Lesson 2.3

STL Decomposition Method

STL Decomposition

STL ("Seasonal and Trend decomposition using Loess") is a versatile and robust method for decomposing time series. Loess is a method for estimating nonlinear relationships.

STL has several advantages over the classical, SEATS and X11 decomposition methods.

1. Unlike SEATS and X11, STL will handle any type of seasonality, not only monthly and quarterly data.
2. The seasonal component is allowed to change over time, and the rate of change can be controlled by the user.
3. The smoothness of the trend-cycle can also be controlled by the user.
4. It can be robust to outliers (occasional unusual observations will not affect the estimates of the trend-cycle and seasonal components but can affect the remainder component).
5. It can not handle trading day or calendar variation automatically, and it only provides facilities for additive decomposition.
6. It is possible to obtain a multiplicative decomposition by first taking logarithms of the data, then back-transforming the components

Decompositions between additive and multiplicative can be obtained using a Box-Cox transformation of the data with $0 < \lambda < 1$

- A value of $\lambda = 0$ corresponds to the multiplicative decomposition
- A value of $\lambda = 1$ corresponds to the additive decomposition

The two main parameters to be chosen when using STL are the trend-cycle window (*t.window*) and the seasonal window (*s.window*). The *t.window* is the number of consecutive observations to be used when estimating the trend-cycle and *s.window* is the number of consecutive years to be used in estimating each value in the seasonal component.

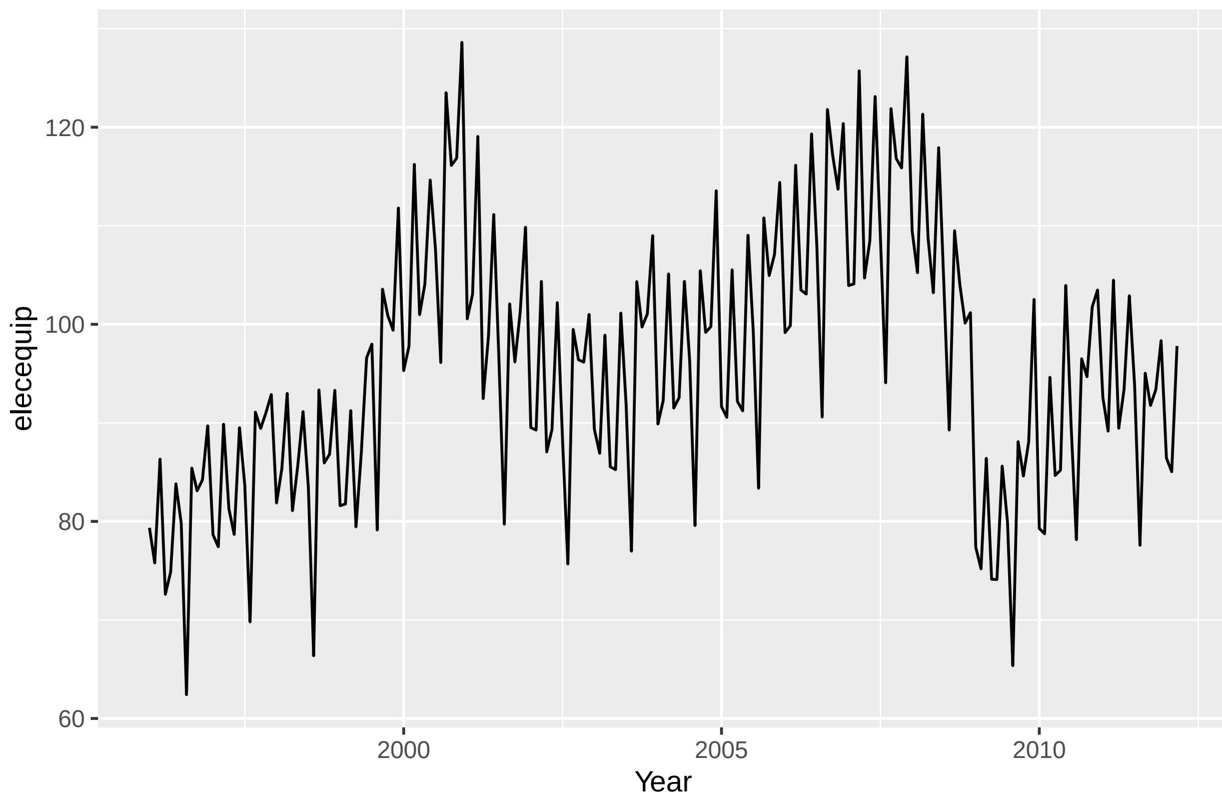
These parameters control how rapidly the trend-cycle and seasonal components can change. Smaller values allow for more rapid changes. Both parameters must be odd numbers. The user must specify *s.window* as there is no default but specifying *t.window* is optional, and a default value will be used if it is omitted.

Below is the time series plot of the electrical equipment orders data.

```r
library(fpp2)
```

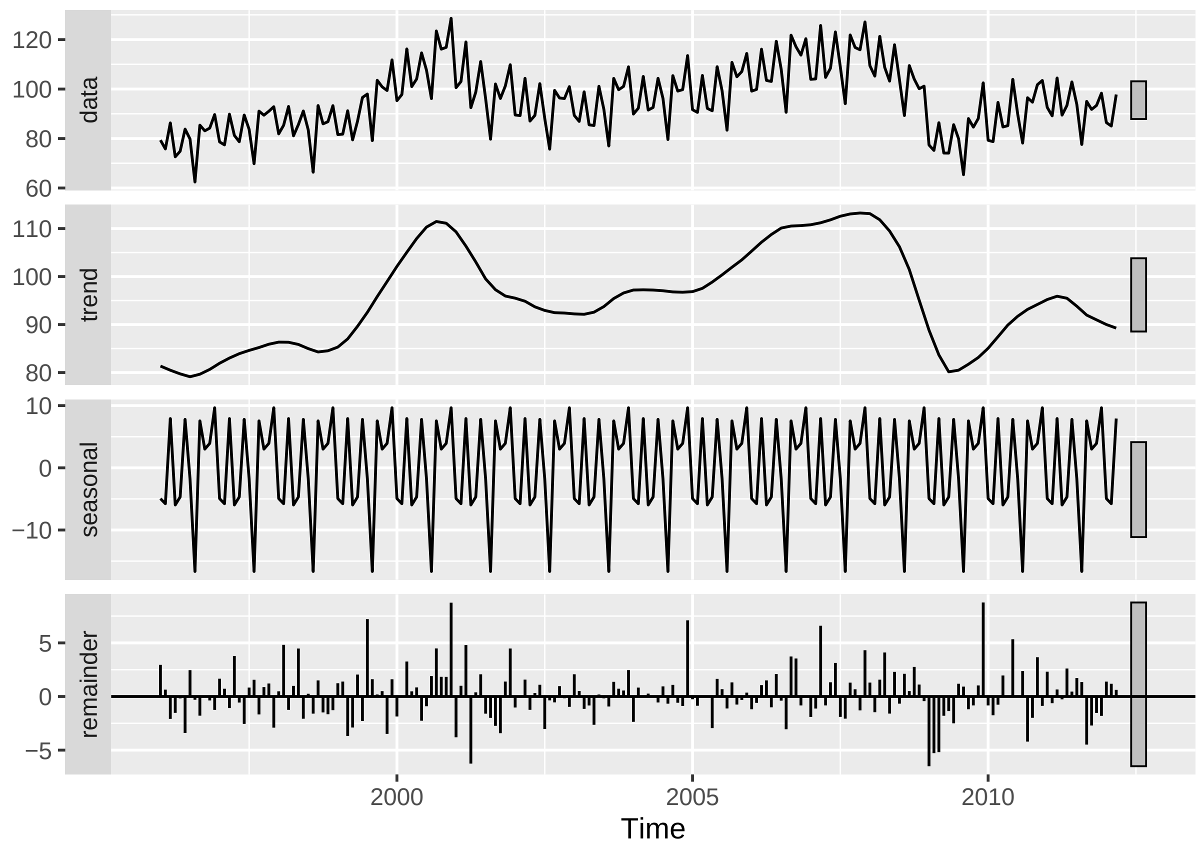
```
## Warning: package 'fpp2' was built under R version 4.5.2

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## -- Attaching packages -------------------------------------------- fpp2 2.5.1 --

## v ggplot2   4.0.0      v fma       2.5
## v forecast  8.24.0     v expsmooth 2.3

## Warning: package 'forecast' was built under R version 4.5.2

## Warning: package 'fma' was built under R version 4.5.2

## Warning: package 'expsmooth' was built under R version 4.5.2

##
```

```r
autoplot(elecequip) +
  xlab("Year")
```



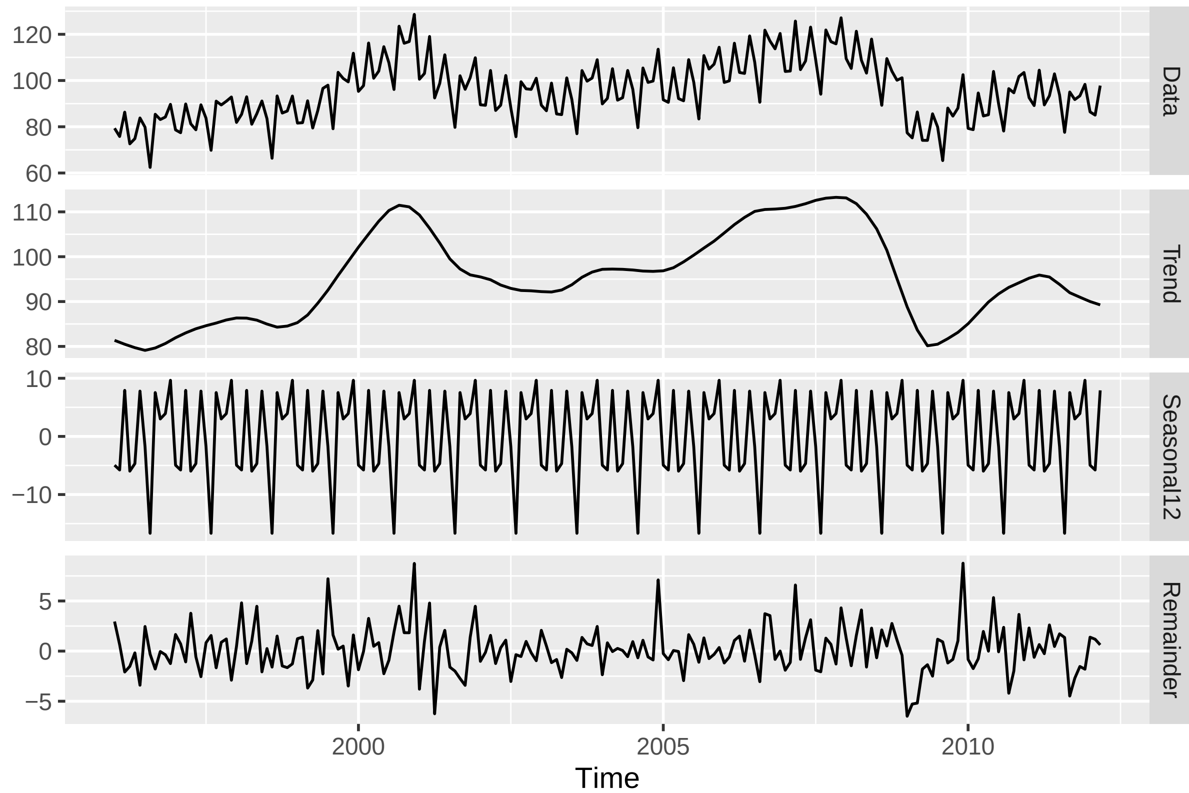

```r
elecequip %>%
  stl(t.window=13, s.window="periodic", robust=TRUE) %>%
  autoplot()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## i The deprecated feature was likely used in the forecast package.
##   Please report the issue at <https://github.com/robjhyndman/forecast/issues>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



The *mstl()* function provides a convenient automated STL decomposition using s.window=13, and t.window also chosen automatically. This usually gives a good balance between overfitting the seasonality and allowing it to slowly change over time. But, as with any automated procedure, the default settings will need adjusting for some time series.

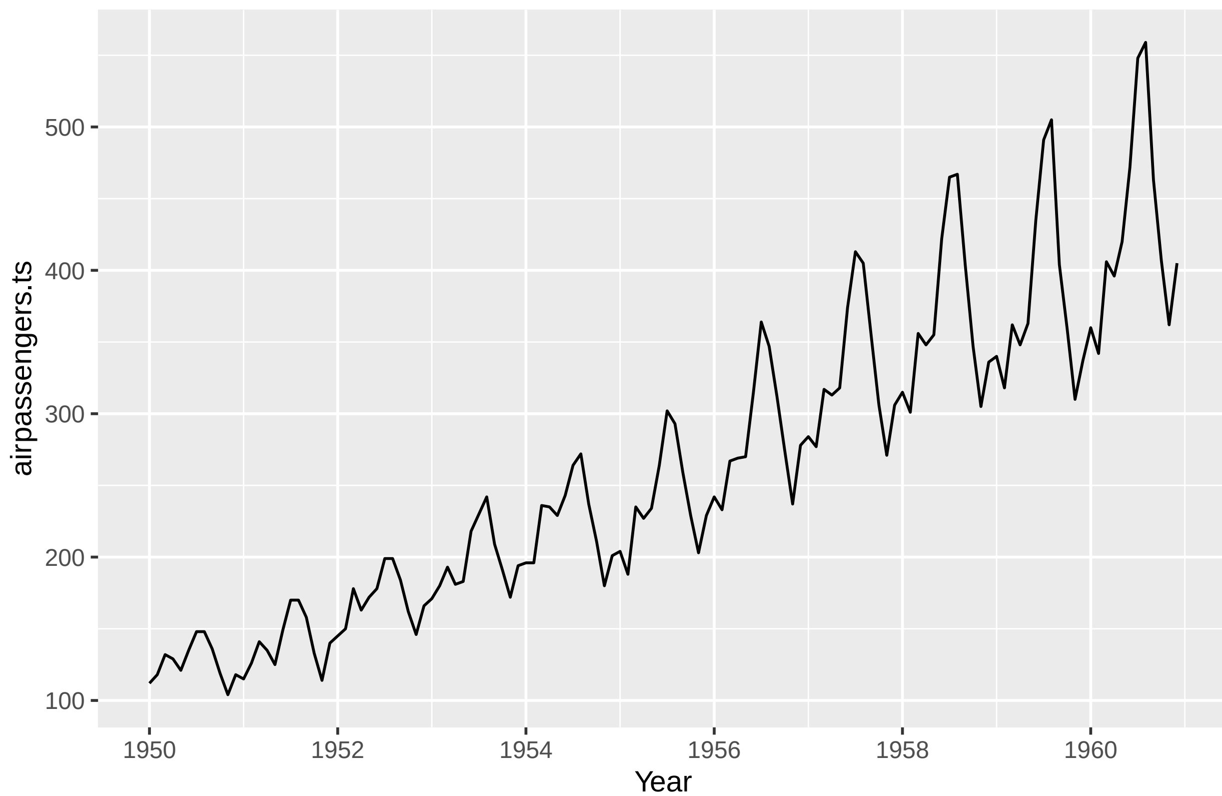
```
elecequip %>%
  mstl(t.window=13, s.window="periodic", robust=TRUE) %>%
  autoplot()
```



As with the other decomposition methods, to obtain the separate components, use the *seasonal()* function for the seasonal component, the *trendcycle()* function for trend-cycle component, and the *remainder()* function for the remainder component. The *seasadj()* function can be used to compute the seasonally adjusted series.

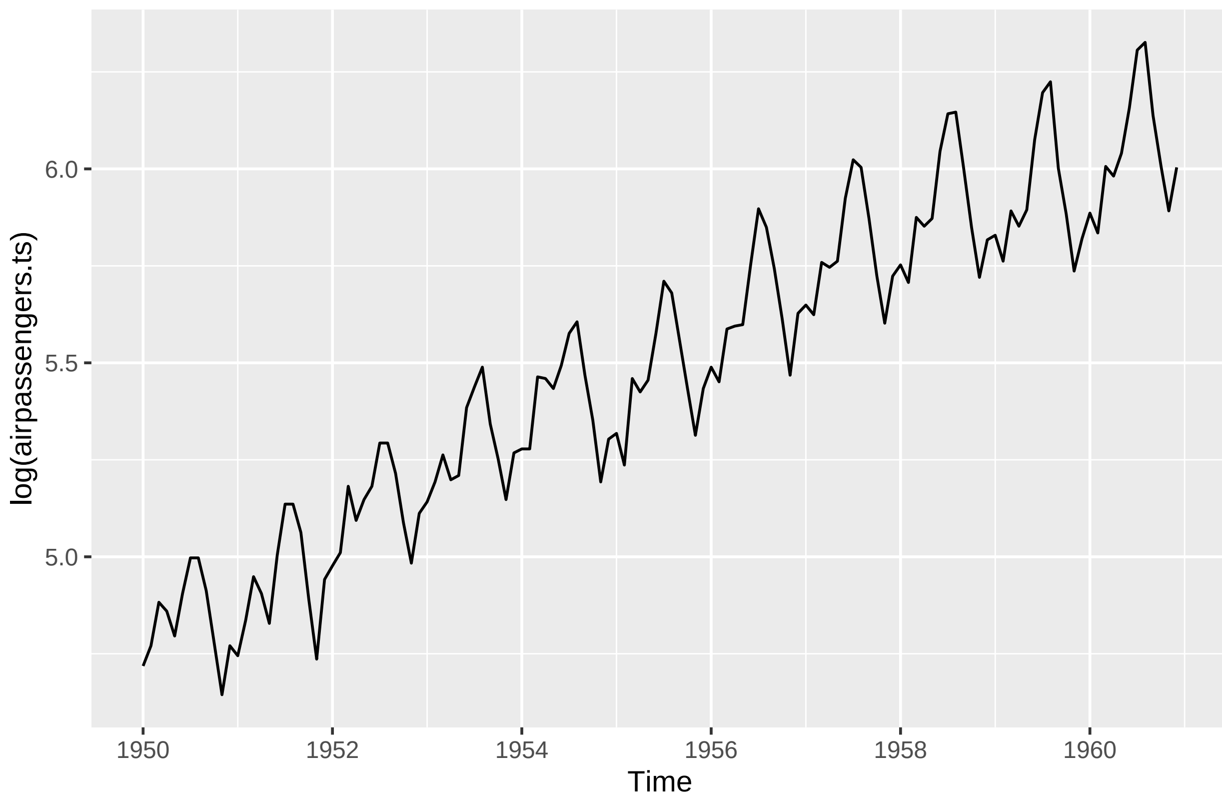
The air passengers data exhibits multiplicative seasonality.

```
library(seasonal)
airpassengers <- read.csv("AirPassengers.csv")
airpassengers.ts <- ts(airpassengers$Passengers,
                frequency = 12,
                start = c(1950, 1),
                end = c(1960, 12))
autoplot(airpassengers.ts) + xlab("Year")
```

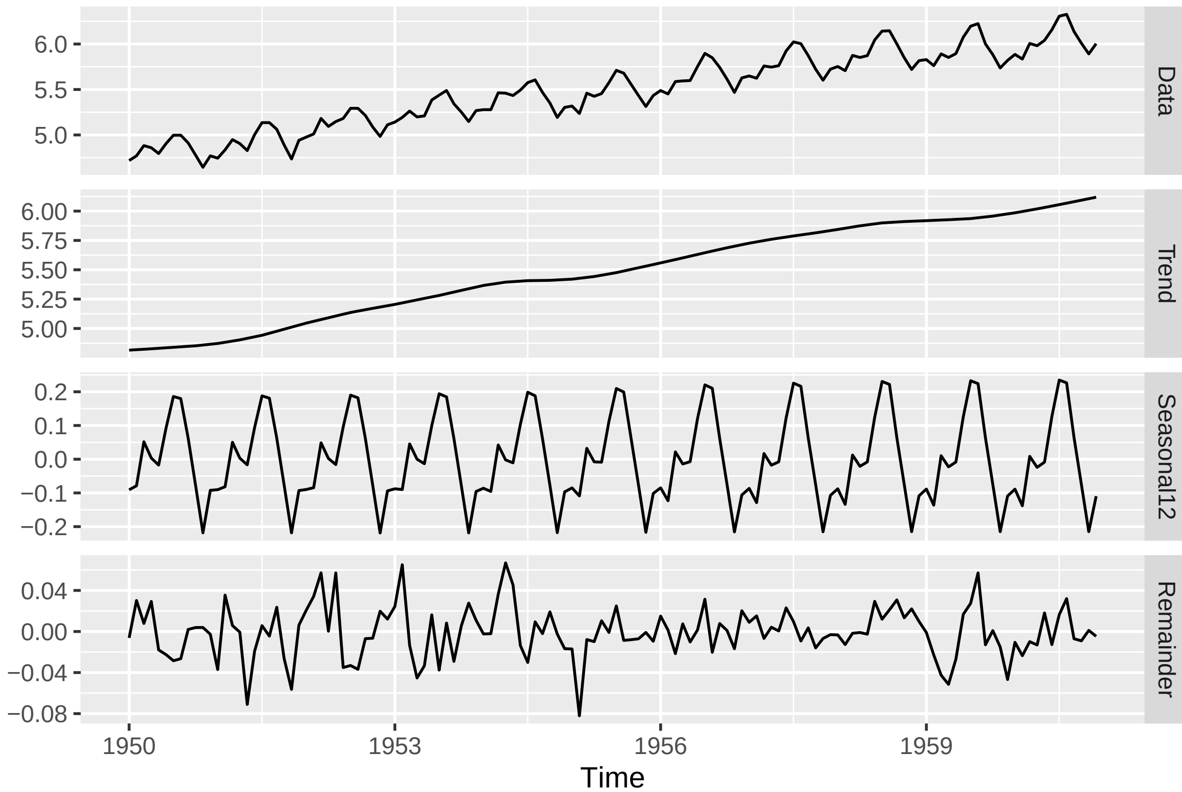


To apply STL we need to use the log of the time series. Below is the time series plot of the log-transformed data. An additive seasonality is now apparent.

```
autoplot(log(airpassengers.ts))
```




```
mstl(log(airpassengers.ts)) %>%
  autoplot()
```



6.0
5.5
5.0
Data
6.00
5.75
5.50
5.25
5.00
Trend
0.2
0.1
0.0
−0.1
−0.2
Seasonal12
0.04
0.00
−0.04
−0.08
Remainder
1950
1953
1956
1959
Time