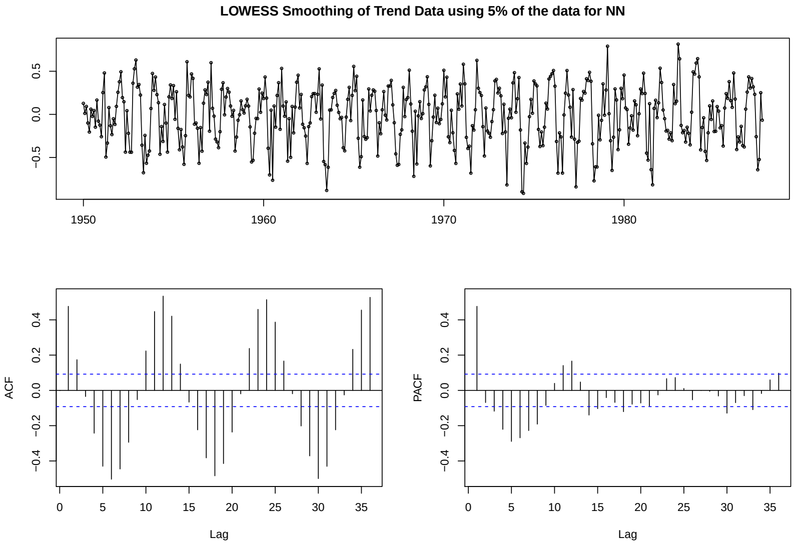
  where h is the k-th smallest distance $|Y_i - Y_t|$, $i = 1, ..., T$ and $Tri(x) = (1 - |x|^3)^3$ if $|x| < 1$ and $Tri = 0$ otherwise.
- Calculate the residuals $\widehat{\epsilon}_t$ and the scale parameter $\widehat{\sigma} = Median(\widehat{\epsilon}_t)$;
- Define robustness weights $\delta_i = K\left(\dfrac{\widehat{\epsilon}_i}{6\widehat{\sigma}}\right)$, where $K(u) = (15/16)(1 - u)^2$, if $|u| \leq 1$ and $K(u) = 0$ otherwise.
- Use $\delta_i W_{k,i}(Y_t)$ instead of $W_{k,i}(Y_t)$ and estimate new parameters $\widehat{\beta}_0, ..., \widehat{\beta}_p$;
- Repeat the process a predefined number of times.

Recommended to use $p = 1$ **for computational efficiency.** Note that we sometimes use f (or λ), which is often expressed as a fraction, or **span** k/T of the total sample T.

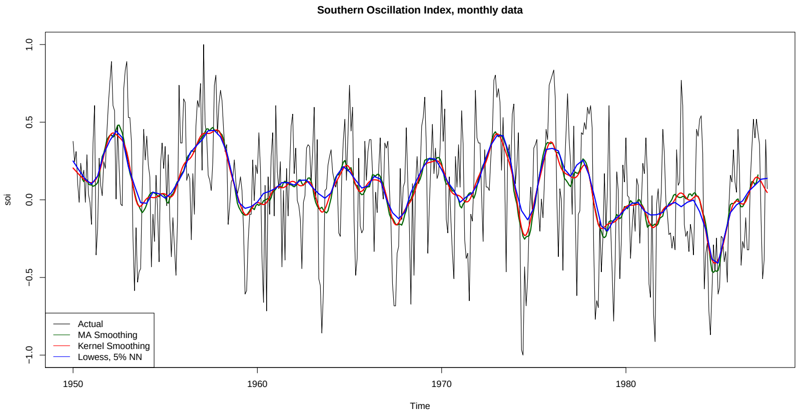
The larger the fraction of nearest neighbors included, the smoother the fit will be.



Insight from the decomposition: a decreasing (or, negative) trend in SOI indicates the long-term warming of the Pacific Ocean.



Since we have eliminated the trend component - the seasonality is more pronounced in the residuals.

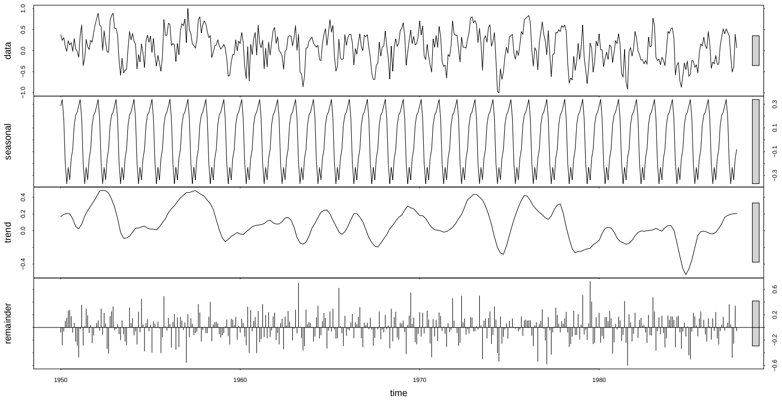


Q: Which method is better?

A: Depends on whether we can decompose seasonality after removing the trend.

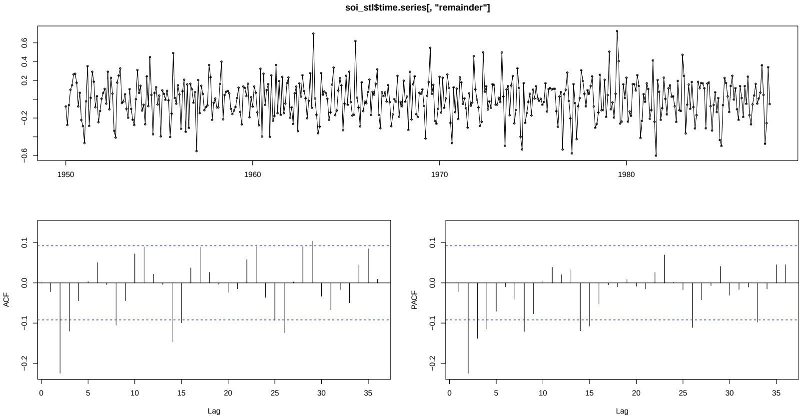
Seasonal Decomposition of Time Series by Loess

To decompose the seasonal component, use the `stl()` function in R. `t.window` controls the 'wiggliness' of the trend components and `s.window` controls the variation on the seasonal component.



It is a very versatile and robust decomposition method - the seasonal component is allowed to change over time, the smoothness of the trend cycle can also be controlled by the user.

This method is robust to outliers, however it is only available for additive decompositions.



- No more seasonality;
- No more trend;

In R the two main robust “user-friendly” methods are:

- `stl()` - does not allow forecasting. To remedy this `forecast::forecast()` can be used, which applies a non-seasonal forecasting method to the seasonally adjusted data and re-seasonalizes using **the last year of the seasonal component**. See here.

- `ets()` - Holt-Winters Exponential smoothing, allows forecasting.

Remarks on LOWESS (LOESS)

Advantages:

- LOWESS does not require specification of a functional form to fit a model to the data sample;
- LOWESS is very flexible, making it ideal for modeling complex processes for which no theoretical models exist.

Disadvantages:

- LOWESS makes less efficient use of data than other least squares methods. It requires fairly large, densely sampled data sets in order to produce good models;
- LOWESS does not produce a regression function that is easily represented by a mathematical formula. This can make it difficult to transfer the results of an analysis to other people;
- LOWESS is a computationally intensive method that is also prone to the effects of outliers in the data set, like other least squares methods.

Local Linear Forecast Using Cubic Splines

Suppose that our time series Y_t, $t = 1, ..., T$ exhibits a non-linear trend. We are interested in forecasting this series by extrapolating the trend using a linear function, which we estimate from the historical data.

An obvious way to smooth data would be to fit a polynomial regression in terms of time. For example, a cubic polynomial would have $Y_t = T_t + E_t$, where

$$T_t = \beta_0 + \beta_1 \cdot t + \beta_2 \cdot t^2 + \beta_3 \cdot t^3$$

In practice we would fit this cubic polynomial on Y_t via OLS to obtain $\widehat{T}_t$.

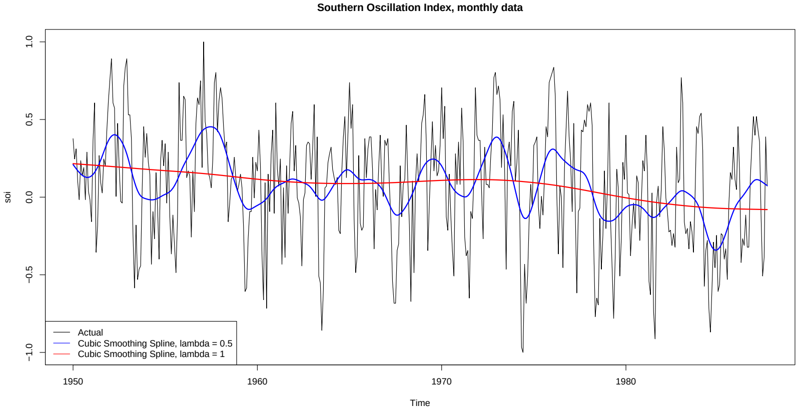
- An extension of polynomial regression is to first divide time $t = 1, ..., T$ into k intervals: $[t_0, t_1], [t_1 + 1, t_2], ..., [t_{k-1} + 1, t_k]$ with $t_0 = 1$ and $t_k = T$. The values $t_0, ..., t_k$ are called **knots**. Each interval fits a polynomial regression, typically of order 3, and this is called **cubic splines**.
- A similar method is called **smoothing splines**, which minimizes a compromise between the fit and the degree of smoothness.

For equally spaced time series, a **cubic smoothing spline** can be defined as the function $\hat{f}(t)$, which minimizes:

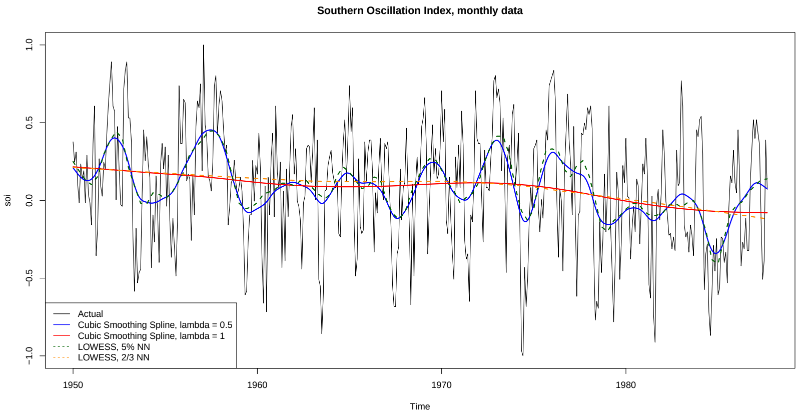
$$\sum_{t=1}^{T}(Y_t - f(t))^2 + \lambda \int_S (f''(u))^2 du$$

over all twice differentiable functions f on S where $[1, T] \subseteq S \subseteq \mathbb{R}$. The smoothing parameter λ is controlling the trade-off between fidelity to the data and roughness of the function estimate. The larger the value of λ, the smoother the fit.

- Link to the paper presenting this method can be found `[here]`.
- The cubic smoothing spline model is equivalent to an $ARIMA(0, 2, 2)$ model (this model will be presented later) but with a restricted parameter space.
- The advantage of the cubic smoothing spline approach over the full $ARIMA$ model is that it provides a smooth historical trend as well as a linear forecast function.

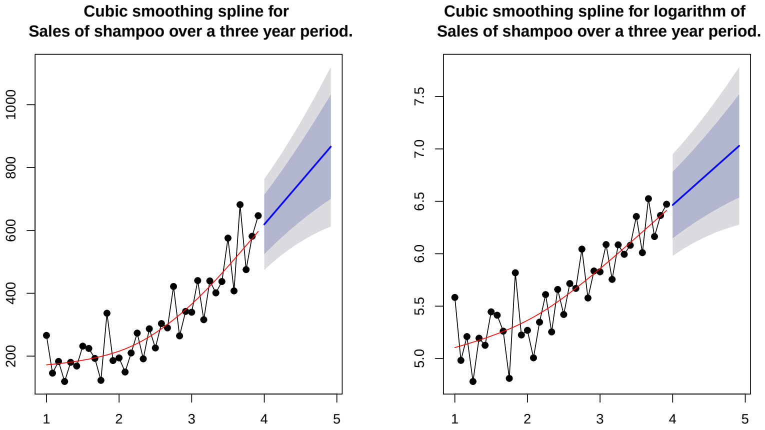


Results appear similar to other trend estimation methods . . .



. . . like the recently introduced LOWESS.

```
data(shampoo)
fcast <- splinef(shampoo, h = 12)
fcast.l <- splinef(log(shampoo), h = 12)
par(mfrow = c(1, 2))
plot(fcast, main = "Cubic smoothing spline for \n Sales of shampoo over a three
plot(fcast.l, main = "Cubic smoothing spline for logarithm of \n Sales of shamp
```



The X-12-ARIMA or X-13-ARIMA-SEATS Seasonal Adjustment

Link to R package documentation [here] and [here].

X-13ARIMA-SEATS is a seasonal adjustment software produced, distributed, and maintained by the United States Census Bureau.

X-13ARIMA-SEATS combines the current filters used in X-12-ARIMA with ARIMA-model-based adjustment as implemented in the program SEATS.

In SEATS, the seasonal and trend filters are estimated simultaneously based on the ARIMA model.

The new program still provides access to all of X-12-ARIMA's seasonal and trend filters and to the diagnostics.

```r
X_13 <- seasonal::seas(x = AirPassengers)
capture.output(summary(X_13))[6:11]
```

```
[1] "                      Estimate Std. Error z value Pr(>|z|)    "
[2] "Weekday             -0.0029497  0.0005232  -5.638 1.72e-08 ***"
[3] "Easter[1]            0.0177674  0.0071580   2.482   0.0131 *  "
[4] "AO1951.May           0.1001558  0.0204387   4.900 9.57e-07 ***"
[5] "MA-Nonseasonal-01    0.1156205  0.0858588   1.347   0.1781    "
[6] "MA-Seasonal-12       0.4973600  0.0774677   6.420 1.36e-10 ***"
```

We can generate a nice `.html` output of our model with:

```r
seasonal::out(X_13)
```

where (using the [documentation, Tables 4.1 and 7.28]):

- `Weekday` - One Coefficient Trading Day, the difference between the number of weekdays and the 2.5 times the number of Saturdays and Sundays
- `AO1951.May` - Additive (point) outlier variable, AO, for the given date or observation number. In this case it is the *regARIMA* (regression model with ARIMA residuals) outlier factor for the point at time 1951-May of the series;
- `Easter[1]` - Easter holiday regression variable for monthly or quarterly flow data which assumes the level of daily activity changes on the `[1]`-st day before Easter and remains at the new level through the day before Easter.
- `MA-Nonseasonal-01` - coefficients of the non-seasonal components of the ARMA model for the *differenced* residuals, $\nabla \epsilon_t$.
- `MA-Seasonal-12` - coefficients of the seasonal components of the ARMA model for the *differenced* residuals $\nabla_{12} \epsilon_t$.

Looking at ?series, we can extract different data:

```r
#Estimate of the Seasonal factors:
X_13.seas <- seasonal::series(X_13, "history.sfestimates")
```

```
## specs have been added to the model: history
```

```r
#Estimate of the seasonally adjusted data
X_13.deseas <- seasonal::series(X_13, "history.saestimates")
```

```
## specs have been added to the model: history
```

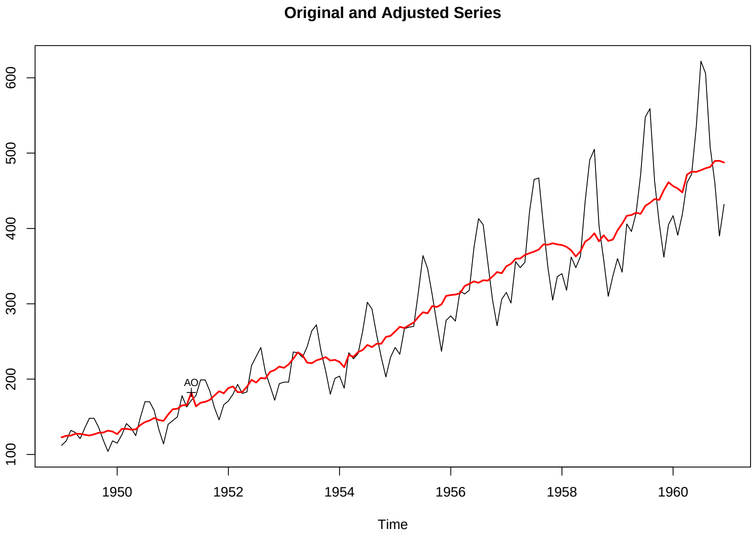
```r
#Estimate of the trend component
X_13.trend <- seasonal::series(X_13, "history.trendestimates")
```

```
## specs have been added to the model: history
```

```r
#Forecasts:
X_13.forc <- seasonal::series(X_13, "forecast.forecasts")
```

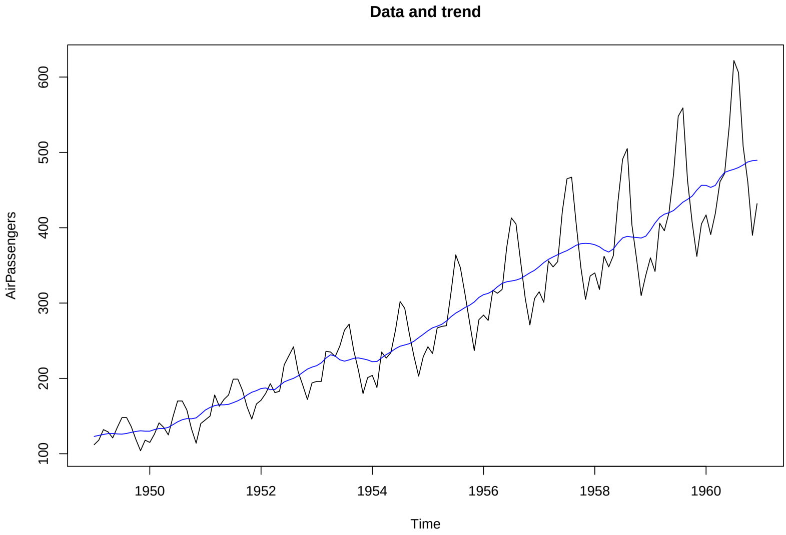
```
## specs have been added to the model: forecast
```

```
plot(X_13)
```



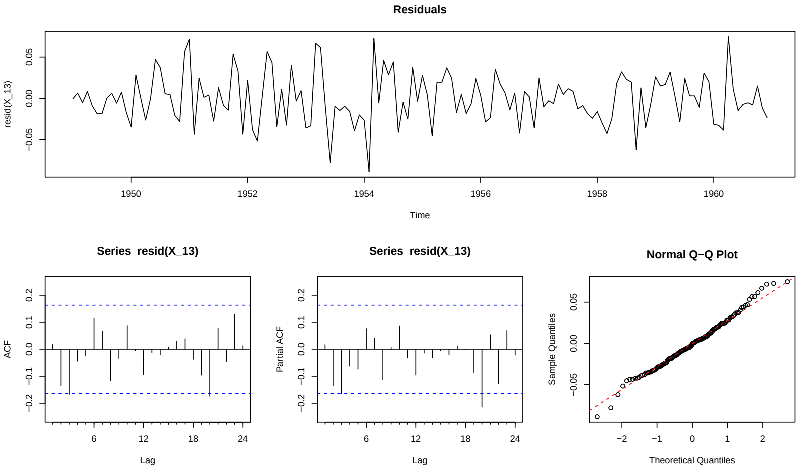
Note: The series is adjusted for seasonality only (i.e. seasonality removed).

```r
plot(AirPassengers, main = "Data and trend")
lines(X_13$data[, "trend"], col = "blue")
```



Note: The blue line is the trend (no random component).

```r
layout(matrix(c(1, 1, 1, 2, 3, 4), 2, 3, byrow = TRUE))
plot.ts(resid(X_13), main = "Residuals")
forecast::Acf(resid(X_13)); forecast::Pacf(resid(X_13))
qqnorm(resid(X_13)); qqline(resid(X_13), lty = 2, col = "red")
```



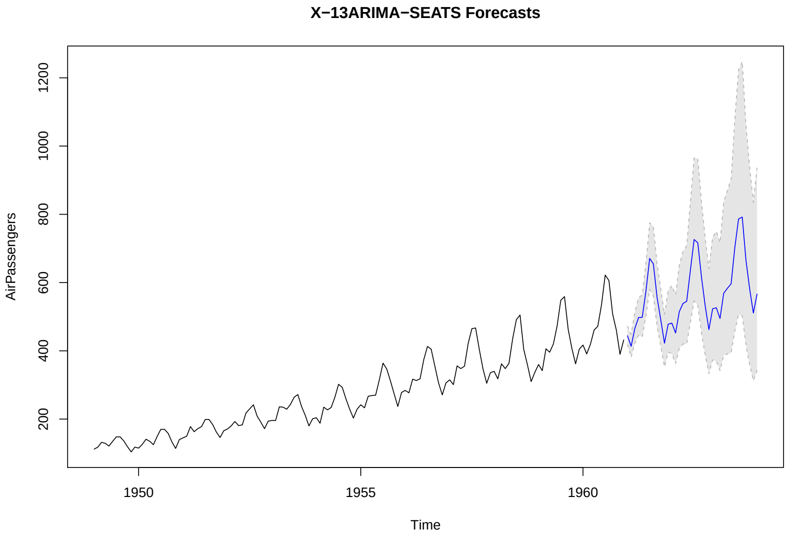
We can also plot the forecasts along with their confidence intervals:

```
#Set the x and y axis separtely
x.lim = c(head(time(AirPassengers), 1), tail(time(X_13.forc), 1))
y.lim = c(min(AirPassengers), max(X_13.forc[,"upperci"]))


#Plot the time series:
plot.ts(AirPassengers,  xlim = x.lim, ylim = y.lim,
        main = "X-13ARIMA-SEATS Forecasts")

#Plot the shaded forecast confidence area:
polygon(c(time(X_13.forc), rev(time(X_13.forc))),
        c(X_13.forc[,"upperci"], rev(X_13.forc[,"lowerci"])),
        col = "grey90", border = NA)


#Plot the forecasts along with their lower and upper bounds:
lines(X_13.forc[,"forecast"], col = "blue")
lines(X_13.forc[,"lowerci"], col = "grey70", lty = 2)
lines(X_13.forc[,"upperci"], col = "grey70", lty = 2)
```


X−13ARIMA−SEATS Forecasts
AirPassengers
Time
200
400
600
800
1000
1200
1950
1955
1960

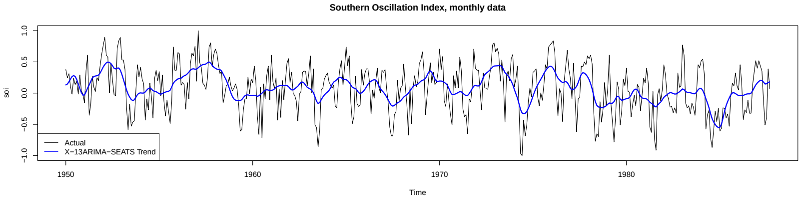
Looking back at our SOI data example:

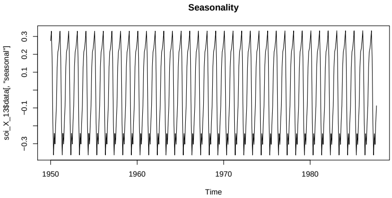
```
soi_X_13 <- seasonal::seas(x = soi)
```

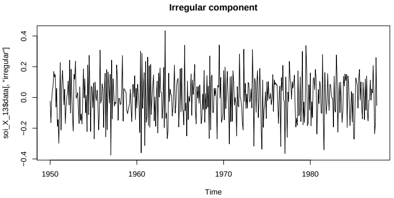
It may sometimes be the case that `seasonal::series(...)` will not work, so the relevant results can always be extracted directly.

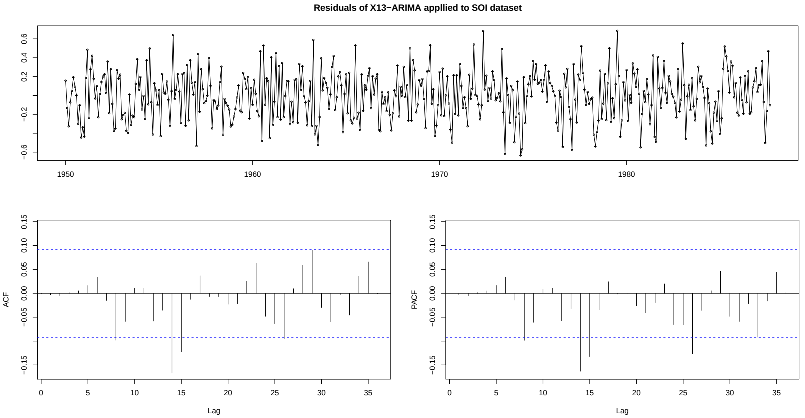
```
print(head(soi_X_13$data))
```

```
               final   seasonal seasonaladj     trend   irregular  adjustfac
Jan 1950  0.10145039  0.2755496  0.10145039 0.1316089 -0.02457600  0.2755496
Feb 1950 -0.08329878  0.3292988 -0.08329878 0.1452373 -0.16458940  0.3292988
Mar 1950  0.13144671  0.1795533  0.13144671 0.1808291 -0.02679486  0.1795533
Apr 1950  0.28609804 -0.1820980  0.28609804 0.2264334  0.04469685 -0.1820980
May 1950  0.34707990 -0.3630799  0.34707990 0.2641907  0.07335916 -0.3630799
Jun 1950  0.47731379 -0.2423138  0.47731379 0.2810873  0.17002810 -0.2423138
```


Southern Oscillation Index, monthly data
soi
1.0
0.5
0.0
−0.5
−1.0
Actual
X−13ARIMA−SEATS Trend
1950
1960
1970
1980
Time


Seasonality
soi_X_13$data[, "seasonal"]
0.3
0.2
0.1
−0.1
−0.3
1950
1960
1970
1980
Time


Irregular component
soi_X_13$data[, "irregular"]
0.4
0.2
0.0
−0.2
−0.4
1950
1960
1970
1980
Time



Note: In this case, we are interested in the forecasts of airplane passengers - not just the forecasts of the random component, but the trend and seasonality as well - after all, in most empirical applications we want our forecasts to be for the same **original series**.

Recap: Seasonality

- Seasonality in a time series is a regular pattern, which repeats over d time units, where d is called the (seasonality) period - the number of time units/periods, until the pattern repeats again. This can be written as $S_t = S_{t+d}$;

- Seasonality usually causes the series to be nonstationary, since the average values from particular seasonal periods (e.g. summer) may be different than the average values at other times (e.g. winter);
  - We can remove the trend by differencing the series, e.g. by transforming the series to $(1 - L)Y_t$;
  - We can remove the seasonality by seasonally differencing the series, e.g. by transforming the series to $(1 - L^d)Y_t$;
  - To remove both trend and seasonality by applying both non-seasonal and seasonal differencing, e.g. $(1 - L)(1 - L^d)Y_t$;

- The seasonal ARMA model incorporates both non-seasonal and seasonal factors in a multiplicative model. In a seasonal ARMA model, the seasonal AR and MA terms predict Y_t using lagged values, where the lags are multiples of d.

Recap: Time Series Decomposition

- Decomposition procedures are used in time series to describe the trend and seasonal factors in a time series. More extensive decompositions might also include long-run cycles, holiday effects, day of week effects and so on (see $X - 13ARIMA - SEATS$).
- Decomposition can be used to estimate and remove seasonality in order to calculate seasonally adjusted values. These adjusted values can then be used to analyse the trend more clearly. For instance, U.S. unemployment tends to decrease in the summer due to increased employment in agricultural areas. So, it would appear that unemployment decreased from winter to summer, however, this does not indicate that there is a trend toward lower unemployment in the country.
- Decomposition is usually done in three steps: (i) trend estimation; (ii) seasonality estimation on the de-trended series; (ii) remainder (i.e. random) component estimation by removing the seasonal and trend component from the original series. The decomposition method depends on whether the series is additive or multiplicative.
- The random component can be analyzed for such things as the mean, variance, or possibly even for whether the component is actually truly random (WN) or might be modeled with an ARIMA model.

Recap: Time Series Smoothing

- Smoothing is usually done to help us better see patterns in the time series. The term **filter** is sometimes used to describe a smoothing procedure.
  - For Series with a trend, we may smooth out the irregular roughness to see a clearer signal.
    - For non-seasonal data you should experiment with moving averages of different spans. Those spans of time could be relatively short. The objective is to knock off the rough edges to see what trend or pattern might be there.
  - For seasonal data, we might smooth out the seasonality so that we can identify the trend.
    - To take away seasonality from a series, so we can better see the trend, we would use a moving average with a length equal to d (the seasonal length). If d is even, then a centered moving average is needed.
- Some possible smoothing methods include - moving average smoothing; single, double or triple (Holt-Winters) exponential smoothing. Additional methods include kernel smoothing, lo(w)ess smoothing, cubic smoothing splines and more.
- Smoothing doesn't provide us with a model, but it can be a good first step in describing various components of the series.