MONASH University

MONASH BUSINESS SCHOOL

# ETW3420: Principles of Forecasting and Applications



Topic 4: Time Series Decomposition

Dr. Jason Ng

# Outline

# Introduction



- Time series data can exhibit a huge variety of patterns and it is helpful to categorize some of the patterns and behaviours that can be seen in time series.
- It is also sometimes useful to try to split a time series into several components, each representing one of the underlying categories of pattern.
- In this topic, we consider some common patterns and methods to extract the associated components from a time series.
- Often this is done to help understand the time series better, but it can also be used to improve forecasts.

# Outline

# Time series patterns

**Recall**



**Trend** pattern exists when there is a long-term increase or decrease in the data.

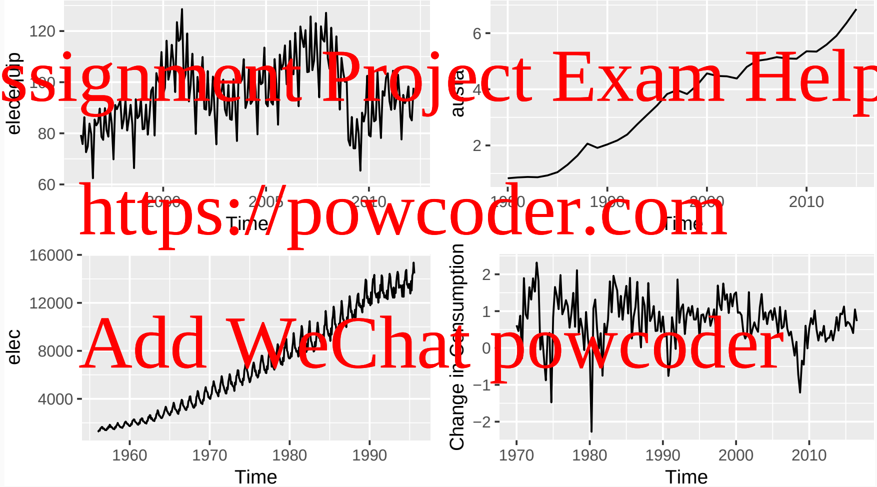


**Cyclic** pattern exists when data exhibit rises and falls that are *not of fixed period* (duration usually of at least 2 years).



**Seasonal** pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week).

# Time series patterns



# Time series decomposition



$$y_t = f(S_t, T_t, R_t)$$



| | | |
|---|---|---|
| where | $y_t =$ | data at period $t$ |
| | $T_t =$ | trend-cycle component at period $t$ |
| | $S_t =$ | seasonal component at period $t$ |
| | $R_t =$ | remainder component at period $t$ |

## Time series decomposition



$$y_t = f(S_t, T_t, R_t)$$



| | | |
|---|---|---|
| where | $y_t =$ | data at period $t$ |
| | $T_t =$ | trend-cycle component at period $t$ |
| | $S_t =$ | seasonal component at period $t$ |
| | $R_t =$ | remainder component at period $t$ |



**Additive decomposition:** $y_t = S_t + T_t + R_t$.

**Multiplicative decomposition:** $y_t = S_t \times T_t \times R_t$.

## Time series decomposition

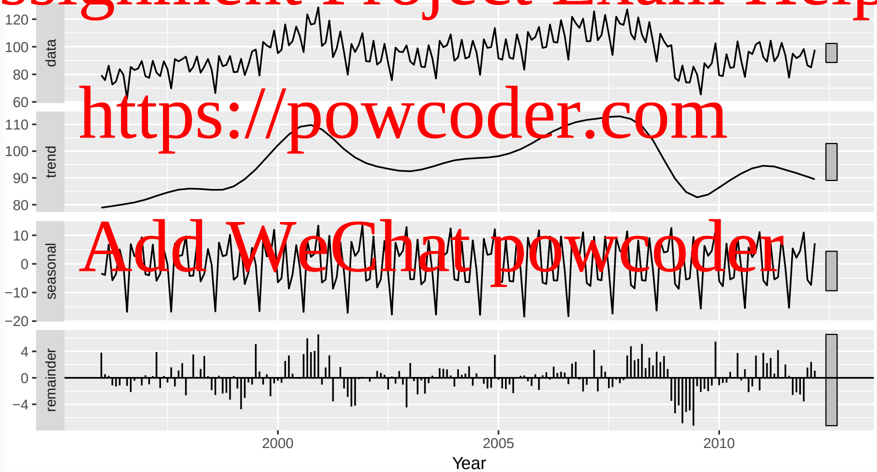
- Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.
- If seasonal are proportional to level of series, then multiplicative model appropriate.
- Multiplicative decomposition more prevalent with economic series
- Alternative: use a Box-Cox transformation, and then use additive decomposition.
- Logs turn multiplicative relationship into an additive relationship:

$$y_t = S_t \times T_t \times E_t \quad \Rightarrow \quad \log y_t = \log S_t + \log T_t + \log R_t.$$

# Euro electrical equipment

```r
fit <- stl(elecequip, s.window=7)
autoplot(fit) + xlab("Year")
```

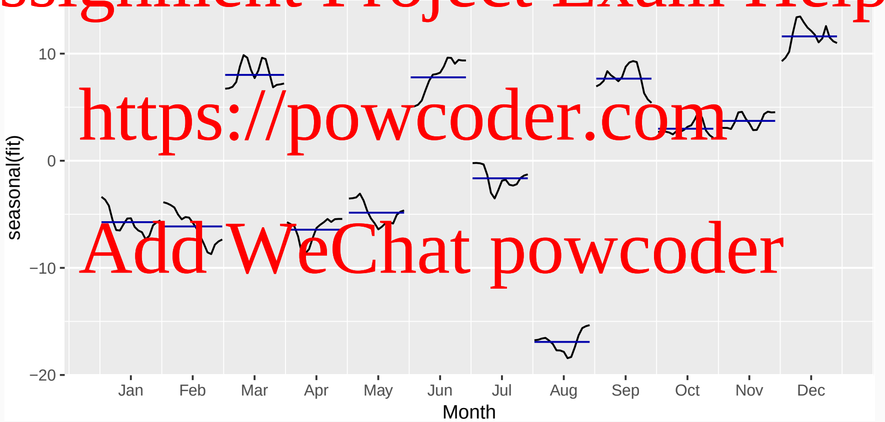




# Euro electrical equipment

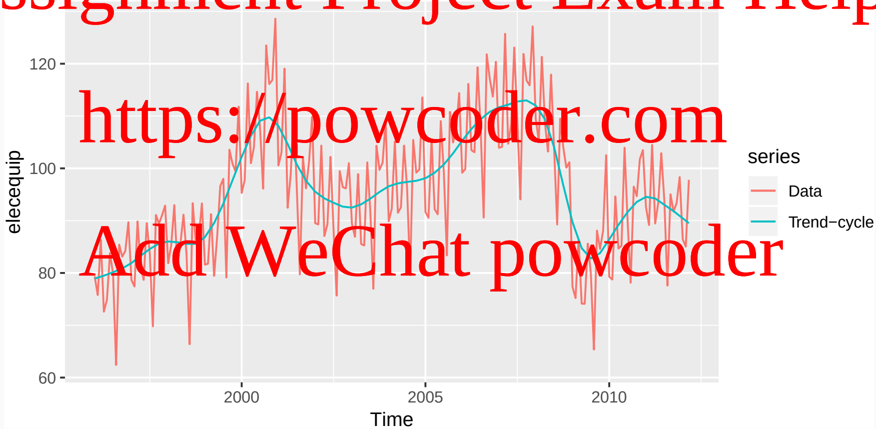
```
ggsubseriesplot(seasonal(fit))
```





# Euro electrical equipment

```
autoplot(elecequip, series="Data") +
  autolayer(trendcycle(fit), series="Trend-cycle")
```



# Helper functions



- `seasonal()` extracts the seasonal component
- `trendcycle()` extracts the trend-cycle component
- `remainder()` extracts the remainder component.
- `seasadj()` returns the seasonally adjusted series.

# Self-Practice



Repeat the decomposition using



```
elecequip %>%
  stl(s.window=7, t.window=11) %>%
  autoplot()
```



What happens as you change `s.window` and `t.window`?

# Outline

## Seasonal adjustment



- Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.
- Additive decomposition: seasonally adjusted data given by



$$y_t - S_t = T_t + R_t$$

- Multiplicative decomposition: seasonally adjusted data given by

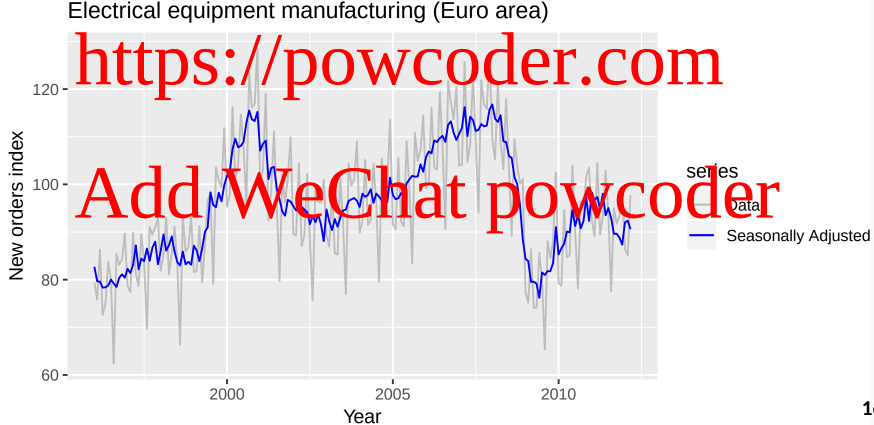


$$y_t / S_t = T_t \times R_t$$

# Euro electrical equipment

```
fit <- stl(elecequip, s.window=7)
autoplot(elecequip, series="Data") +
  autolayer(seasadj(fit), series="Seasonally Adjusted")
```





## Seasonal adjustment



- We use estimates of *S* based on past values to seasonally adjust a current value.



- Seasonally adjusted series reflect **remainders** as well as **trend**. Therefore they are not “smooth”" and “downturns”" or “upturns” can be misleading.



- It is better to use the trend-cycle component to look for turning points.

# Outline

## History of time series decomposition

- Classical method originated in 1920s.
- Census II method introduced in 1957. Basis for X-11 method and variants (including X-12-ARIMA, X-13-ARIMA)
- STL method introduced in 1983
- TRAMO/SEATS introduced in 1990s.

# History of time series decomposition

- Classical method originated in 1920s.
- Census II method introduced in 1957. Basis for X-11 method and variants (including X-12-ARIMA, X-13-ARIMA)
- STL method introduced in 1983
- TRAMO/SEATS introduced in 1990s.





## National Statistics Offices

- ABS uses X-12-ARIMA
- US Census Bureau uses X-13-ARIMA-SEATS
- Statistics Canada uses X-12-ARIMA
- ONS (UK) uses X-12-ARIMA
- EuroStat use X-13-ARIMA-SEATS
- Department of Statistics Malaysia ????

# Outline

# Classical Decomposition



- The classical method of time series decompostion originated in the 1920s and was widely used until the 1950s.



- It still forms the basis of many time series decomposition methods, so it is important to know how it works.
- The first step in a classical decomposition is to use a moving average method to estimate the trend-cycle.



- Thus, this section discusses moving averages.

# Moving average smoothing

- A moving average of order $m$ (i.e. $m$-MA) can be written as



$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j}$$

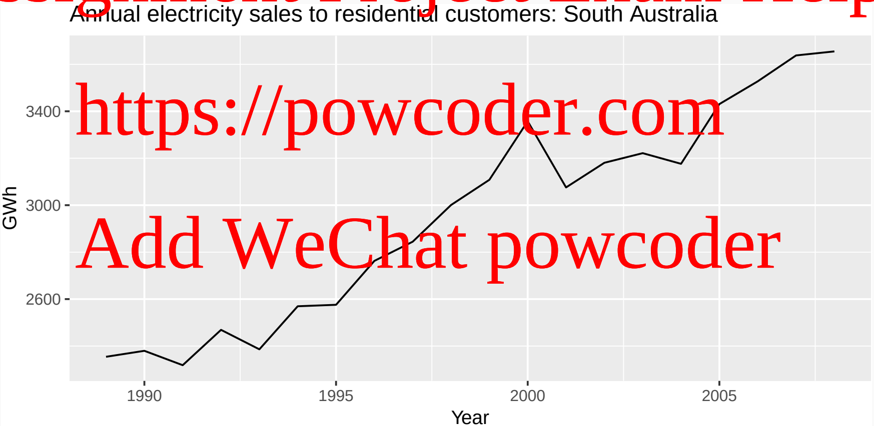
where $m = 2k + 1$.



- The estimate of the trend-cycle at time $t$ is obtained by averaging values of the time series within $k$ periods of $t$.
- Observations that are nearby in time are also likely to be close in value.
- Therefore, the average eliminates some of the randomness in the data, leaving a smooth trend-cycle component.

# Moving average smoothing: Example

```
autoplot(elecsales) + xlab("Year") + ylab("GWh") +
  ggtitle("Annual electricity sales to residential custo
```



# Moving average smoothing: Example

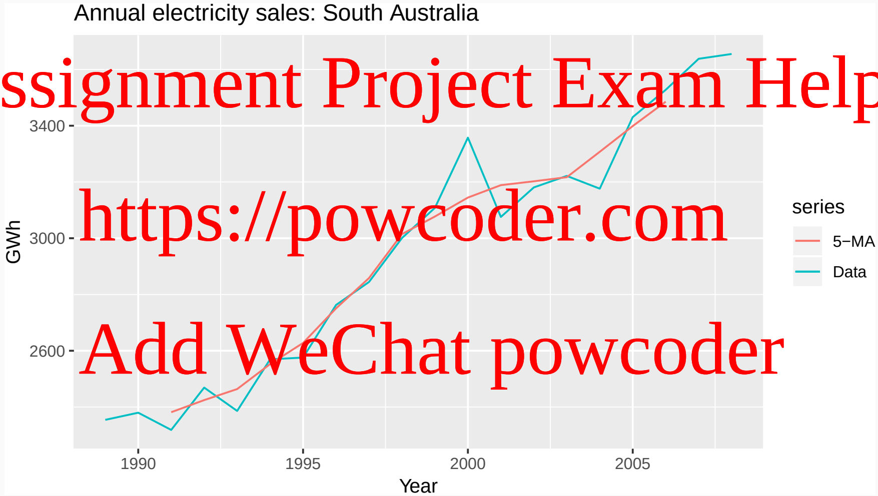
| Year | Sales | X5.MA |
|---|---|---|
| 1989 | 2354.34 | NA |
| 1990 | 2379.71 | NA |
| 1991 | 2318.52 | 2381.530 |
| 1992 | 2468.99 | 2424.556 |
| 1993 | 2386.09 | 2463.758 |

| | Year | Sales | X5.MA |
|---|---|---|---|
| 16 | 2004 | 3176.20 | 3307.296 |
| 17 | 2005 | 3430.60 | 3398.754 |
| 18 | 2006 | 3527.48 | 3485.434 |
| 19 | 2007 | 3637.89 | NA |
| 20 | 2008 | 3655.00 | NA |

# Moving average smoothing: Example

- The last column of the table provides an estimate of the trend cycle using a 5-MA. (i.e. $m = 5$, $k = 2$)
- The first value in this column is the average of the first five observations (1989-1993); the second value in the 5-MA column is the average of the values for 1990-1994; and so on.
- Each value in the 5-MA column is the average of the observations in the five year window centred on the corresponding year.
- The MA values are easily computed using `ma(elecsales, 5)`.
- There are no values for either the first two years or the last two years, because we do not have two observations on either side.

# Moving average smoothing: Example

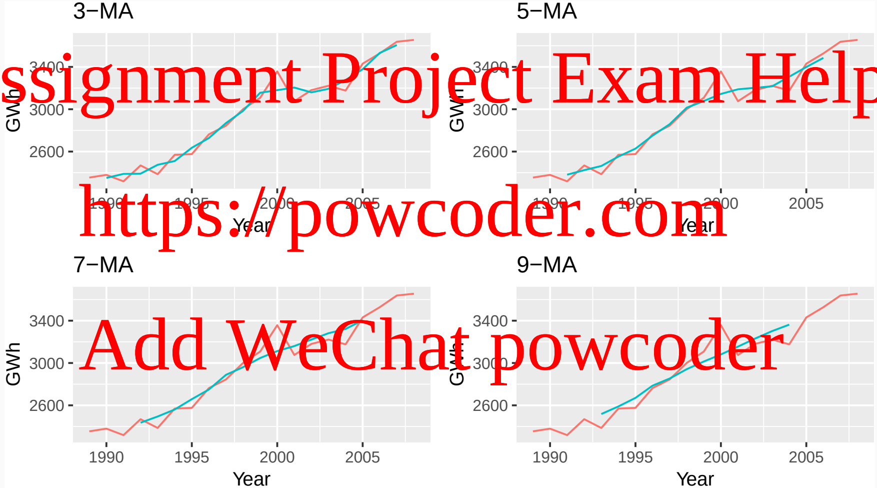


# Moving average smoothing: Example



- Notice that the trend-cycle (in red) is smoother than the original data and captures the **main movement** of the time series without all of the minor fluctuations.
- The order of the MA determines the **smoothness** of the trend-cycle estimate.
- Larger order $\Rightarrow$ smoother curve.

# Moving average smoothing: Example



# Moving average smoothing



- Simple moving averages such as the above are usually of an odd order (e.g. 3, 5, 7, etc.).



- This is so that they are symmetric: in a moving average of order $m = 2k + 1$, the middle observation, and $k$ observations on either side, are averaged.



- If $m$ was even $\Rightarrow$ no longer symmetric.

# Moving averages of moving averages



- If $m$ is even, a moving average has to be applied to a moving average to make an even-order moving average symmetric.
- For example, we might take a MA of order 4, and then apply another MA of order 2 to the results.

# Moving averages of moving averages: Example

```
beer2 <- window(ausbeer, start = 1992)
ma4 <- ma(beer2, order = 4, centre = F)
ma2x4 <- ma(beer2, order = 4, centre = T)
```

```
##          Observation   4-MA  2x4-MA
## 1992 Q1          443     NA      NA
## 1992 Q2          410 451.25      NA
## 1992 Q3          420 448.75 450.000
## 1992 Q4          532 451.50 450.125
## 1993 Q1          433 449.00 450.250
## 1993 Q2          421 444.00 446.500
## 1993 Q3          410 448.00 446.000
## 1993 Q4          512 438.00 443.000
## 1994 Q1          449 441.25 439.625
```

# Moving averages of moving averages: Example



- The notation 2 x 4-MA in the last column means a 4-MA followed by a 2-MA.
- The values in the last column are obtained by taking a moving average of order 2 of the values in the previous column.
- For example, the first two values in the 4-MA column are



$$451.25 = \frac{(443 + 410 + 420 + 532)}{4}$$

and



$$448.75 = \frac{(410 + 420 + 532 + 433)}{4}$$

# Moving averages of moving averages: Example



- The first value in the 2x4-MA column is the average of these two:



$$450.00 = \frac{(451.25 + 448.75)}{2}$$

## Moving averages of moving averages

- When a 2-MA follows a moving average of an even order (such as 4), it is called a centred moving average of order 4.
- This is because the results are now symmetric. To see that this is the case, we can write the $2 \times 4$-MA as follows:

$$\hat{T}_t = \frac{1}{2}\Big[\frac{1}{4}(y_{t-2} + y_{t-1} + y_t + y_{t+1}) + \frac{1}{4}(y_{t-1} + y_t + y_{t+1} + y_{t+2})\Big]$$

$$= \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}$$

- Note that the $2 \times m$-MA is now equivalent to a weighted moving average of order $m + 1$ where all observations take the weight $1/m$, except for the first and last terms which take weights $1/(2m)$. In our example, $m = 4$.

# Moving averages of moving averages: Application

- The most common application of CMA is for estimating the **trend-cycle from seasonal data**.
- Consider the $2 \times 4$-MA:

$$\hat{T}_t = \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}$$

- When applied to quarterly data, each quarter of the year is given equal weight as the first and last terms apply to the same quarter in consecutive years.
- Consequently, the seasonal variation will be averaged out and the resulting values of $\hat{T}_t$ will have little or no seasonal variation remaining.

# Moving averages of moving averages: Application



- If the seasonal period is even and of order $m$, we use a $2 \times m$-MA to estimate the trend-cycle.
- If the seasonal period is odd and of order $m$, we use a $m$-MA to estimate the trend-cycle.



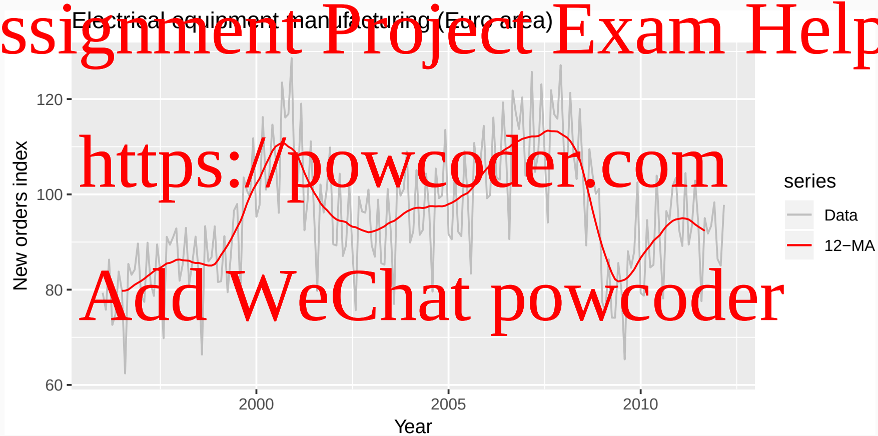
- For example, a $2 \times 12$-MA can be used to estimate the trend-cycle of **monthly data**; a 7-MA can be used to estimate the trend-cycle of **daily data** with a weekly seasonality.



- Other choices for the order of the MA will usually result in trend-cycle estimates being contaminated by seasonality in the data.

# Example: 2 x 12 MA for Electrical Equipment Manufacturing





# Outline

# Classical Decomposition

- It is a relatively simple procedure, and forms the starting point for most other methods of time series decomposition.
- Two forms of classical decomposition: an additive decomposition and a multiplicative decomposition.
- We will describe them for a time series with seasonal period *m*.
- The seasonal component is assumed constant from year to year.
- For multiplicative seasonality, the *m* values that form the seasonal component are sometimes called the seasonal indices.

## Additive Decomposition

**Step 1**

- If $m$ is an even number, compute the trend-cycle component $\hat{T}_t$ using a $2 \times m$-MA. If $m$ is an odd number, compute the trend-cycle component $\hat{T}_t$ using an $m$-MA.

**Step 2**

- Calculate the detrended series: $y_t - \hat{T}_t$

**Step 3**

- To estimate the seasonal component for each season, simply average the detrended values for that season. For example, with monthly data, the seasonal component for March is the average of all the detrended March values in the data. These seasonal component values are then adjusted to ensure that they add to zero.

# Additive Decomposition



**Step 3 (Continued)**

- The seasonal component is obtained by stringing together these monthly values, and then replicating the sequence for each year of data. This gives $\hat{S}_t$.

**Step 4**

- The remainder component is calculated by subtracting the estimated seasonal and trend-cycle components: $\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$.

# Multiplicative Decomposition



- Similar with additive, except:

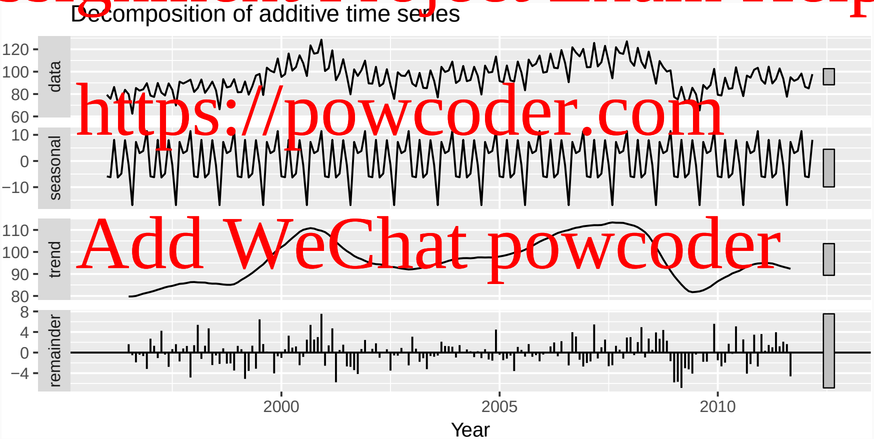


  - Subtractions from Step 2 and Step 4 are replaced by divisions;
  - In Step 3, the seasonal indexes are adjusted to ensure that they add to $m$.

# Euro Electrical Equipment

```r
fit <- decompose(elecequip, type = "additive")
autoplot(fit) + xlab("Year")
```

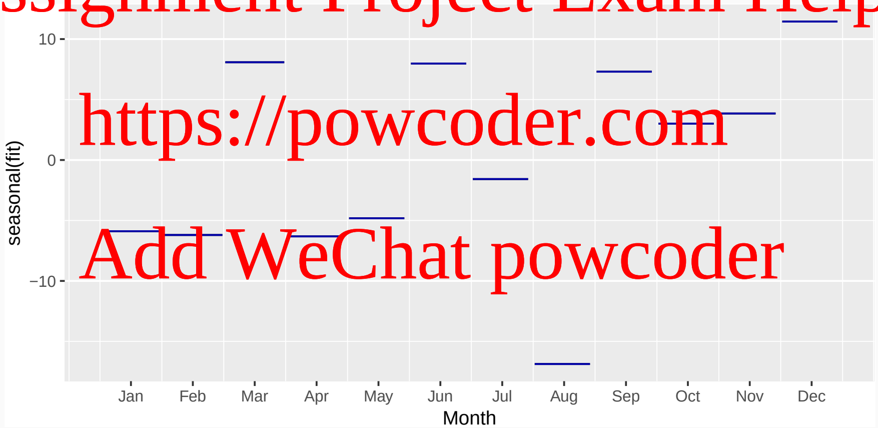




# Euro Electrical Equipment

```
ggsubseriesplot(seasonal(fit))
```

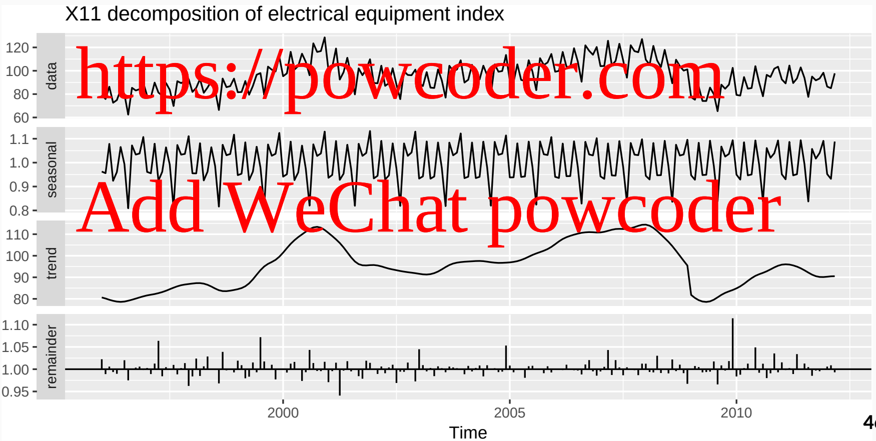




# X-11 decomposition

```
library(seasonal)
fit <- seas(elecequip, x11="")
autoplot(fit)
```





# (Dis)advantages of X-11

**Advantages**

- Relatively robust to outliers
- Completely automated choices for trend and seasonal changes
- Very widely tested on economic data over a long period of time.

# (Dis)advantages of X-11

**Advantages**

- Relatively robust to outliers
- Completely automated choices for trend and seasonal changes
- Very widely tested on economic data over a long period of time.

**Disadvantages**

- No prediction/confidence intervals
- Ad hoc method with no underlying model
- Only developed for quarterly and monthly data

# Extensions: X-12-ARIMA and X-13-ARIMA



- The X-11, X-12-ARIMA and X-13-ARIMA methods are based on Census II decomposition.
- These allow adjustments for trading days and other explanatory variables.



- Known outliers can be omitted.
- Level shifts and ramp effects can be modelled.
- Missing values estimated and replaced.
- Holiday factors (e.g., Easter, Labour Day) can be estimated.

# Outline

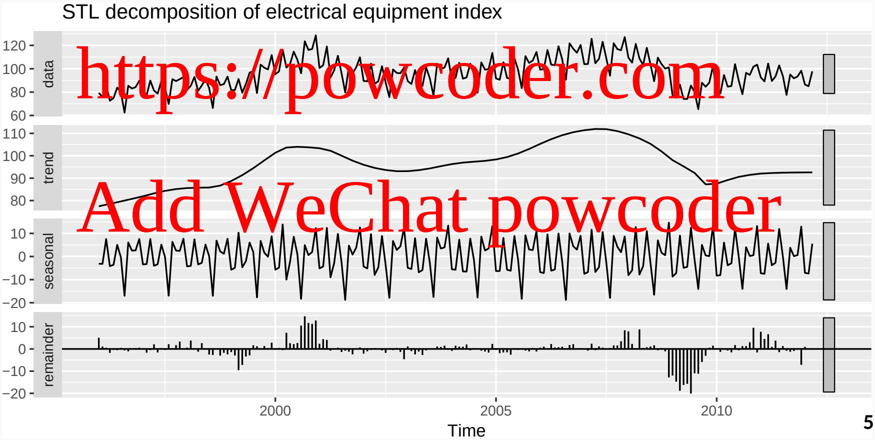
# STL decomposition

- STL: “Seasonal and Trend decomposition using Loess”; Loess refers to a non-linear regression technique
- Very versatile and robust.
- STL will handle any type of seasonality.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Robust to outliers, so that occasional unusual observations will not affect the estimates of the trend-cycle and seasonal components.
- Will not go through technical derivations of it, but will learn by looking at examples and experiment with settings

# STL decomposition

```
fit <- stl(elecequip, s.window=5, robust=TRUE)
autoplot(fit) +
  ggtitle("STL decomposition of electrical equipment index")
```

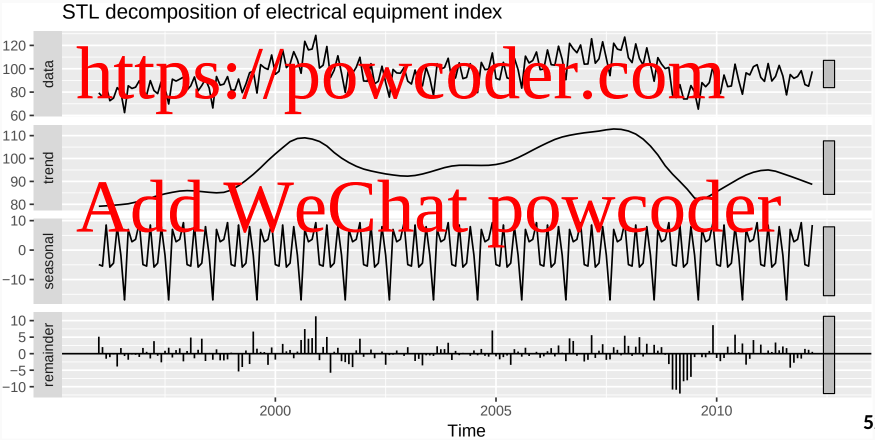




# STL decomposition

```
fit <- stl(elecequip, s.window="periodic", robust=TRUE)
autoplot(fit) +
  ggtitle("STL decomposition of electrical equipment index")
```





## STL decomposition

- The two main parameters to be chosen when using STL are the trend-cycle window (`t.window`) and the seasonal window (`s.window`).
- These control how rapidly the trend-cycle and seasonal components can change.
- Smaller values allow for more rapid changes.
- Both `t.window` and `s.window` should be odd numbers;
  - `t.window` is the number of consecutive observations to be used when estimating the trend-cycle; controls wiggliness of trend component.
  - `s.window` is the number of consecutive years to be used in estimating each value in the seasonal component; controls variation on seasonal component

# STL decomposition



- `s.window` must be specified; there is no default value for it in the `stl()` function.

- If `s.window = "periodic"`, then it is equivalent to forcing the seasonal component to be identical across years.

# Outline

# Forecasting and decomposition

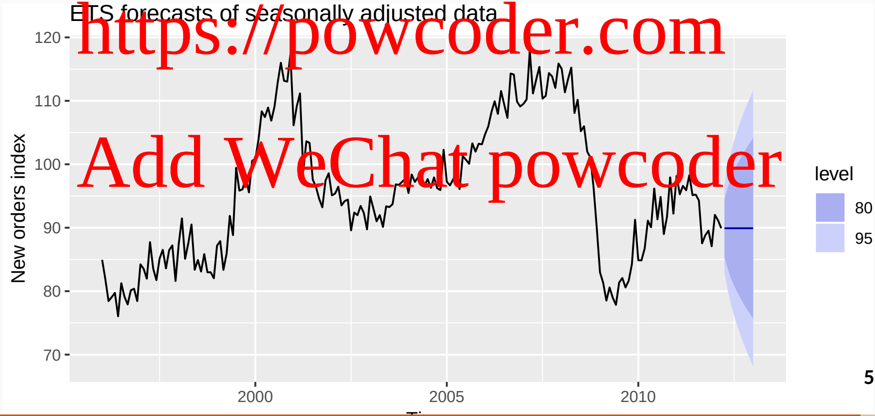


- Decompose time series into seasonally adjusted and seasonal component
- Forecast seasonal component by repeating the last year (i.e. seasonal naive forecast)
- Forecast seasonally adjusted data using non-seasonal time series method.
- Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.
- Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.

# Electrical equipment

```
fit <- stl(elecequip, t.window=13, s.window="periodic")
fit %>% seasadj() %>% naive() %>%
  autoplot() + ylab("New orders index") +
  ggtitle("ETS forecasts of seasonally adjusted data")
```

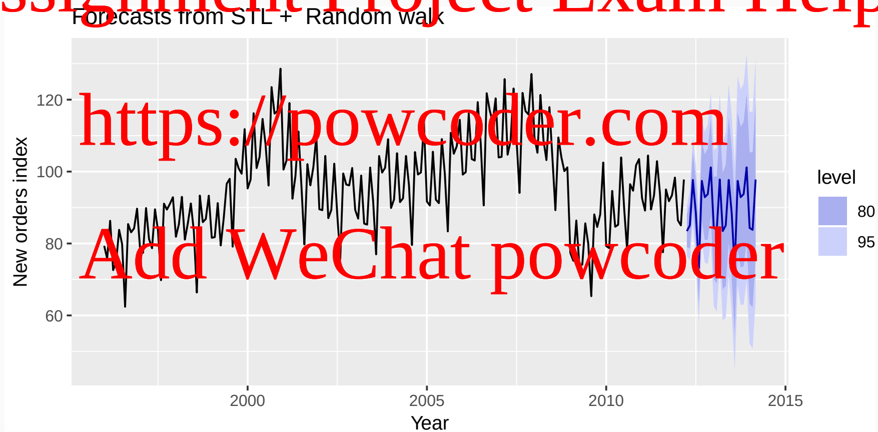




# Electrical equipment

```
fit %>% forecast(method='naive') %>%
  autoplot() + ylab("New orders index") + xlab("Year")
```

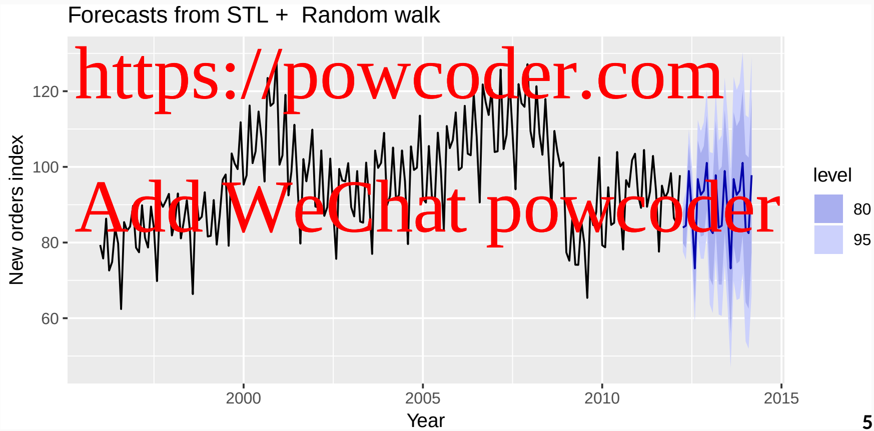




A short cut approach: `stlf()`

```
elecequip %>% stlf(method='naive') %>%
  autoplot() + ylab("New orders index") + xlab("Year")
```





## Decomposition and prediction intervals

- It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component. That is, the upper and lower limits of the prediction intervals on the seasonally adjusted data are reseasonalised by adding in the forecasts of the seasonal component.
- This ignores the uncertainty in the seasonal component estimate.
- It also ignores the uncertainty in the future seasonal pattern.