Introduction to Time Series Analyis

Dr Rafael de Andrade Moral
Associate Professor of Statistics, Maynooth University

rafael.deandrademoral@mu.ie
https://rafamoral.github.io

Outline

- White noise, autocorrelation, stationarity
- Time plots, seasonal plots, autocorrelation plots
- Time series decomposition
- Benchmark forecasting methods
- Exponential smoothing and ETS
- ARIMA modelling
- Dynamic linear regression

Recommended reading: Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3

Forecasting

- The predictability of an event or quantity depends on several factors, including
  1. how well we understand the factors that contribute to it;
  2. how much data is available;
  3. how similar the future is to the past;
  4. whether the forecasts can affect the thing we are trying to forecast.
- Examples
  - electricity demand
  - exchange rates
  - lotto numbers
  - predicting heads or tails in the next coin toss
- Is it reasonable to assume past patterns will continue into the future?

Why?

Why forecast the future? (a few examples)

- How much traffic will be on a particular city's roads tomorrow?
  - we can broadcast information about the city's traffic conditions on radio.
- What will the GDP be in a particular country next year?
  - the government can propose a budget.
- What will global temperatures be in 10 years?
  - we can prepare for the consequences of climate change.
- How many people will be infected with a disease next month?
  - healthcare systems can prepare for a potential increase in demand for hospital beds.

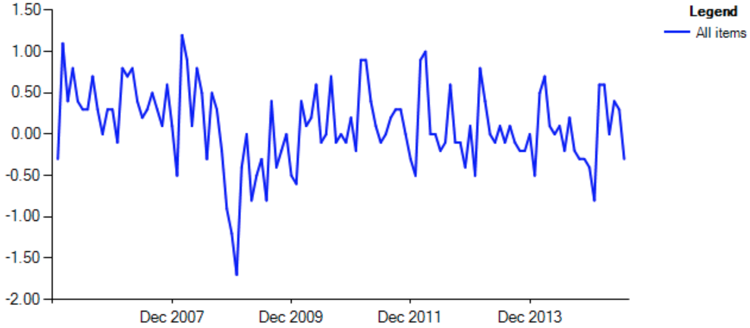
What is a Time Series?

- A time series is a sequence of data points, whose measurements are taken over time, i.e. we observe some quantity Y_t at time t.
- We can fix t in advance, but we don't know what Y_t will be; therefore Y_t is a random variable.
- There may be other variables included too (covariates).
- There may be missing values or outliers.
- Occasionally there may be more than one response variable (multivariate time series).

Goals of Time Series Analysis

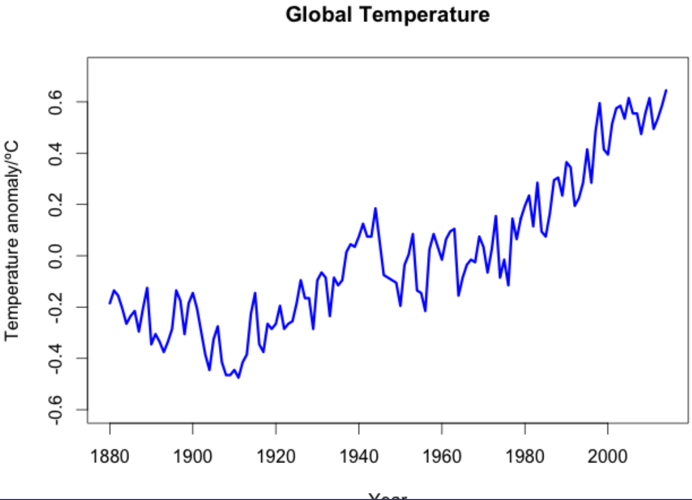
- Forecasting future values given the past information we observed.
- Interpolate or smooth the response variable for missing or non-measured times.
- Identify which factors are causing the time series to change.
- Understand the underlying behaviour of the time series.

Example 1: Consumer Price Index



Source: Irish CSO

Example 2: Global Temperature



Definition

Definition - Time series

A time series is a collection of random variables $Y_t, t \in \mathbb{N}$ or $Y_t, t \in \mathbb{Z}$. The X_t variables are $\mathbb{R}$-valued.

We can examine the multivariate joint distribution of any finite collection of random variables e.g.

$$f(Y_1, Y_2)$$
$$f(Y_2, Y_3, Y_4)$$

Considering all possible joint distibutions is not easy, therefore, it is sufficient to only examine the **moments** of these distributions.

In this course we will focus on the *mean*, *variance*, and *autocorrelation*

Autocorrelation

- A useful way to characterise the behaviour of a time series is the autocorrelation function
- (auto = self, i.e. correlation of the series with itself)
- We calculate the correlation of the series with itself shifted by 1 time point
- The shifted data set is known as the lagged time series
- We repeat the autocorrelation calculation for 1 lag, 2 lags, 3 lags, etc

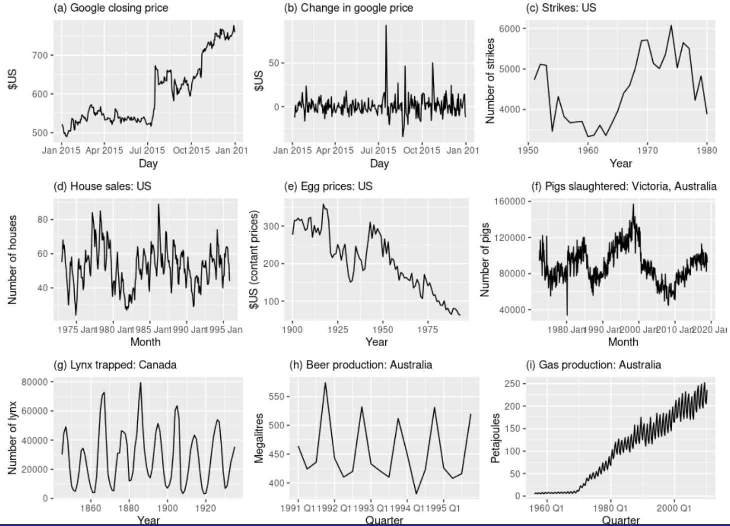
Stationarity

A time series Y_t is weakly stationary if the mean, variance and covariances do not evolve with time.

\begin{block}{Definition - Weak Stationarity} The time series Y_t is said to be (weakly) stationary if:

- the **mean** is constant
- the **variance** is constant
- the **autocorrelation** may depend only on the time lag

Which series are stationary?



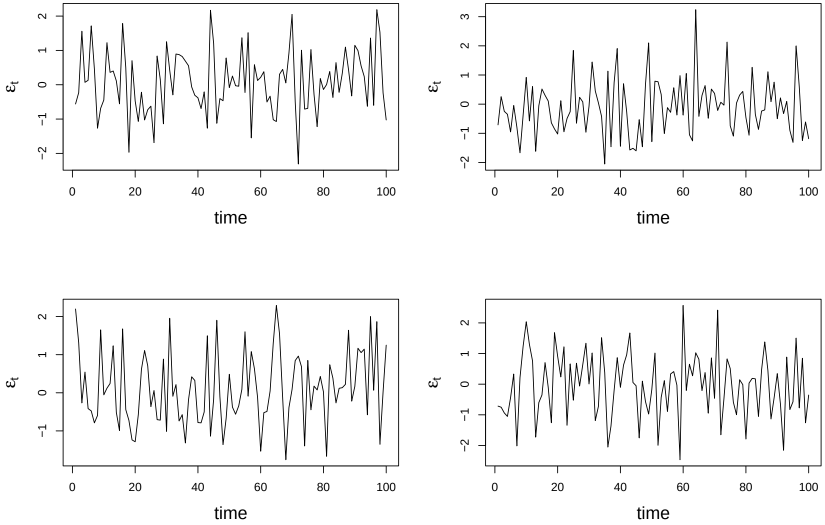
White noise process

- A process ε_t is called white noise if ε_t is a sequence of independent, identically distributed (IID) random variables.
- It is often assumed that the white noise variables ε_t are normally distributed with zero mean and variance σ^2.
- E.g.

$$\varepsilon_t \sim N(0, \sigma^2)$$

  is a white noise process.
- Based on these assumptions the time series ε_t is stationary.

White noise process



Examples

- A model with trend:

$$Y_t = \beta_0 + \beta_1 t + \varepsilon_t$$

- A moving average:

$$Y_t = \varepsilon_t - \frac{1}{2}\varepsilon_{t-1}$$

- A product of white noise variables:

$$Y_t = \prod_{i=1}^{t} \varepsilon_i$$

tsibbles

- `tsibble` objects extend tidy data frames (`tibble` objects) by introducing temporal structure.
- We may set up an `index` identifying our time variable t.
- Example: let's set the time series index to be the `Year` column, which associates the measurements (`Observation`) with the time of recording (`Year`).

```
library(fpp3)

y <- tsibble(Year = 2015:2019,
             Observation = c(123, 39, 78, 52, 110),
             index = Year)
y
```

```
## # A tsibble: 5 x 2 [1Y]
##    Year Observation
##   <int>       <dbl>
## 1  2015         123
## 2  2016          39
## 3  2017          78
```

tsibbles

- `tibbles` can be converted to `tsibbles` by adding the index and using `yearmonth()`

```
z <- tibble(Month = c("2019 Jan","2019 Feb","2019 Mar","2019 Apr","2019 May"),
            Observation = c(50,23,34,30,25))

z %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)
```

- First, the Month column is being converted from text to a monthly time object with `yearmonth()`.
- We then convert the data frame to a `tsibble` by identifying the index variable using `as_tsibble()`.
- Note the addition of `[1M]` on the first line indicating this is monthly data.

tsibbles

- Other useful functions:
  - quarterly: `yearquarter()`
  - monthly: `yearmonth()`
  - weekly: `yearweek()`
  - daily: `as_date()`, `ymd()`
  - subdaily: `as_datetime()`, `ymd_hms()`

Example

```r
prison <- readr::read_csv("https://OTexts.com/fpp3/extrafiles/prison_population

prison <- prison %>%
  mutate(Quarter = yearquarter(Date)) %>%
  select(-Date) %>%
  as_tsibble(key = c(State, Gender, Legal, Indigenous),
             index = Quarter)
```

- This `tsibble` contains 64 separate time series corresponding to the combinations of the 8 states, 2 genders, 2 legal statuses and 2 indigenous statuses.
- Each of these series is 48 observations in length, from 2005 Q1 to 2016 Q4.

Time plots

- For time series data, the obvious graph to start with is a time plot.
- The observations are plotted against the time of observation, with consecutive observations joined by straight lines.
- Example: weekly economy passenger load on Ansett airlines between Australia's two largest cities, Melbourne and Sydney:

```
## # A tsibble: 7,407 x 4 [1W]
## # Key:       Airports, Class [30]
##        Week Airports Class    Passengers
##      <week> <chr>    <chr>         <dbl>
##  1 1989 W28 ADL-PER  Business        193
##  2 1989 W29 ADL-PER  Business        254
##  3 1989 W30 ADL-PER  Business        185
##  4 1989 W31 ADL-PER  Business        254
##  5 1989 W32 ADL-PER  Business        191
##  6 1989 W33 ADL-PER  Business        136
##  7 1989 W34 ADL-PER  Business          0
##  8 1989 W35 ADL-PER  Business          0
##  9 1989 W36 ADL-PER  Business          0
## 10 1989 W37 ADL-PER  Business          0
## # i 7,397 more rows
```

Time plots

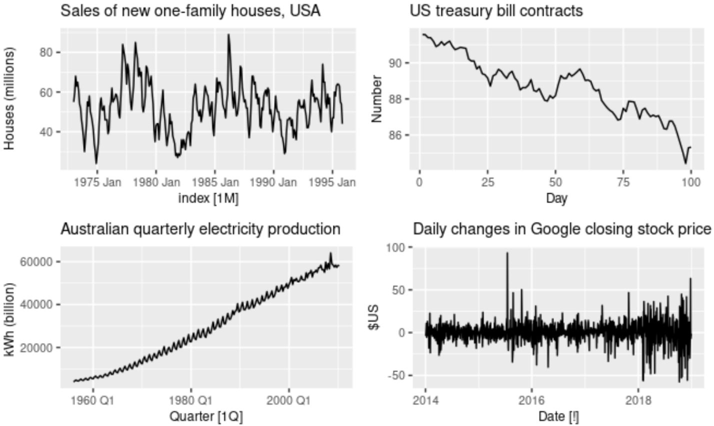
- The autoplot() command automatically produces an appropriate plot of whatever you pass to it in the first argument.
- In this case, it recognises melsyd_economy as a time series and produces a time plot.
- Interesting features:
- There was a period in 1989 when no passengers were carried – this was due to an industrial dispute.
- There was a period of reduced load in 1992. This was due to a trial in which some economy class seats were replaced by business class seats.
- A large increase in passenger load occurred in the second half of 1991.
- There are some large dips in load around the start of each year. These are due to holiday effects.
- There is a long-term fluctuation in the level of the series which increases during 1987, decreases in 1989, and increases again through 1990 and 1991.
- Any model will need to take all these features into account in order to effectively forecast the passenger load into the future.

Time Series Patterns: Trend

- A **trend** exists when there is a *long-term increase or decrease* in the data.
- It does not have to be linear.
- Sometimes we will refer to a trend as "changing direction" when it might go from an increasing trend to a decreasing trend.

Time Series Patterns: Trend

Which of the following time series show a trend?

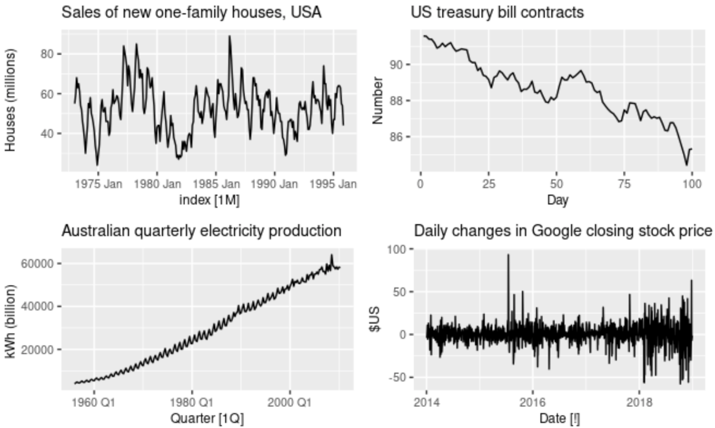


Time Series Patterns: Seasonal

- A **seasonal** pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week.
- Seasonality is always of a *fixed and known period*.

Time Series Patterns: Seasonal

Which of the following time series show seasonality?

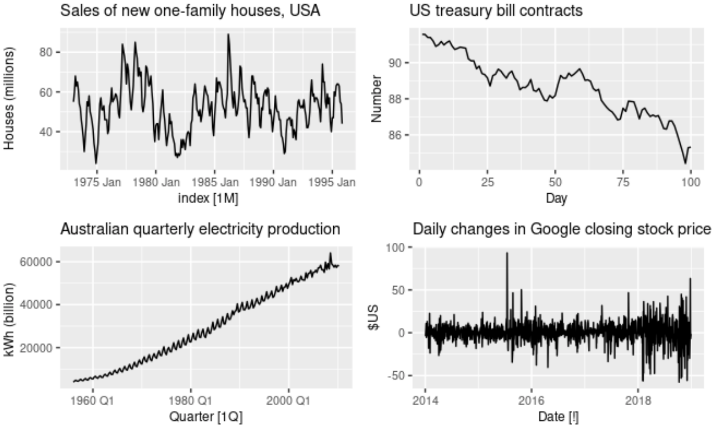


Time Series Patterns: Cyclic

- A **cycle** occurs when the data exhibit rises and falls that are *not of a fixed frequency*.
- These fluctuations are usually due to economic conditions, and are often related to the "business cycle".
- The duration of these fluctuations is usually at least 2 years.

Time Series Patterns: Cyclic

Which of the following time series show cyclic behaviour?



Time Series Patterns

- The monthly housing sales (top left) show strong seasonality within each year, as well as some strong cyclic behaviour with a period of about 6-10 years. There is no apparent trend in the data over this period.
- The US treasury bill contracts (top right) show results from the Chicago market for 100 consecutive trading days in 1981. Here there is no seasonality, but an obvious downward trend. Possibly, if we had a much longer series, we would see that this downward trend is actually part of a long cycle, but when viewed over only 100 days it appears to be a trend.
- The Australian quarterly electricity production (bottom left) shows a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behaviour here.
- The daily change in the Google closing stock price (bottom right) has no trend, seasonality or cyclic behaviour. There are random fluctuations which do not appear to be very predictable, and no strong patterns that would help with developing a forecasting model.

Seasonal plots

- A seasonal plot allows the underlying seasonal pattern to be seen more clearly, and is especially useful in identifying years in which the pattern changes.
- The code below creates a plot of energy demand in Ireland in 2017. The dataset includes the energy demand every 15 minutes for every day of the year of 2017.

```
library(fpp3)
library(aimsir17)

demand <- eirgrid17 %>%
  distinct(date, .keep_all = TRUE) %>%
  as_tsibble(index = date)

demand %>%
  autoplot(IEDemand) +
  ggtitle("Energy demand in Ireland")
```

Seasonal plots

- Where the data has more than one seasonal pattern, the `period` argument can be used to select which seasonal plot is required.
- The `demand` data contains quarter-hourly energy demand for Ireland.
- We can plot the daily pattern, weekly pattern or monthly pattern by specifying the `period` argument.

```
demand %>%
  fill_gaps() %>%
  gg_season(IEDemand, period = "day") +
  theme(legend.position = "none") +
  labs(y = "MWh", title = "Daily energy demand in 2017: Ireland")
```

Example: Mauna Loa Atmospheric CO2 Concentration

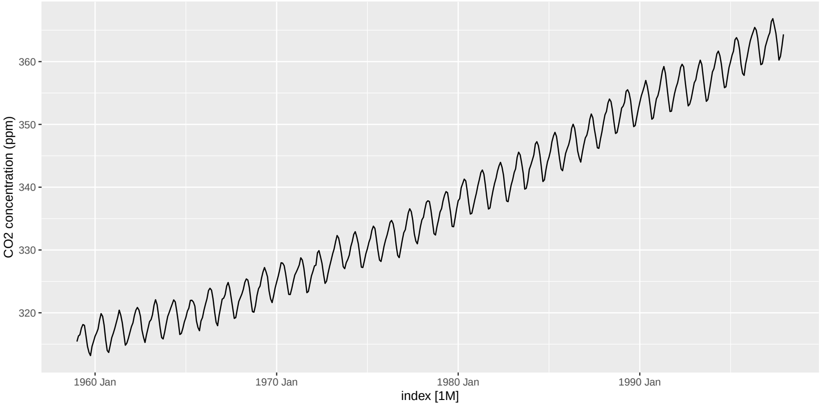
- The co2 data contains atmospheric CO2 concentrations in ppm measured at the North face of the Mauna Loa volcano.
- We can convert it to `tsibble` format easily, as shown below.

```
new_co2 <- as_tsibble(co2)
new_co2
```

```
## # A tsibble: 468 x 2 [1M]
##       index value
##       <mth> <dbl>
##  1 1959 Jan  315.
##  2 1959 Feb  316.
##  3 1959 Mar  316.
##  4 1959 Apr  318.
##  5 1959 May  318.
##  6 1959 Jun  318
##  7 1959 Jul  316.
##  8 1959 Aug  315.
##  9 1959 Sep  314.
## 10 1959 Oct  313.
## # i 458 more rows
```

Example: Mauna Loa Atmospheric CO_2 Concentration

- A time series plot shows an increasing trend and strong yearly seasonality.



Decomposing a Time Series

- Time series data can exhibit a variety of patterns, and it is often helpful to split a time series into several components, each representing an underlying pattern category.
- We discussed three types of time series patterns: trend, seasonality and cycles.
- We usually combine the trend and cycle into a single *trend-cycle* component (often just called the trend for simplicity).
- We can think of a time series as comprising three components:

1. a trend-cycle component
2. a seasonal component (or multiple),
3. and a remainder component (containing anything else in the time series).

- We will see the most common methods for extracting these components from a time series.
- Often this is done to help improve understanding of the time series, but it can also be used to improve forecast accuracy.

Additive and Multiplicative Decompositions

- Additive decomposition:

$$y_t = S_t + T_t + R_t$$

- Multiplicative decomposition:

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

- S_t is the seasonal component, T_t is the trend-cycle component and R_t is the remainder component, all at time t.
- The *additive decomposition* is the most appropriate if the magnitude of the seasonal fluctuations, or the variation around the trend-cycle, *does not vary* with the level of the time series.
- The *multiplicative decomposition* is more appropriate when there is *variation* in the seasonal pattern or around the trend-cycle.

Additive and Multiplicative Decompositions

- An alternative to using a multiplicative decomposition is to first transform the data until the series appears stable and then perform an additive decomposition.
- What happens when you use a log transformation:

$$\log y_t = \log S_t + \log T_t + \log R_t$$

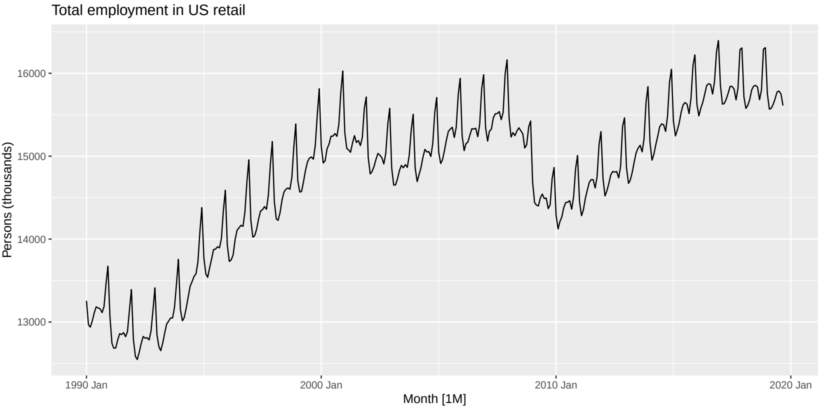
Example: Employment in US Retail

- The data shows the total monthly number of persons in thousands employed in the retail sector across the US since 1990.

```
us_retail_employment <- us_employment %>%
  filter(year(Month) >= 1990, Title == "Retail Trade") %>%
  select(-Series_ID)

autoplot(us_retail_employment, Employed) +
  labs(y = "Persons (thousands)",
       title = "Total employment in US retail")
```

Example: Employment in US Retail



Example: Employment in US Retail

```
dcmp <- us_retail_employment %>%
  model(stl = STL(Employed))

components(dcmp) %>% head
```

```
## # A dable: 6 x 7 [1M]
## # Key:     .model [1]
## # :        Employed = trend + season_year + remainder
##   .model    Month Employed  trend season_year remainder season_adjust
##   <chr>     <mth>    <dbl>  <dbl>       <dbl>     <dbl>         <dbl>
## 1 stl    1990 Jan   13256. 13288.       -33.0     0.836        13289.
## 2 stl    1990 Feb   12966. 13269.      -258.    -44.6          13224.
## 3 stl    1990 Mar   12938. 13250.      -290.    -22.1          13228.
## 4 stl    1990 Apr   13012. 13231.      -220.      1.05         13232.
## 5 stl    1990 May   13108. 13211.      -114.     11.3          13223.
## 6 stl    1990 Jun   13183. 13192.       -24.3    15.5          13207.
```

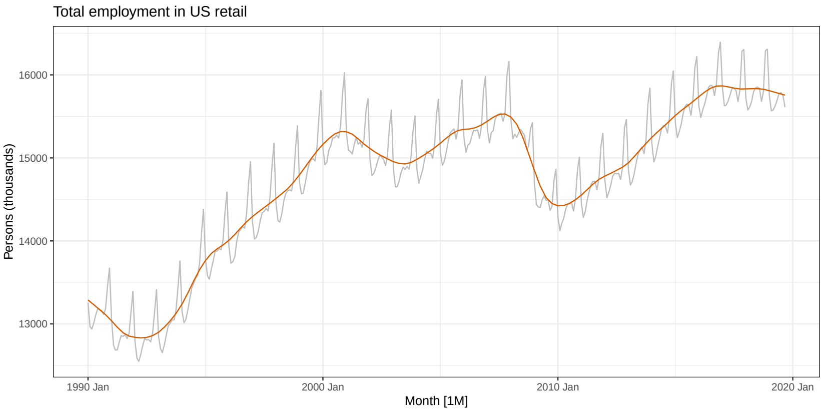
- The output above shows the components of an additive STL decomposition.
- The original data is shown (as Employed), followed by the estimated components.
- This output forms a "decomposition table", or a `dable`.

Example: Employment in US Retail

- The trend column contains the trend-cycle component T_t, and follows the overall movement of the series, ignoring seasonality and random fluctuations

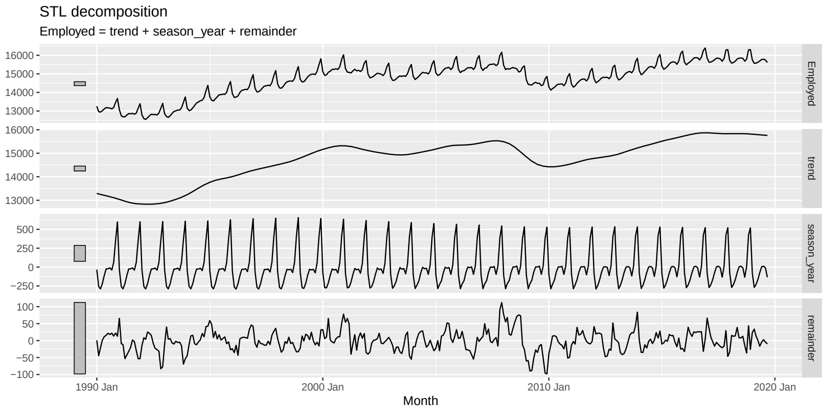
```
components(dcmp) %>%
  as_tsibble() %>%
  autoplot(Employed, colour="gray") +
  theme_bw() +
  geom_line(aes(y = trend), colour = "#D55E00") +
  labs(y = "Persons (thousands)",
       title = "Total employment in US retail")
```

Example: Employment in US Retail



Example: Employment in US Retail

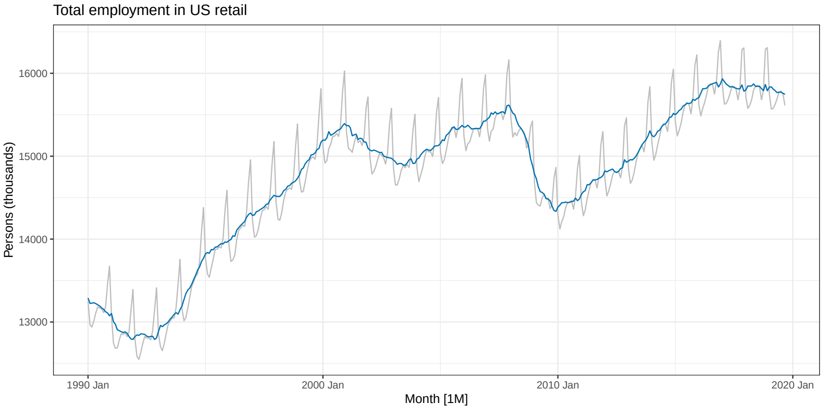
```
components(dcmp) %>% autoplot
```



Seasonally Adjusted Data

- If we remove S_t from the time series, we obtain the "seasonally adjusted" data.
- For an additive decomposition it is obtained using $y_t - S_t$
- For a multiplicative decomposition it is obtained using y_t/S_t

Seasonally Adjusted Data



Seasonally Adjusted Data

- If the variation due to seasonality is not of primary interest, the seasonally adjusted series can be useful.
- E.g., monthly unemployment data are usually seasonally adjusted in order to highlight variation due to the underlying state of the economy rather than the seasonal variation.
- An increase in unemployment due to school leavers seeking work is seasonal variation, while an increase in unemployment due to an economic recession is non-seasonal.
- Seasonally adjusted series contain the remainder component as well as the trend-cycle, therefore, they are not "smooth".
- If the purpose is to look for turning points in a series, and interpret any changes in direction, then it is better to use the trend-cycle component only.

Moving Averages

- The classical method of time series decomposition 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 understand how it works.
- The *first step* in a classical decomposition is to use a **moving average** method to estimate the trend-cycle.

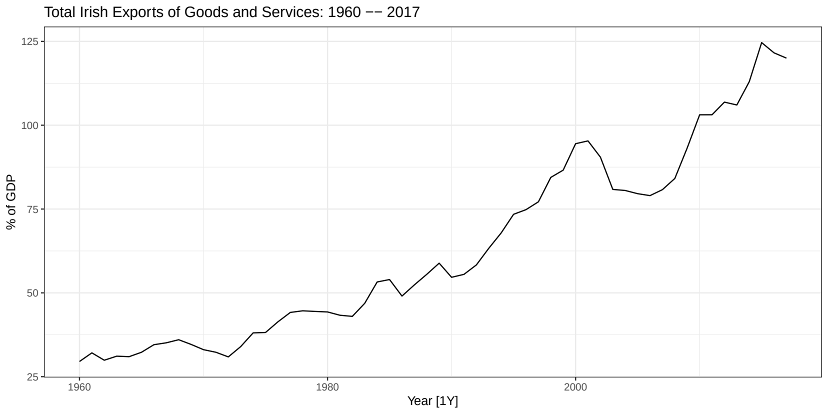
Moving Averages

- A moving average of order m can be written as

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

where $m = 2k + 1$
- In words, the estimate of the trend-cycle component is obtained by averaging values of the time series within k periods of t.
- The averaging eliminates some of the randomness, leaving a smooth trend-cycle component.
- We denote a "moving average of order m" by $m-$MA

Example: Irish Exports



Moving Averages

- Simple moving averages such as these are usually of an odd order (e.g., 3, 5, 7, etc.).
- This is so they are symmetric: in a moving average of order $m = 2k + 1$, the middle observation, and k observations on either side, are averaged. But if m was even, it would no longer be symmetric.
- To correct for the lack of symmetry, it is possible to apply a moving average to a moving average.
- For example, we might take a moving average of order 4, and then apply another moving average of order 2 to the results.
- Example: Australian quarterly beer production

Moving Averages

- The notation $2 \times 4-$MA means a $4-$MA followed by a $2-$MA.
- When a $2-$MA ollows 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.
- We can write the $2 \times 4-$MA as follows:

$$\begin{aligned}\hat{T}_y &= \frac{1}{2}\left[\frac{1}{4}(y_{t-2}+y_{t-1}+y_t+y_{t+1})+\frac{1}{4}(y_{t-1}+y_t+y_{t+1}+y_{t+2})\right] \\ &= \frac{1}{8}y_{t-2}+\frac{1}{4}y_{t-1}+\frac{1}{4}y_t+\frac{1}{4}y_{t+1}+\frac{1}{8}y_{t+2}\end{aligned}$$

- This is a special case of a *weighted moving average*:

$$\hat{T}_t = \sum_{j=-k}^{k} a_j y_{t+j},$$

where the weights a_j sum to one and are symmetric.

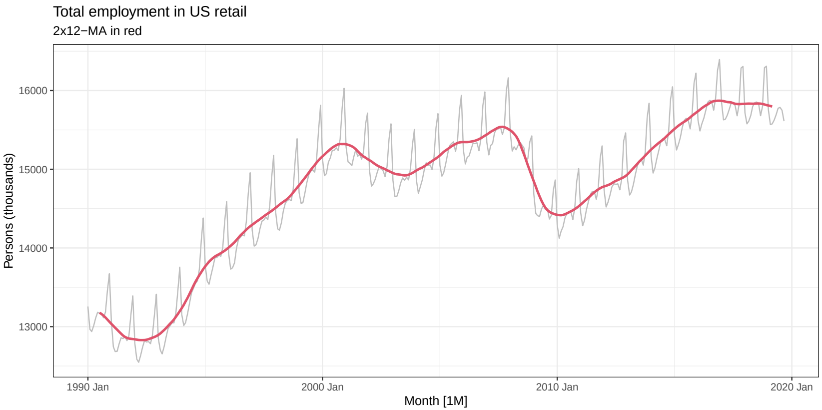
Moving Averages

- Centred moving averages are useful to estimate trend-cycle for seasonal data. Take 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 will get the same weight (the first and last terms are the same quarter for two consecutive years)
- In general, a $2 \times m-$MA is equivalent to a weighted $(m+1)-$MA where all observations take weight $1/m$, except the first and last, which take weight $1/(2m)$
- This is useful to estimate the trend-cycle for seasonal data with period m.
- What MA would you use for monthly data with annual seasonality?
- What MA would you use for daily data with weekly seasonality?

Example: Employment in the US Retail Sector



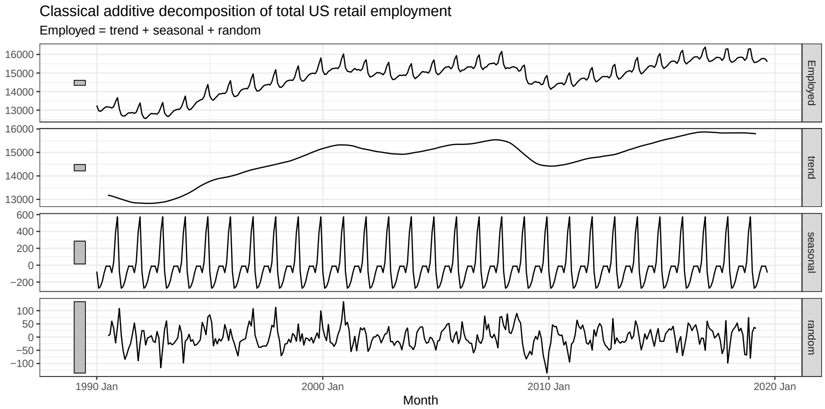
Classical Decomposition

- This method originated in the 1920s.
- We describe the additive and multiplicative decomposition of 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).

Classical Decomposition: Additive

- **Step 1**: If m is even, compute the trend-cycle component $\hat{T}_t$ using a $2 \times m-$MA. If m is odd, use a $m-$MA.
- **Step 2**: Calculate the detrended series $y_t - \hat{T}_t$
- **Step 3**: Estimate the seasonal component by averaging the detrended values for that season. E.g. for monthly data, the seasonal component for March is the average of all the detrended March values in the data. Then adjust the seasonal component values such that they sum to zero. Then replicate the sequence for each year of data. This gives $\hat{S}_t$.
- **Step 4**: Calculate the remainder $\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$

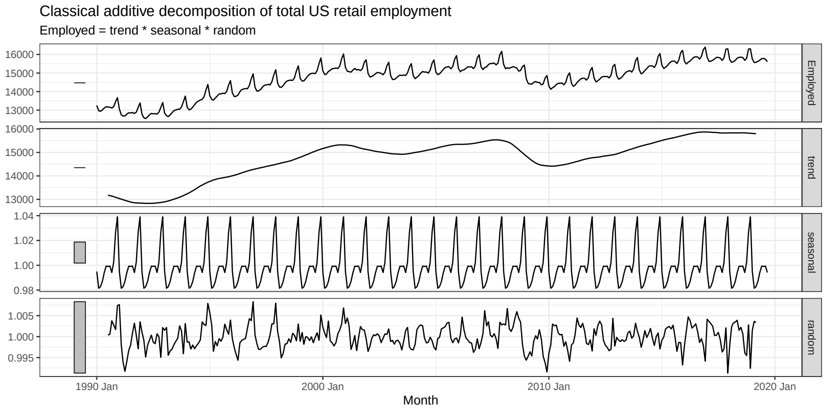
Classical Decomposition: Additive



Classical Decomposition: Multiplicative

- **Step 1**: If m is even, compute the trend-cycle component $\hat{T}_t$ using a $2 \times m-$MA. If m is odd, use a $m-$MA.
- **Step 2**: Calculate the detrended series $y_t/\hat{T}_t$
- **Step 3**: Estimate the seasonal component by averaging the detrended values for that season. E.g. for monthly data, the seasonal component for March is the average of all the detrended March values in the data. Then adjust the seasonal component values such that they add to m. Then replicate the sequence for each year of data. This gives $\hat{S}_t$.
- **Step 4**: Calculate the remainder $\hat{R}_t = y_t/(\hat{T}_t\hat{S}_t)$

Classical Decomposition: Multiplicative



Classical Decomposition

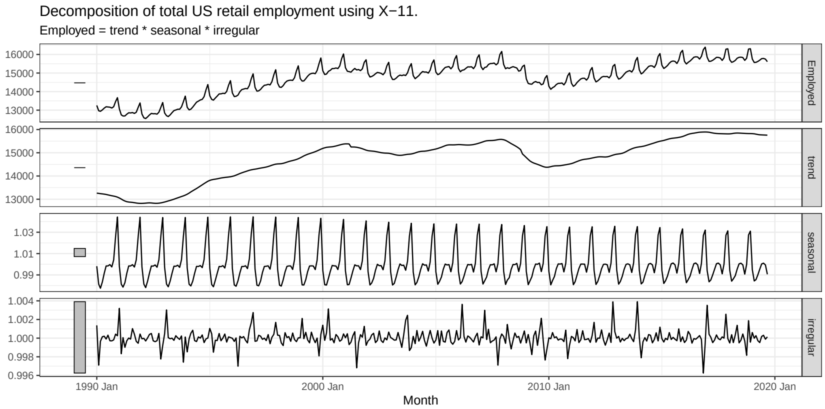
While classical decomposition is still widely used, it is not recommended, as there are now several much better methods. Some of the problems with classical decomposition are summarised below.

- The estimate of the trend-cycle is unavailable for the first few and last few observations. Consequently, there is also no estimate of the remainder component for the same time periods.
- The trend-cycle estimate tends to over-smooth rapid rises and falls in the data.
- Classical decomposition methods assume that the seasonal component repeats from year to year. For many series, this is a reasonable assumption, but for some longer series it is not. E.g. electricity demand over the last 100 years.
- Occasionally, the values of the time series in a small number of periods may be particularly unusual. For example, the monthly air passenger traffic may be affected by an industrial dispute, making the traffic during the dispute different from usual. The classical method is not robust to these kinds of unusual values.

Other Decomposition Methods: X-11

- Developed by Statistics Canada
- Trend-cycle estimates available for all observations, including end-points
- Seasonal component allowed to vary over time
- Robust to outliers and level shifts in the time series

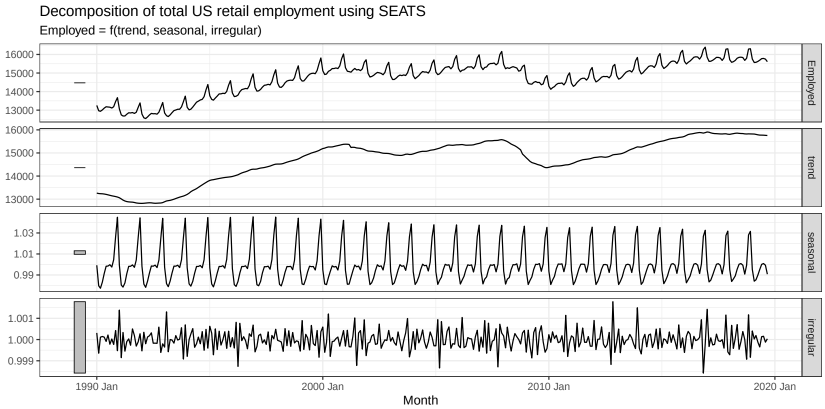
Example: X-11 decomposition



Other Decomposition Methods: SEATS

- SEATS stands for "Seasonal Extraction in ARIMA Time Series"
- Developed by the Bank of Spain
- Widely used by government agencies around the world

Example: SEATS decomposition



Other Decomposition Methods: STL

- STL stands for "Seasonal and Trend Decomposition using Loess"
- STL has several advantages over classical decomposition, X-11 and SEATS
- STL will handle any type of seasonality (not only monthly or quarterly data)
- The seasonal component is allowed to change over time, and the rate of change can be controlled by the user
- The smoothness of the trend-cycle component can also be controlled by the user
- Robust to outliers

Example: STL decomposition