A time series can be thought of as a list of numbers (the measurements), along with some information about what times those numbers were recorded (the index). This information can be stored as a tsibble object in R.

y <- tsibble(

Year = 2015:2019,

Observation = c(123, 39, 78, 52, 110),

index = Year

)

tsibble objects extend tidy data frames (tibble objects) by introducing temporal structure. We have set the time series index to be the Year column, which associates the measurements (Observation) with the time of recording (Year).

For observations that are more frequent than once per year, we need to use a time class function on the index. For example, suppose we have a monthly dataset z:

z

#> # A tibble: 5 × 2

#> Month Observation

#> <chr> <dbl>

#> 1 2019 Jan 50

#> 2 2019 Feb 23

#> 3 2019 Mar 34

#> 4 2019 Apr 30

#> 5 2019 May 25

This can be converted to a tsibble object using the following code:

z %>%

mutate(Month = yearmonth(Month)) %>%

as\_tsibble(index = Month)

#> # A tsibble: 5 x 2 [1M]

#> Month Observation

#> <mth> <dbl>

#> 1 2019 Jan 50

#> 2 2019 Feb 23

#> 3 2019 Mar 34

#> 4 2019 Apr 30

#> 5 2019 May 25

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.

We can use dplyr functions such as mutate(), filter(), select() and summarise() to work with tsibble objects. To illustrate these, we will use the PBS tsibble containing sales data on pharmaceutical products in Australia.

The select() function allows us to select particular columns, while filter() allows us to keep particular rows.

PBS %>%

filter(ATC2 == "A10") %>%

select(Month, Concession, Type, Cost)

other useful function is summarise() which allows us to combine data across keys. For example, we may wish to compute total cost per month regardless of the Concession or Type keys.

PBS %>%

filter(ATC2 == "A10") %>%

select(Month, Concession, Type, Cost) %>%

summarise(TotalC = sum(Cost))

We can create new variables using the mutate() function. Here we change the units from dollars to millions of dollars:

PBS %>%

filter(ATC2 == "A10") %>%

select(Month, Concession, Type, Cost) %>%

summarise(TotalC = sum(Cost)) %>%

mutate(Cost = TotalC/1e6)

Finally, we will save the resulting tsibble for examples later in this chapter.

PBS %>%

filter(ATC2 == "A10") %>%

select(Month, Concession, Type, Cost) %>%

summarise(TotalC = sum(Cost)) %>%

mutate(Cost = TotalC / 1e6) -> a10

At the end of this series of piped functions, we have used a right assignment (->), which is not common in R code, but is convenient at the end of a long series of commands as it continues the flow of the code

prison <- prison %>%

mutate(Quarter = yearquarter(Date)) %>%

select(-Date) %>%

as\_tsibble(key = c(State, Gender, Legal, Indigenous),

index = Quarter)

For a tsibble to be valid, it requires a unique index for each combination of keys. The tsibble() or as\_tsibble() function will return an error if this is not true.

Some graphics and some models will use the seasonal period of the data. The seasonal period is the number of observations before the seasonal pattern repeats. In most cases, this will be automatically detected using the time index variable.

More complicated (and unusual) seasonal patterns can be specified using the period() function in the lubridate package.

**TIME PLOTS –**

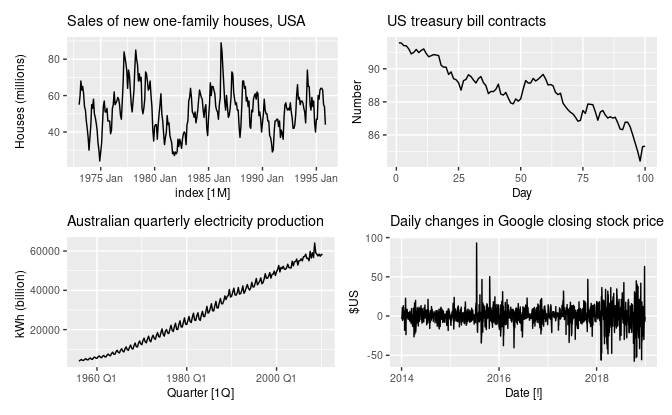
For time series data, the obvious graph to start with is a time plot. That is, the observations are plotted against the time of observation, with consecutive observations joined by straight line

We will use the autoplot() command frequently. It 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.

**TIME-SERIES PATTERNS**

Many people confuse cyclic behaviour with seasonal behaviour, but they are really quite different. If the fluctuations are not of a fixed frequency then they are cyclic; if the frequency is unchanging and associated with some aspect of the calendar, then the pattern is seasonal. In general, the average length of cycles is longer than the length of a seasonal pattern, and the magnitudes of cycles tend to be more variable than the magnitudes of seasonal patterns.

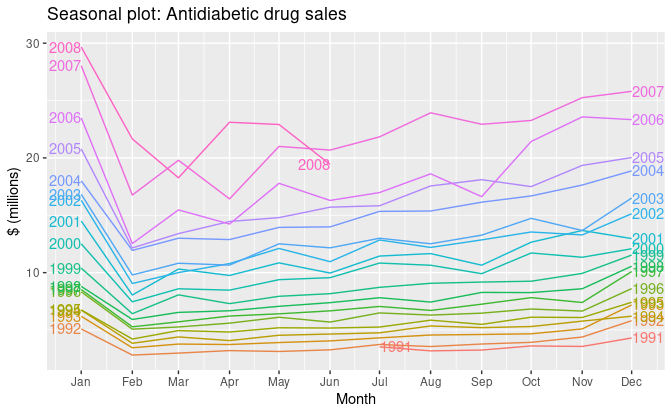
Many time series include trend, cycles and seasonality. When choosing a forecasting method, we will first need to identify the time series patterns in the data, and then choose a method that is able to capture the patterns properly.



1. 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.
2. 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.
3. The Australian quarterly electricity production (bottom left) shows a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behaviour here.
4. 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 PLOT**

A seasonal plot is similar to a time plot except that the data are plotted against the individual “seasons” in which the data were observed.



These are exactly the same data as were shown earlier, but now the data from each season are overlapped. 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.

**SUBSERIES SEASONAL PLOT – GG\_SUBSERIES()**

**a10 %>%**

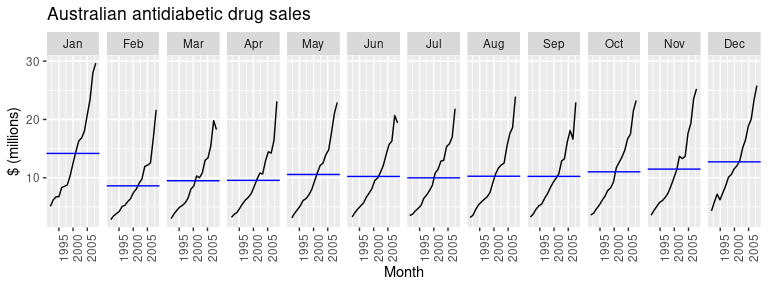
**gg\_subseries(Cost) +**

**labs(**

**y = "$ (millions)",**

**title = "Australian antidiabetic drug sales"**

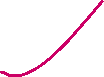
**)**



The blue horizontal lines indicate the means for each month. This form of plot enables the underlying seasonal pattern to be seen clearly, and also shows the changes in seasonality over time. It is especially useful in identifying changes within particular seasons. In this example, the plot is not particularly revealing; but in some cases, this is the most useful way of viewing seasonal changes over time.

**OBSERVATIONS REGARDING CORRELOGRAMS (ACFs againt Lag k) – Trend and Seasonality in Correlograms**

**When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in value. So the ACF of a trended time series tends to have positive values that slowly decrease as the lags increase.**

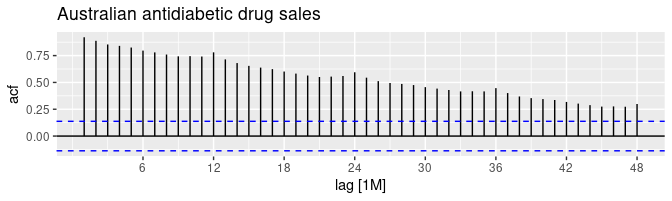


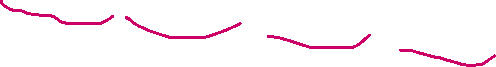
**When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal period) than for other lags.**



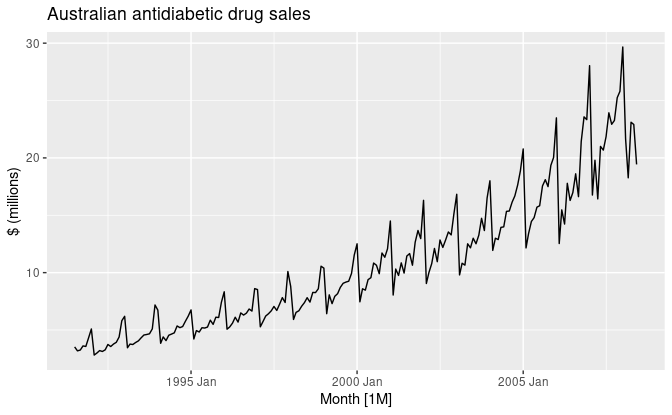
**When data are both trended and seasonal, you see a combination of these effects.**

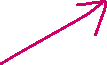






. The slow decrease in the ACF as the lags increase is due to the trend, while the “scalloped” shape is due to the seasonality.





**WHITE NOISE**

Time series that show no autocorrelation are called **white noise**.

For white noise series, we expect each autocorrelation to be close to zero. Of course, they will not be exactly equal to zero as there is some random variation. For a white noise series, we expect 95% of the spikes in the ACF to lie within ±2/√T±2/T where TT is the length of the time series. It is common to plot these bounds on a graph of the ACF (the blue dashed lines above). If one or more large spikes are outside these bounds, or if substantially more than 5% of spikes are outside these bounds, then the series is probably not white noise.