Lesson 2.1

Classical Decomposition of Time Series

Introduction

In Lesson 1.2 we discussed time series patterns components: trend, seasonality, cycles, and noise. For the purpose of choosing adequate forecasting methods, it is useful to dissect a time series into its constituent components. This process is called **decomposition** of a time series.

When we decompose a time series into components, we usually combine the trend and cycle into a single trend-cycle component (often just called the trend for simplicity). Thus, we can think of a time series as comprising three components: a trend-cycle component, a seasonal component, and a remainder component (containing anything else in the time series). For some time series (e.g., those that are observed at least daily), there can be more than one seasonal component, corresponding to the different seasonal periods.

Classical decomposition

The classical decomposition method originated in the 1920s. It is a relatively simple procedure, and forms the starting point for most other methods of time series decomposition. There are two forms of classical decomposition: an **additive decomposition** and a **multiplicative decomposition**. These are described below for a time series with seasonal period m (e.g., $m = 4$ for quarterly data, $m = 12$ for monthly data, and $m = 7$ for daily data with a weekly pattern).

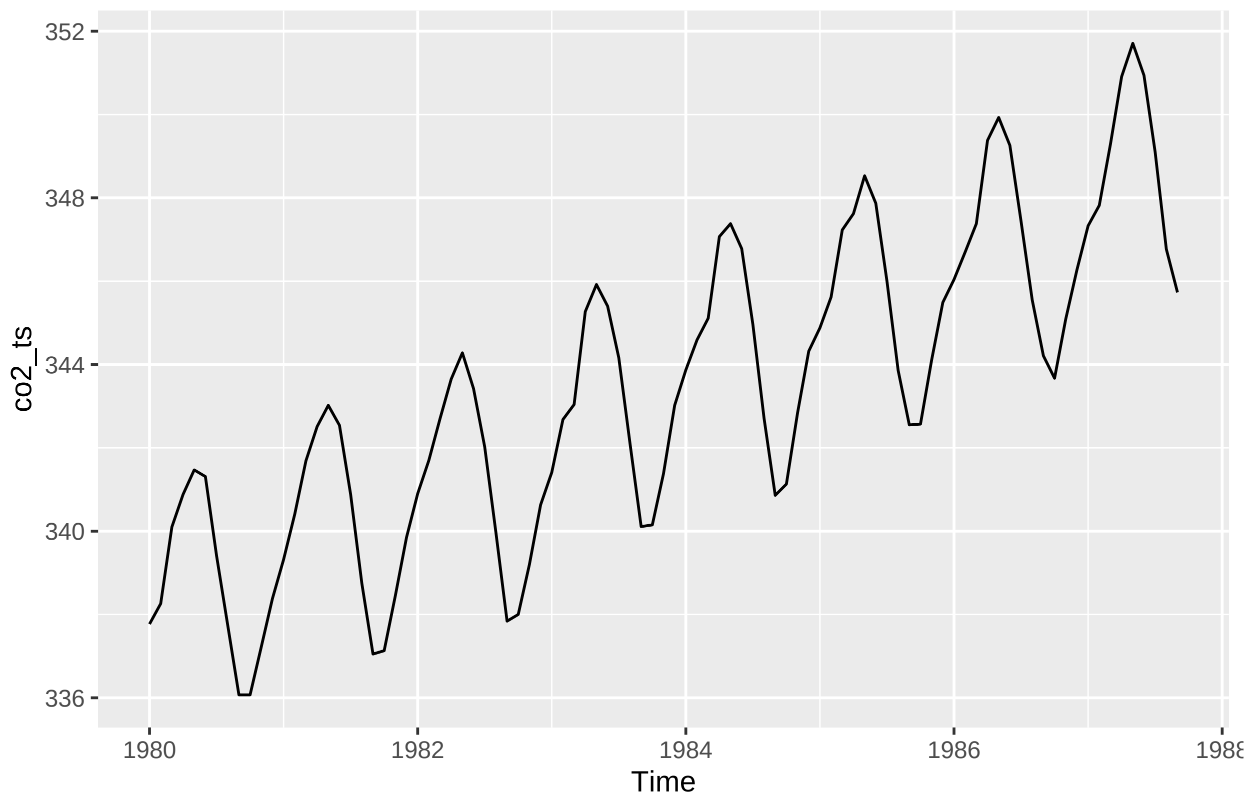
In classical decomposition, we assume that the seasonal component is constant from year to year. For multiplicative seasonality, the m values that form the seasonal component are sometimes called the "seasonal indices".

Additive decomposition

Additive decomposition is used when the seasonal variations are constant over time. Even if the trend is going up or down, the "width" of the seasonal peaks and valleys stays roughly the same. Note also that the sum of the seasonal indices in an additive model is zero.

As an illustration, consider the carbon dioxide dataset we used in Lesson 1.1.

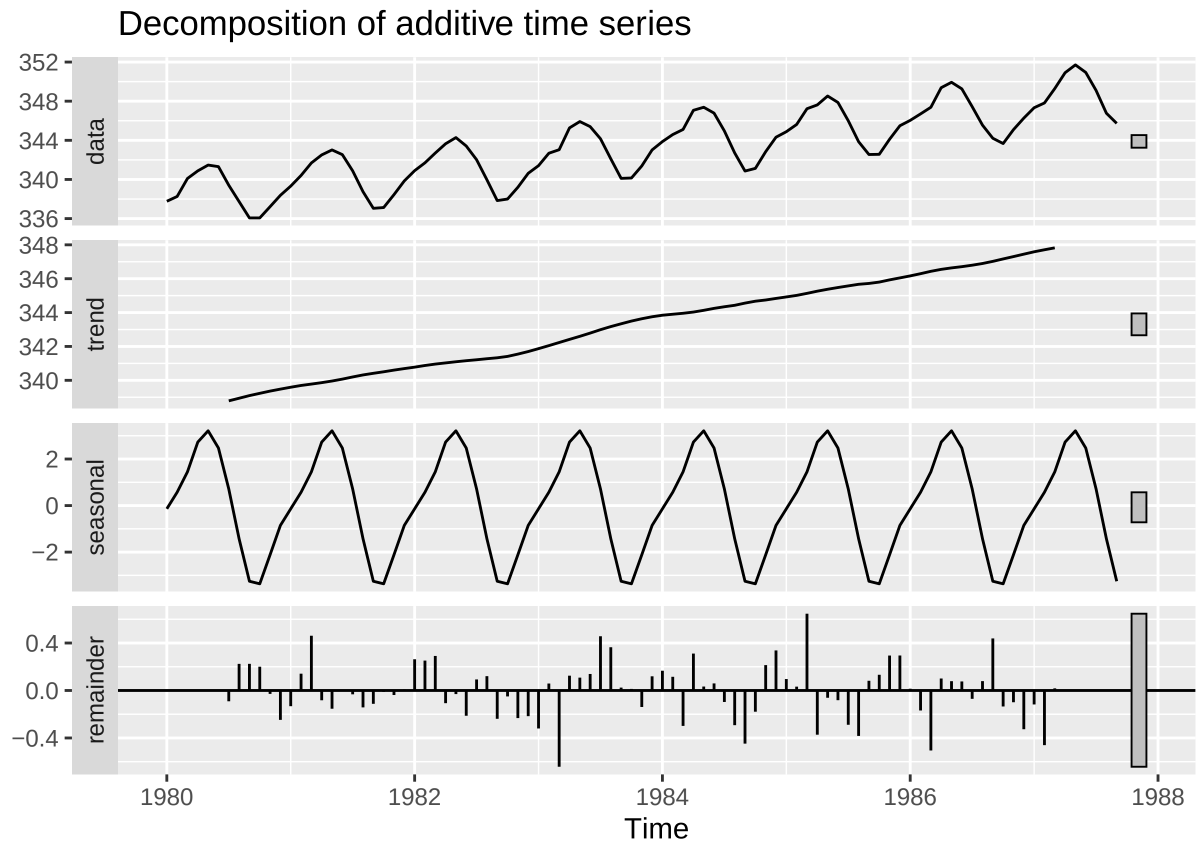
```r
library(fpp2)
co2<-read.csv("CO2 1980-1987.csv")
co2_ts<-ts(co2$CO2,
           start = c(1980,1),
           end = c(1987,9),
           frequency=12)
autoplot(co2_ts)
```



Seasonal fluctuations are roughly constant in size over time and do not seem to depend on the level of the time series. Hence, an additive model can be applied. That is, the seasonal fluctuations do not depend on the level of the series.

The basic command in R for (classical) decomposition is *decompose()*. For an additive model, the code is: **decompose(name of series, type = "additive")**.

```r
co2.comp <- decompose(co2_ts,type="additive")
autoplot(co2.comp)
```



Then we can extract the component values from the output of the *decompose()* function.

```
trend <- co2.comp$trend
seasonality <- co2.comp$seasonal
random <- co2.comp$random
seasonal.indices <- co2.comp$figure
print(trend)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1980       NA       NA       NA       NA       NA       NA 338.7879 338.9421
## 1981 339.5929 339.6962 339.7792 339.8642 339.9596 340.0721 340.1992 340.3187
## 1982 340.7775 340.8758 340.9596 341.0287 341.0962 341.1600 341.2138 341.2758
## 1983 341.8708 342.0492 342.2329 342.4171 342.5975 342.7879 342.9900 343.1721
## 1984 343.8442 343.9025 343.9587 344.0308 344.1325 344.2475 344.3437 344.4287
## 1985 344.9242 345.0154 345.1338 345.2642 345.3771 345.4788 345.5758 345.6692
## 1986 346.1658 346.2962 346.4358 346.5508 346.6375 346.7108 346.7971 346.8975
## 1987 347.5883 347.7083 347.8225       NA       NA       NA       NA       NA
##           Sep      Oct      Nov      Dec
## 1980 339.0979 339.2321 339.3646 339.4804
## 1981 340.4146 340.5042 340.6042 340.6933
## 1982 341.3308 341.4125 341.5483 341.6992
```

```
## 1983 343.3379 343.4992 343.6350 343.7533
## 1984 344.5600 344.6712 344.7421 344.8354
## 1985 345.7204 345.8000 345.9317 346.0479
## 1986 347.0238 347.1671 347.3050 347.4492
## 1987       NA
```

```r
print(seasonality)
```

```
##             Jan        Feb        Mar        Apr        May        Jun
## 1980 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
## 1981 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
## 1982 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
## 1983 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
## 1984 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
## 1985 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
## 1986 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
## 1987 -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
##             Jul        Aug        Sep        Oct        Nov        Dec
## 1980  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
## 1981  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
## 1982  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
## 1983  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
## 1984  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
## 1985  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
## 1986  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
## 1987  0.7131366 -1.4263872 -3.2522801
```

```r
print(random)
```

```
##              Jan         Feb         Mar         Apr         May         Jun
## 1980          NA          NA          NA          NA          NA          NA
## 1981 -0.13230324  0.14150628  0.46126819 -0.08240245 -0.15372189 -0.00511078
## 1982  0.26311343  0.25192295  0.29085152 -0.10698578 -0.03038856 -0.21302745
## 1983 -0.32021991  0.05858962 -0.64248181  0.12468089  0.10836144  0.13905589
## 1984  0.16644676  0.11525628 -0.29831515  0.31093089  0.03336144  0.05947255
## 1985  0.09644676  0.03233962  0.64668485 -0.37240245 -0.06122189 -0.08177745
## 1986  0.01478009 -0.16849372 -0.50539848  0.10093089  0.07836144  0.07613922
## 1987 -0.11771991 -0.46057705  0.01793485          NA          NA          NA
##              Jul         Aug         Sep         Oct         Nov         Dec
## 1980 -0.09105324  0.22430390  0.22436343  0.20031581 -0.02867229 -0.24766038
## 1981 -0.03230324 -0.14236276 -0.11230324 -0.01176753 -0.03825562  0.00942295
## 1982  0.09311343  0.12055390 -0.23855324 -0.05010086 -0.23242229 -0.21641038
## 1983  0.45686343  0.36430390  0.02436343  0.01323247 -0.13908896  0.11942295
```

```
## 1984 -0.09688657 -0.29236276 -0.44771991 -0.17885086  0.21382771  0.33733962
## 1985 -0.28896991 -0.38277943  0.08186343  0.13239914  0.29424438  0.29483962
## 1986 -0.07021991  0.07888724  0.43853009 -0.13468419 -0.09908896 -0.32641038
## 1987          NA          NA          NA
```

```
print(seasonal.indices)
```

```
##  [1] -0.1406134  0.5722437  1.4495651  2.7282358  3.2141386  2.4730274
##  [7]  0.7131366 -1.4263872 -3.2522801 -3.3623991 -2.1159110 -0.8527563
```

```
sum(seasonal.indices)
```

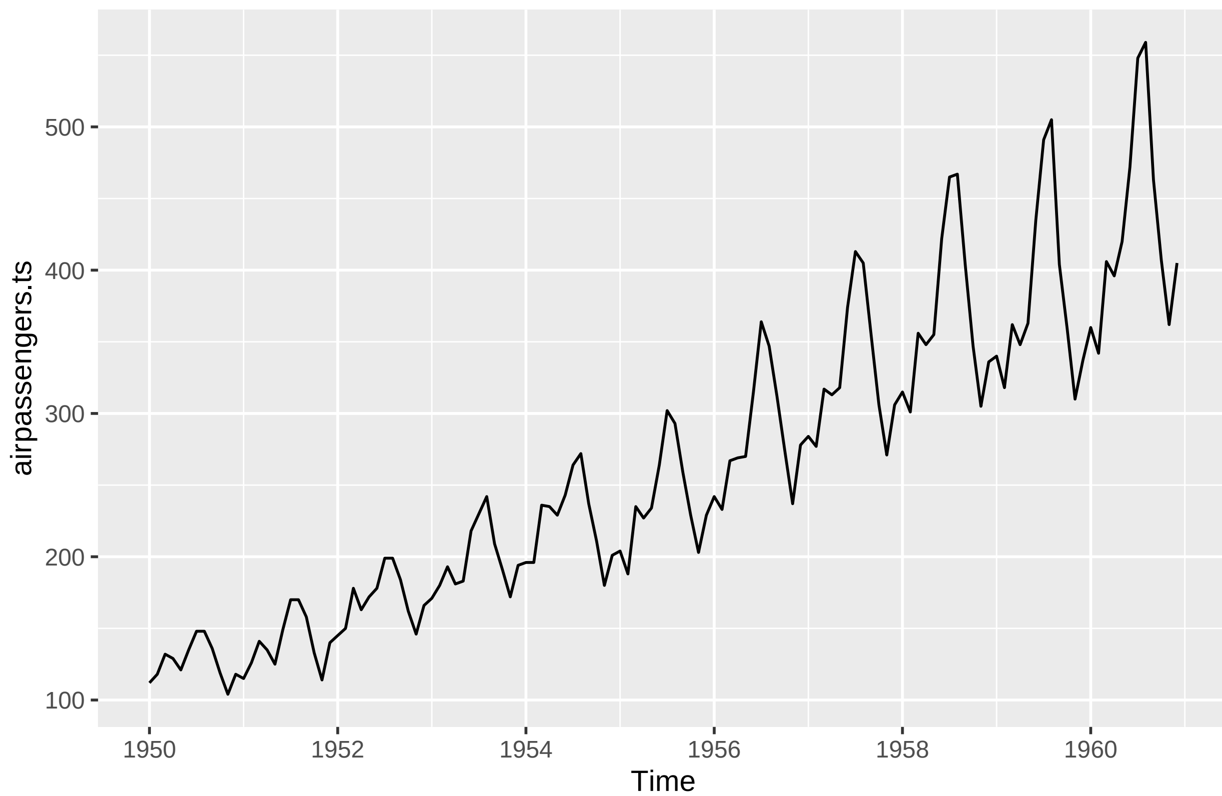
```
## [1] -9.992007e-16
```

Multiplicative decomposition

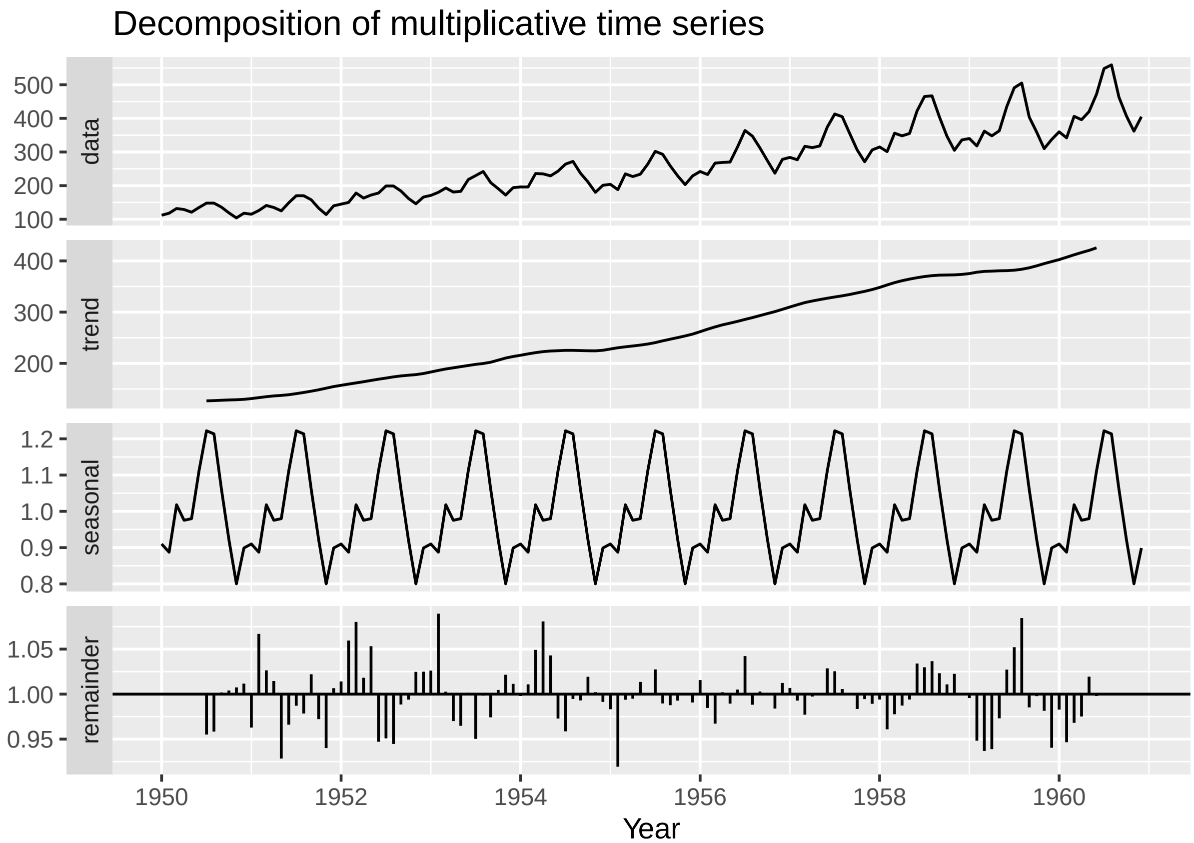
Multiplicative decomposition is used when the seasonal variations change proportionally with the trend. If the trend increases, the seasonal swings get wider (larger); if the trend decreases, the swings get smaller. Thus, multiplicative decomposition is appropriate when the seasonal pattern is a percentage of the trend. The sum of the seasonal indices in multiplicative model is equal to the seasonal frequency of the series.

Consider the monthly number of airline passengers from 1950 to 1960.

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




```r
airpassengers.comp <- decompose(airpassengers.ts,type = "multiplicative")
autoplot(airpassengers.comp) + xlab("Year")
```




```r
print(airpassengers.comp$figure)
```

```
##  [1] 0.9100037 0.8873765 1.0182037 0.9754120 0.9798128 1.1115898 1.2221466
##  [8] 1.2135961 1.0609168 0.9217670 0.8002132 0.8989616
```

```r
sum(airpassengers.comp$figure)
```

```
## [1] 12
```

Remarks

1. In classical decomposition, the estimate of the trend-cycle is unavailable for the first few and last few observations. For example, if $m = 12$, there is no trend-cycle estimate for the first 6 or the last 6 observations. Consequently, there is also no estimate of the remainder component for the same time periods.
2. The trend-cycle estimate tends to over-smooth rapid rises and falls in the data.
3. Classical decomposition methods assume that the seasonal component repeats from year to year. Hence, they are unable to capture seasonal changes over time.
4. The classical method is not robust to small sudden shifts in a time series.