Statistics for Decision Making: Broad Introduction

Classical Decomposition

Prof. Le Wang

Classical Decomposition Method (Chapter 6.3 in FPP)

The **classical decomposition method** originated in the 1920s.

It is a relatively simple procedure and forms the basis for most other methods of time series decomposition.

There are two forms of classical decomposition:

1. an additive decomposition (which we will focus on) and
2. a multiplicative decomposition.

These are described for a time series with seasonal period m. (e.g., $m = 4$ for quarterly data, $m = 12$ for monthly data, $m = 7$ for daily data with a weekly pattern).

The command that we will use is `decompose()`

The results can be immediately plotted with one line of code.
(think of a composite function in mathematics)

```r
library(quantmod)
getSymbols('UNRATE',src='FRED')
```

```
## [1] "UNRATE"
```

```r
head(UNRATE)
```

```
##            UNRATE
## 1948-01-01    3.4
## 1948-02-01    3.8
## 1948-03-01    4.0
## 1948-04-01    3.9
## 1948-05-01    3.5
## 1948-06-01    3.6
```

```r
plot(decompose(UNRATE))
```

```
## Error in decompose(UNRATE): time series has no or less than 2 periods
```

R does not recognize it. This is due to a limitation of the `xts` data structure.

While it is in the time series format, it does not provide **frequency information**. In other words, R recognizes its time series structure (i.e., the chronical order of the data), but does not recognize whether it is quarterly or monthly data.

```
attr(UNRATE, 'frequency') <- 12
```

Data	Frequency
Annual	1
Quarterly	4
Monthly	12
Weekly	52

Once we have assigned the frequency information to our data, we can formally declare its time series feature by typing

```
time <- as.ts(UNRATE)
```

We can now apply the classical decomposition methods to this time series and save the results into an object called `results`

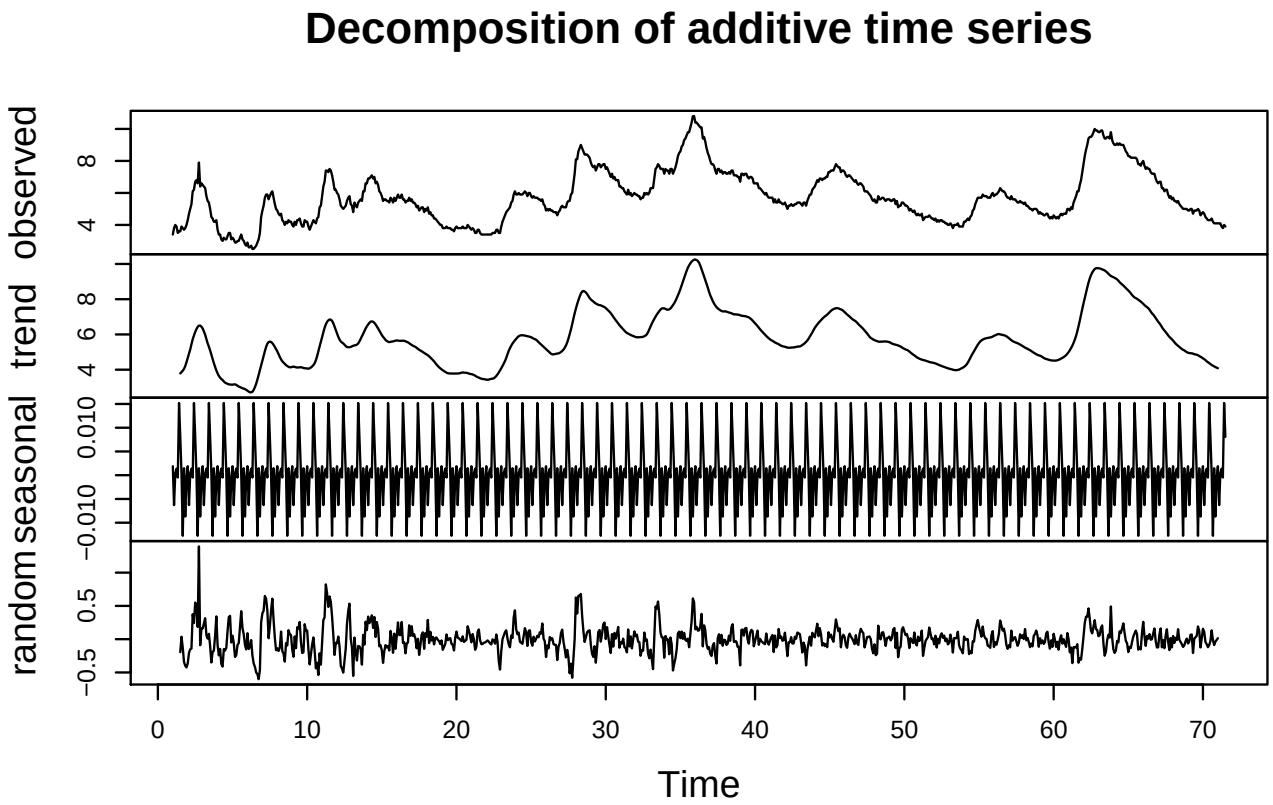
```
results <- decompose(time)
```

This object in which you store your results is a `list` object, one of those data structures that can store objects of different types and sizes. What does it contain?

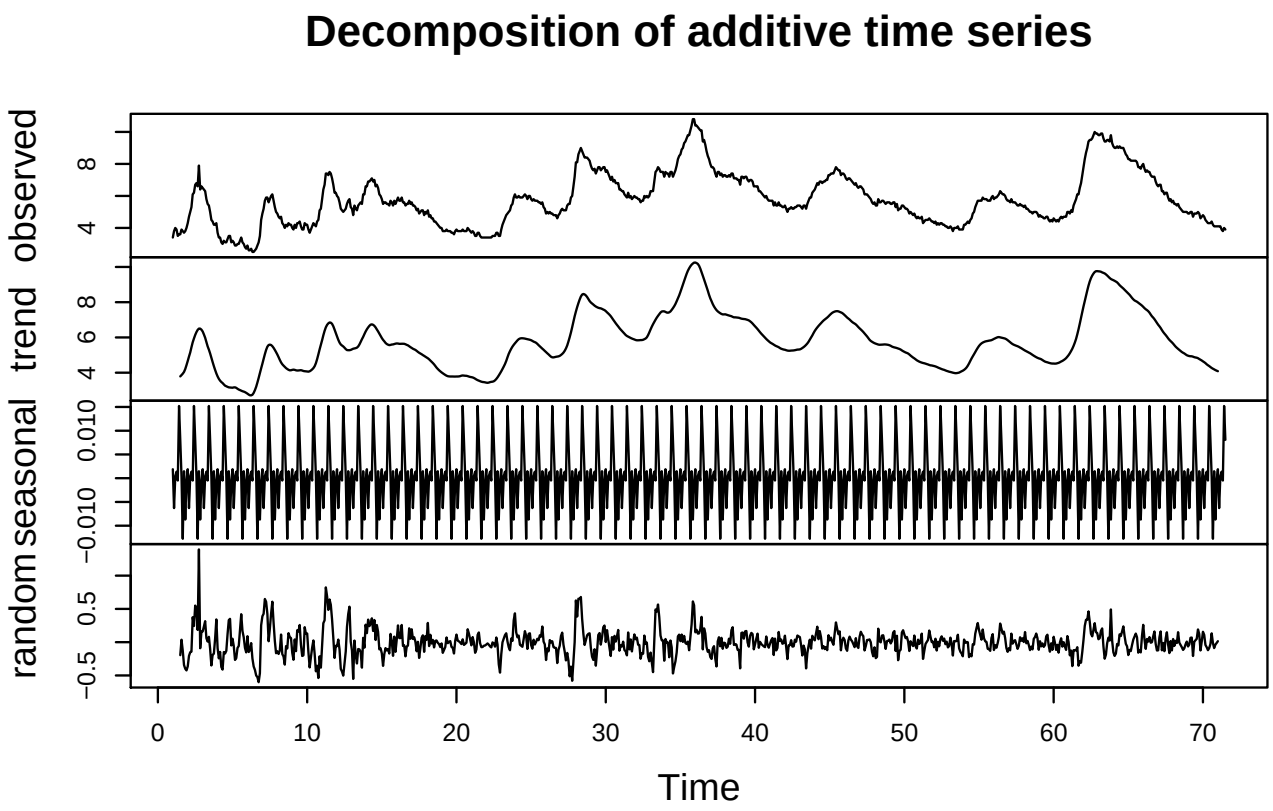
```
ls(results)
```

```
## [1] "figure"   "random"   "seasonal" "trend"    "type"
```

```
plot(results)
```

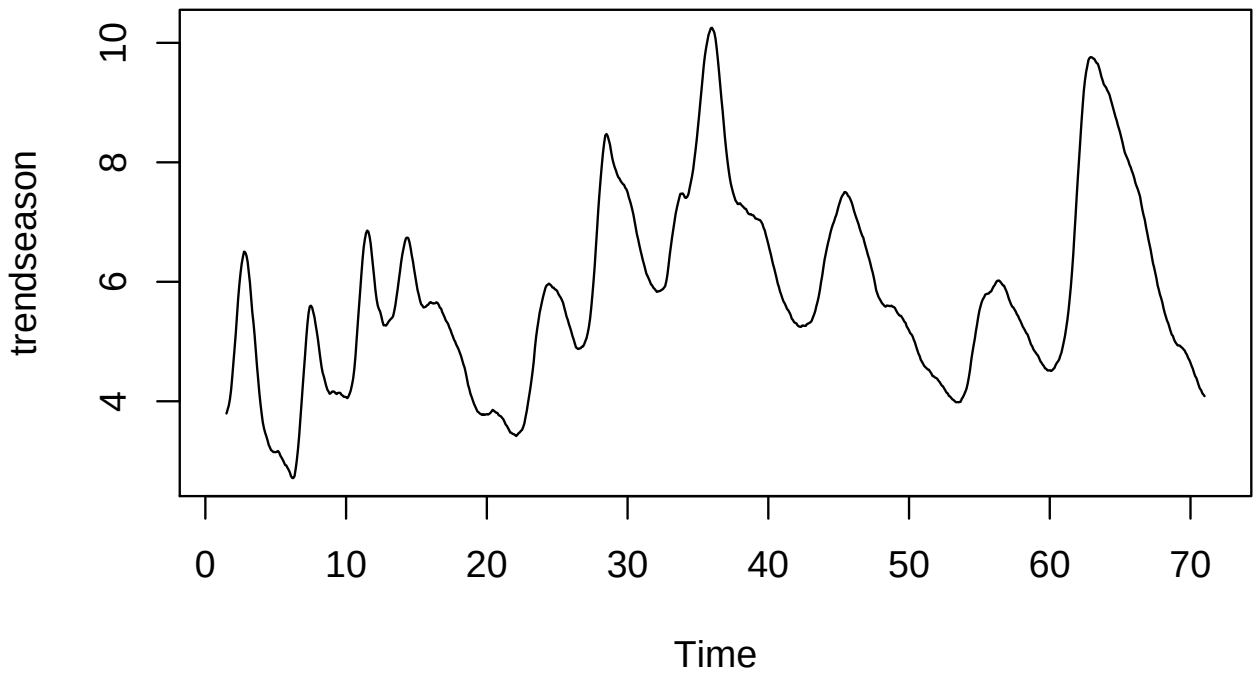


```
plot(decompose(as.ts(UNRATE)))
```



We can focus on the trend and seasonal components

```
trendseason <- results$trend +  results$seasonal
plot(trendseason)
```



While classical decomposition is still widely used, it is not recommended. There are now several much better methods. The issues with this method is listed in the book.

We now turn to one of them: STL