Ch2 | Time Series Graphics

Michael Rose

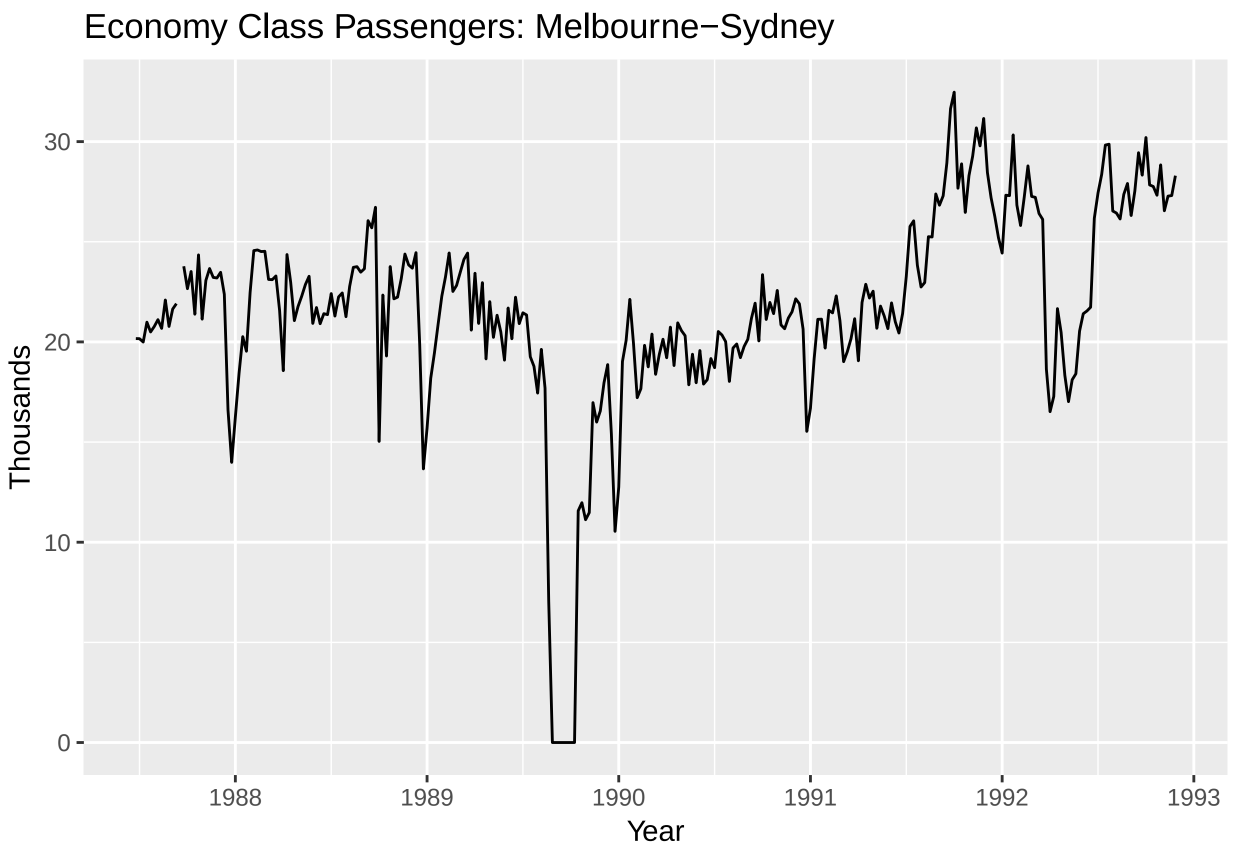
October 27, 2018

2.1 | ts Objects

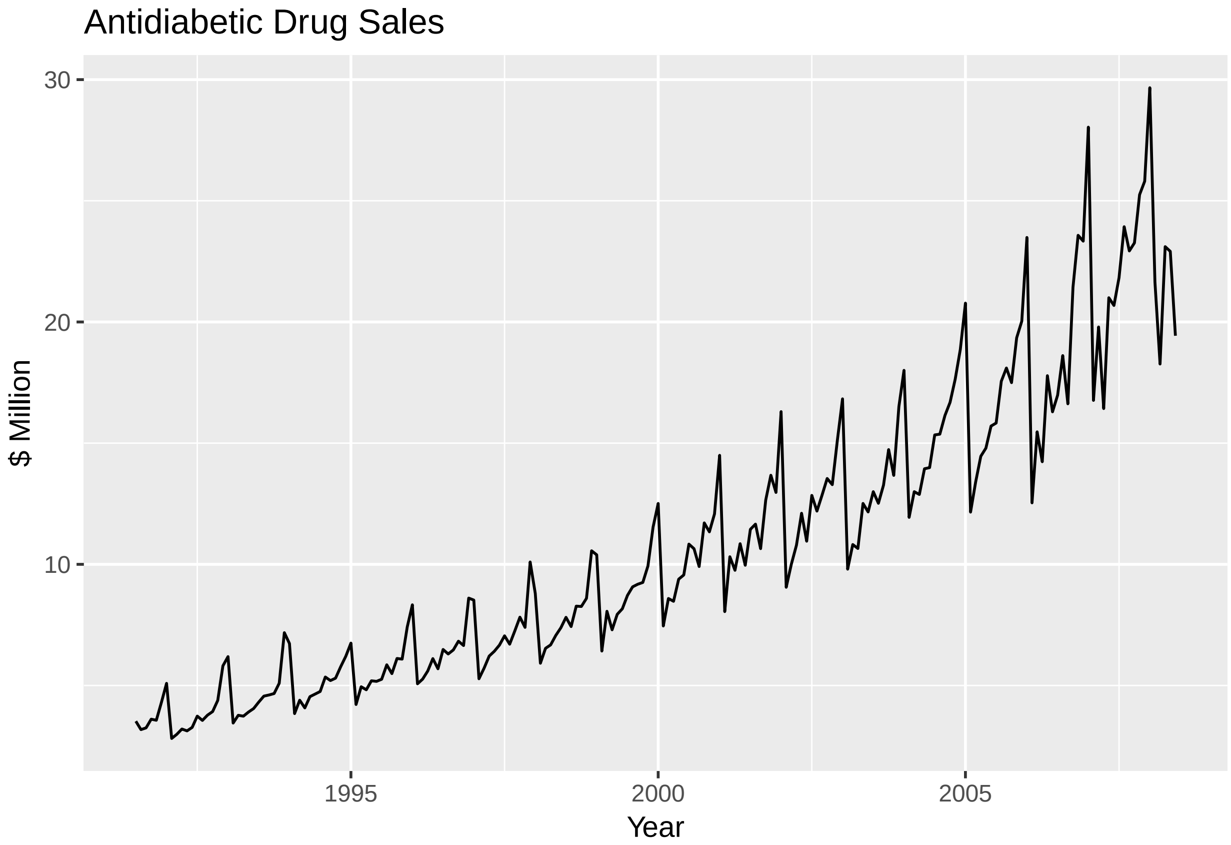
```r
# create a time series object
y <- ts(c(123, 39, 78, 52, 110), start = 2012)

# for observations more frequent than yearly, we can use the frequency argument

# generate some data
vec_length <- 15*12
z <- vector(mode = "numeric", length = vec_length)

for (i in seq_along(z)){
  z[i] <- rnorm(1, mean = 0, sd = 1)
}

# create a monthly data table as a ts object
y <- ts(z, start = 2003, frequency = 12)
```

2.2 | Time Plots

```r
# plot economy class passengers in melbourne-sydney flights
autoplot(melsyd[, "Economy.Class"]) +
  ggtitle("Economy Class Passengers: Melbourne-Sydney") +
  xlab("Year") + ylab("Thousands")
```



```
# antidiabetic drug sales
autoplot(a10) +
  ggtitle("Antidiabetic Drug Sales") +
  ylab("$ Million") + xlab("Year") +
  theme_gray()
```



2.3 | 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.

Seasonal

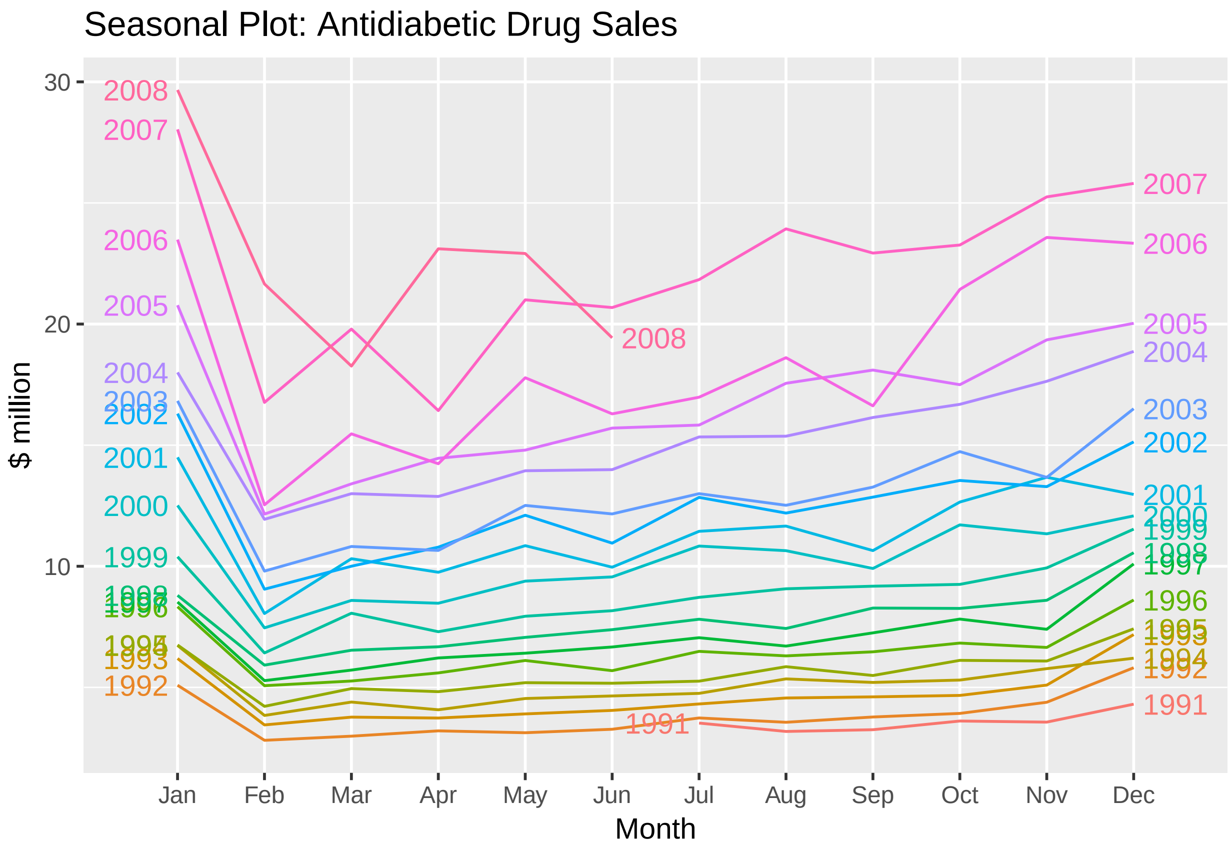
A *seasonal* pattern occurs when a time series is affected by seasonal factors such as the time of the year or day of the week. Seasonality is always of a fixed and known frequency.

Cyclic

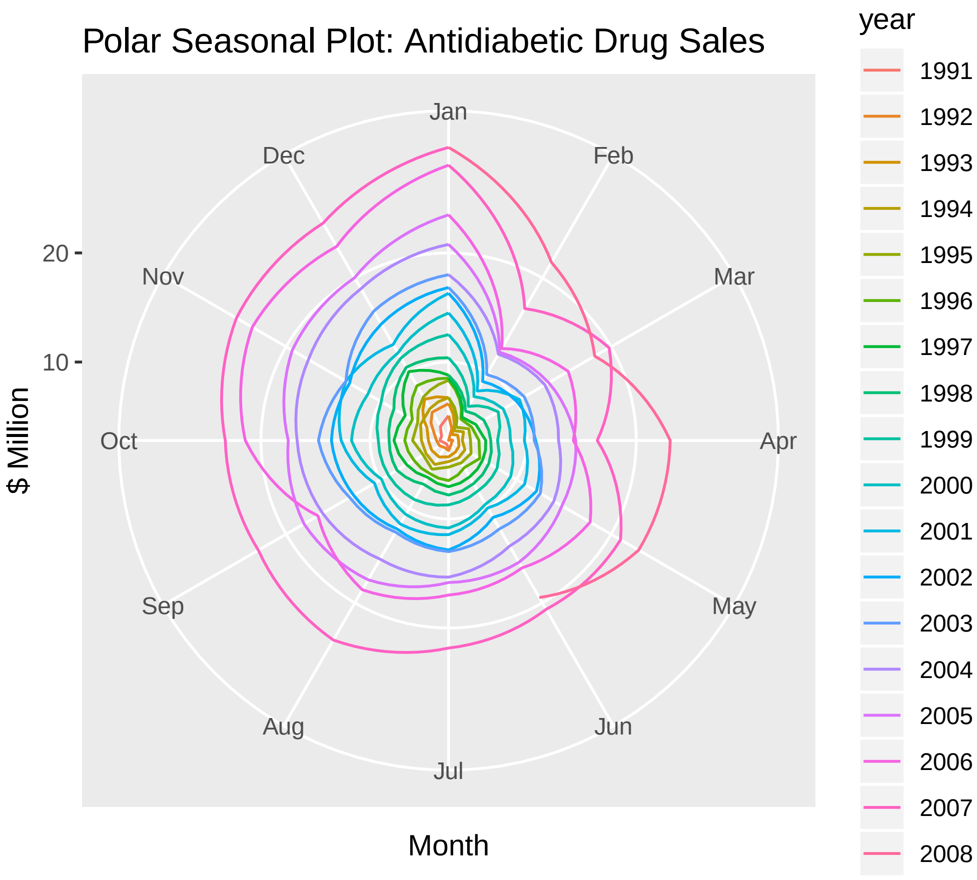
A *cycle* occurs when the data exhibit rises and falls that are not of a fixed frequency. These fluctuations are usually due to economic conditions, and are often related to the business cycle.

2.4 | Seasonal Plots

```r
ggseasonplot(a10, year.labels = TRUE, year.labels.left = TRUE) +
  ylab("$ million") +
  ggtitle("Seasonal Plot: Antidiabetic Drug Sales")
```



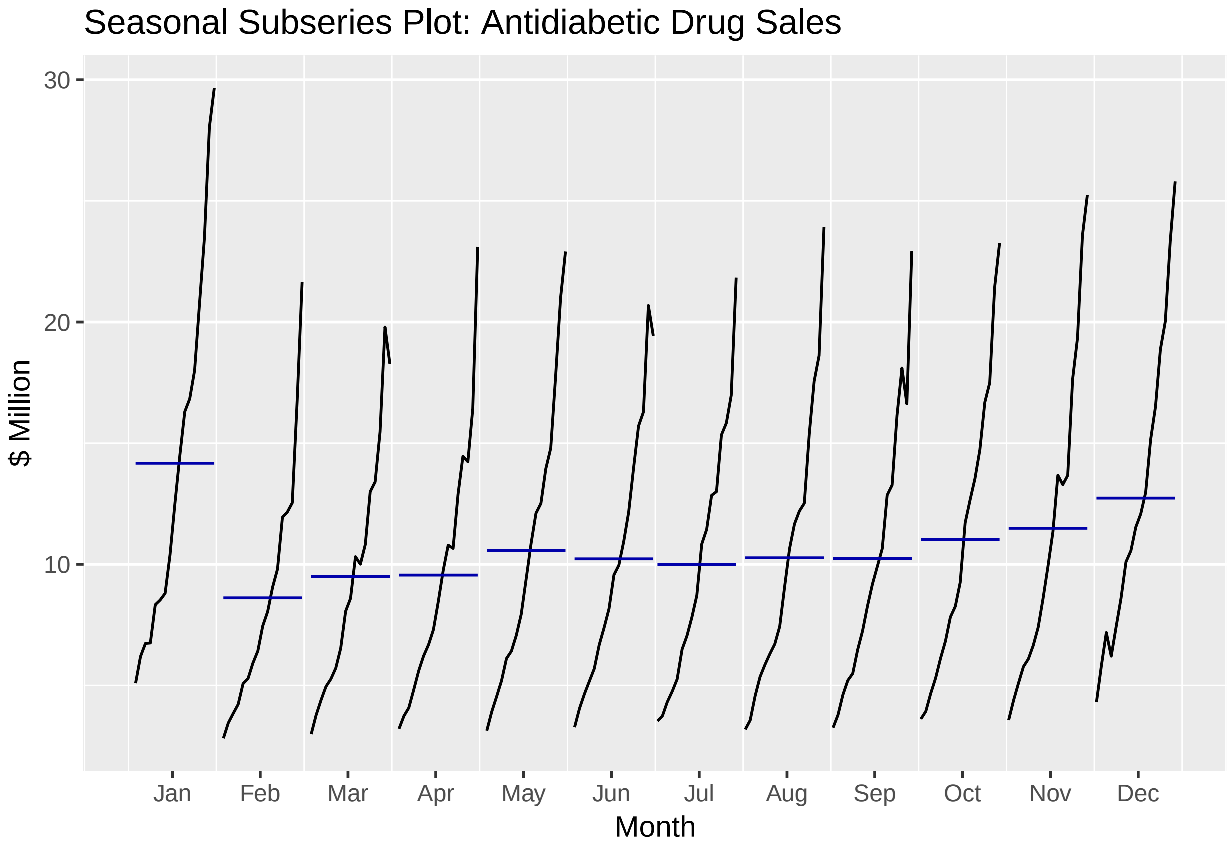
```r
ggseasonplot(a10, polar=TRUE) +
  ylab("$ Million") +
  ggtitle("Polar Seasonal Plot: Antidiabetic Drug Sales")
```



2.5 | Seasonal Subseries Plots

An alternative plot that emphasizes the seasonal patterns is where the data for each season are collected together in seperate mini time plots

```r
ggsubseriesplot(a10) +
  ylab("$ Million") +
  ggtitle("Seasonal Subseries Plot: Antidiabetic Drug Sales")
```



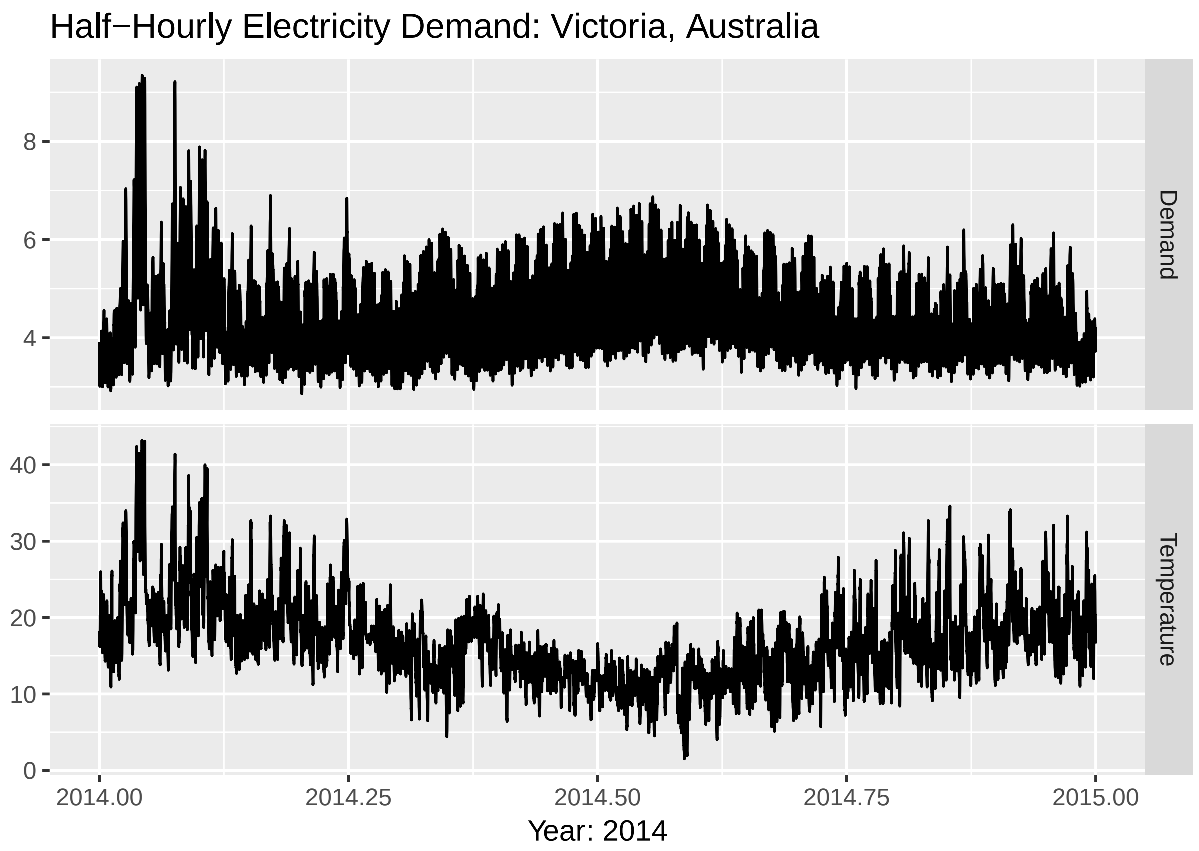
The horizontal lines indicate the means for each month. This plot allows the underlying seasonal pattern to be seen clearly, and shows the change in seasonality over time.

2.6 | Scatterplots

The plots below show two time serieS:

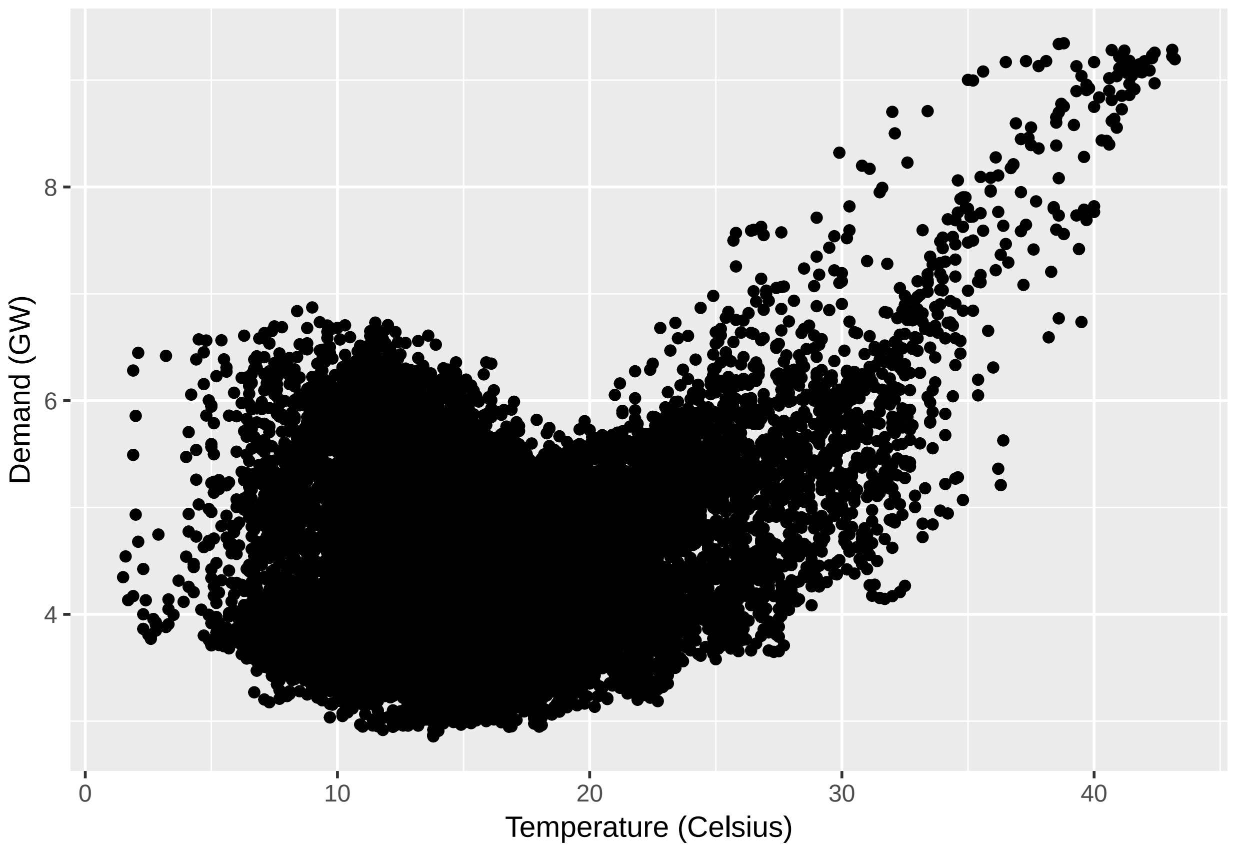
The half hourly electricity demand (in gigawatts) and temperature(in degrees celsius) for 2014 in victoria, australia.

```
autoplot(elecdemand[, c("Demand", "Temperature")], facets = TRUE) +
  xlab("Year: 2014") + ylab("") +
  ggtitle("Half-Hourly Electricity Demand: Victoria, Australia")
```



We can study the relationship between demand and temperature by plotting the series against each other

```
qplot(Temperature, Demand, data = as.data.frame(elecdemand)) +
  ylab("Demand (GW)") + xlab("Temperature (Celsius)")
```



We can see from the scatterplot above that there is high demand for electricity when temperatures are high (likely due to air conditioning). There is also a slight heating effect for very low temperatures.

Correlation

It is common to compute correlation coefficients to measure the strength of the relationshop between two variables. The correlation between some variables x, y is given by

$$r = \frac{\sum(x_t - \bar{x}))(y_t - \bar{y})}{\sqrt(\sum(x_t - \bar{x})^2)(\sqrt(\sum(y_t - \bar{y}^2))}.$$

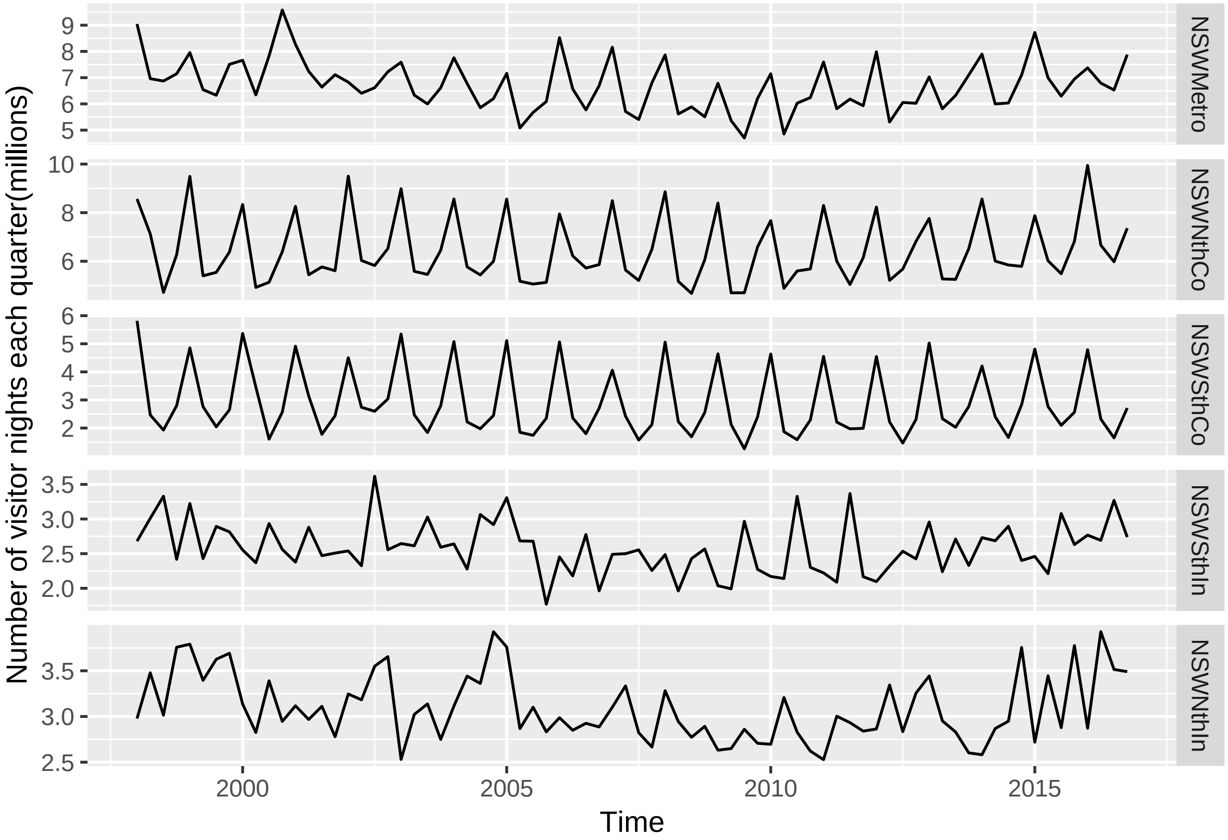
The correlation coefficient $r \in [-1, 1]$ with negative values representing a negative relationship and positive values showing a positive relationship. An important thing to note is that r only measures the strength of the *linear* coefficient.

Scatterplot Matrices

When there are several potential predictor variables, it is useful to plot each variable against each other variable.

Below are 5 time series showing quarterly visitor numbers for five regions of New South Wales, Australia

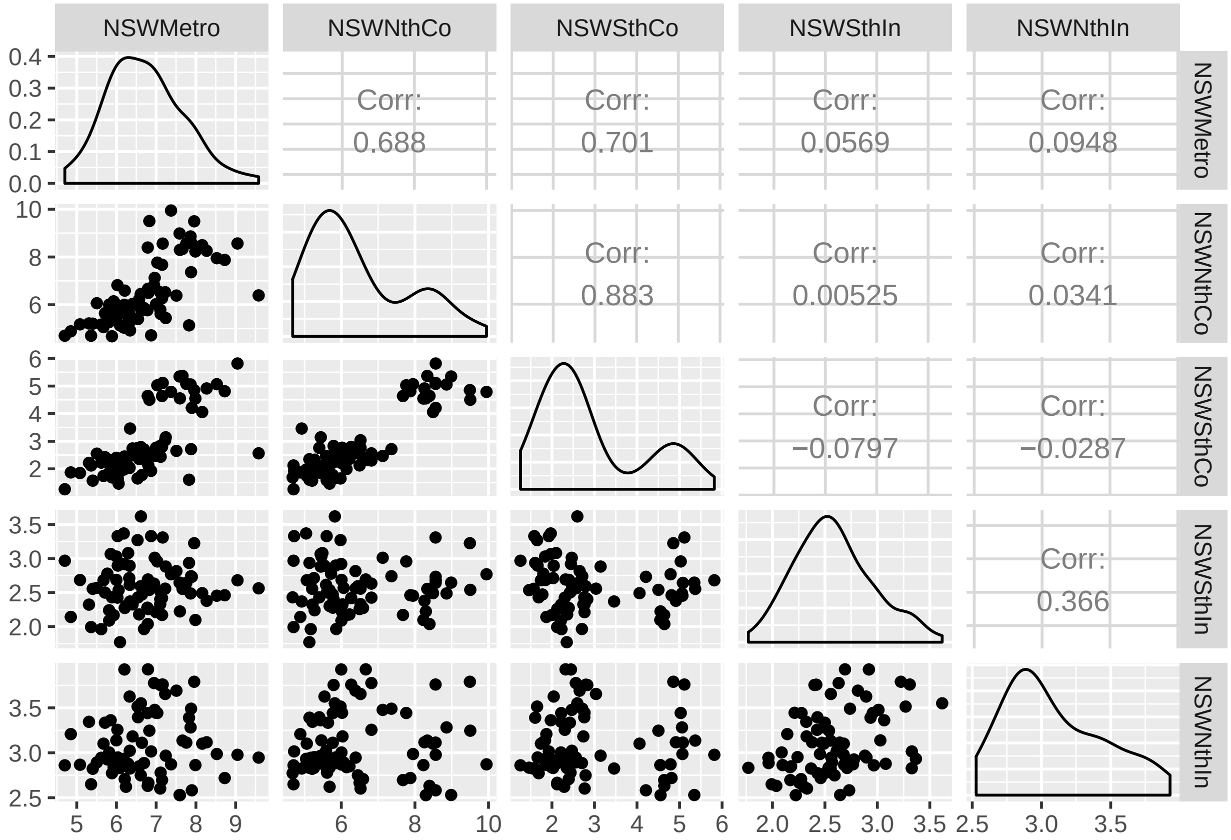
```r
autoplot(visnights[, 1:5], facets = TRUE) +
  ylab("Number of visitor nights each quarter(millions)")
```



To see the relationships between these five time series, we can plot each time series against the others.

We can put these in a scatterplot matrix:

```
GGally::ggpairs(as.data.frame(visnights[, 1:5]))
```



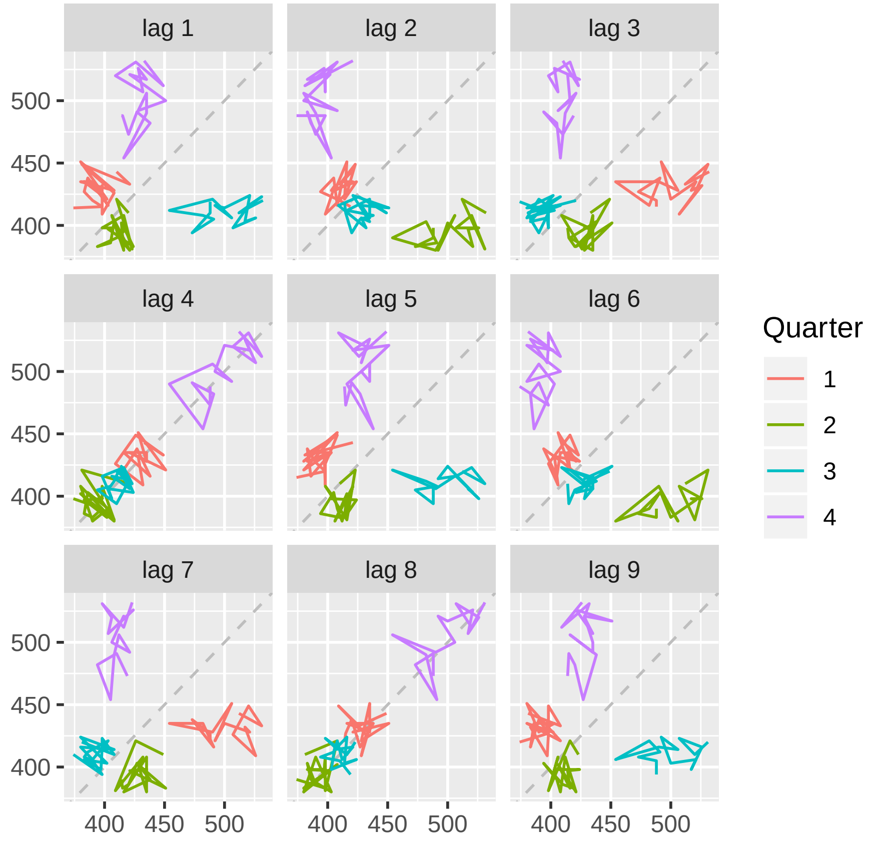
For each panel, the variable on the vertical axis is given by the variable name in that row, and the variable on the horizontal axis is given by the variable name in that column.

This matrix allows us to see all the relationships between pairs of variables easily. In this example, the second column of the plot shows us that there is a positive correlation between visitors to the NSW north coast and the NSW south coast. There is also one unusually high quarter for the NSW Metropolitan region, corresponding to the 2000 Sydney Olympics.

2.7 | Lag Plots

The figure below shows scatterplots of quarterly Australian beer production, where the horizontal axis shows lagged values of the time series. Each plot shows y_t plotted against y_{t-k} for different values of k.

```
beer2 <- window(ausbeer, start = 1992)
gglagplot(beer2)
```



Here the colors indicate the quarter of the variable on the veritcal axis. The lines connect points in chronological order. We cam see that the lags are strongly linear for lag 4 and 8, representting strong seasonality in the data. The negative relationship in lags 2 and 6 show because peaks in Q4 and plotted with troughs in Q2.

2.8 | Autocorrelation

Autocorrelation: or serial correlation, is the correlation of a signial with a delayed copy of itself as a function of delay. Informally, it is the similarity between observations as a function of the time lag between them. The analysis of autocorrelation is a tool for finding repeating patterns, generally obscured by noise.

Just as correlation measures the extent of a linear relationship between 2 variables, autocorrelation measures the linear relationship between *lagged* values of a time series.

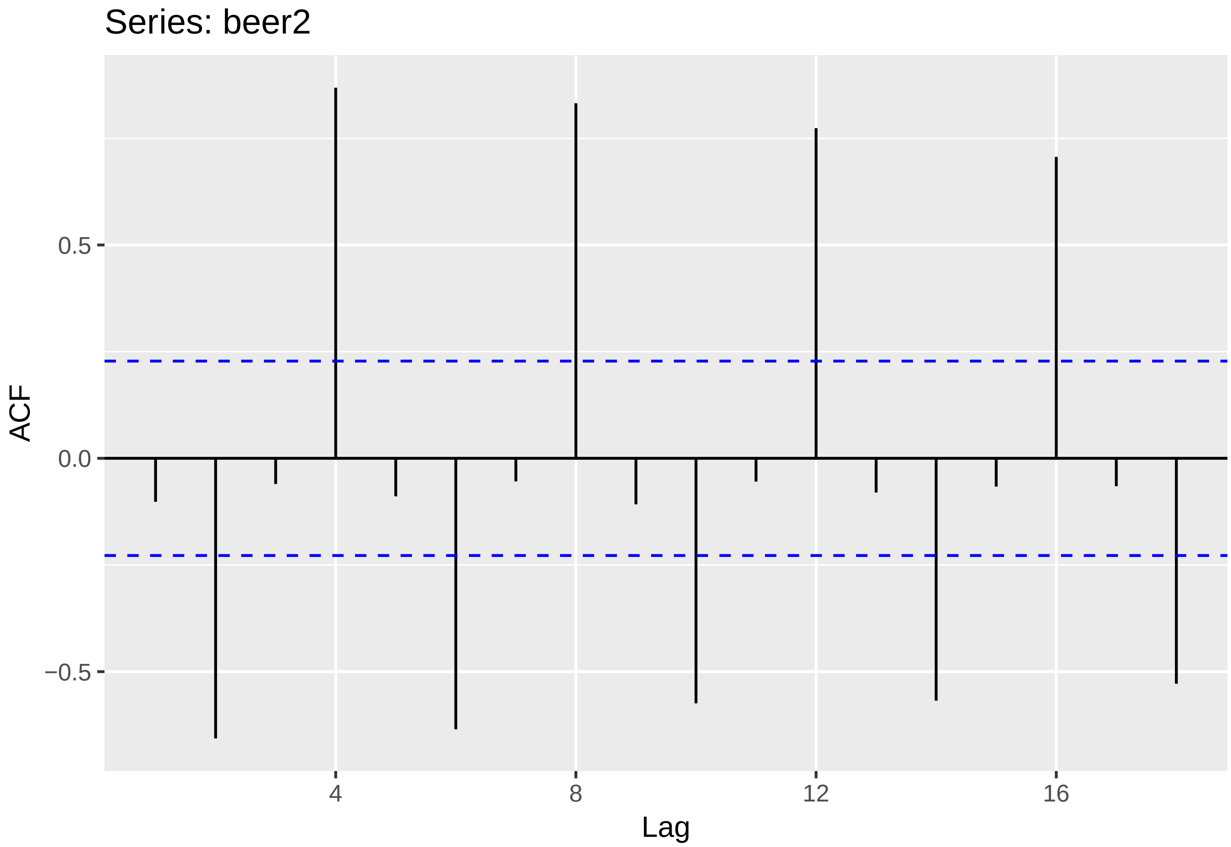
There are several autocorrelation coefficients, corresponding to each panel in the lag plot. For example, r_1 measures the relationship between y_t and y_{t-1}, r_2 y_t, y_{t-2}, etc.

r_k can be written as

$$r_k = \frac{\sum_{t=k+1}^{T}(y_t-\bar{y})(y_{t-k}-\bar{y})}{\sum_{t=1}^{T}(y_t-\bar{y})^2}$$

We can plot the autocorrelation coefficients to show the *autocorrelation function* or ACF. This plot is also known as a *correlogram*.

```
ggAcf(beer2)
```



In this graph we see that r_4 is the highest coefficient. This is due to the seasonal pattern in the data. The peaks tend to be 4 quarters apart and the troughs tend to be 4 quarters apart.

Similarly, r_2 is the most negative, as the peaks tend to come 2 quarters before troughs.

The dashed blue lines indicate whether the correlations are significantly different from 0.

Trend and Seasonality in ACF Plots

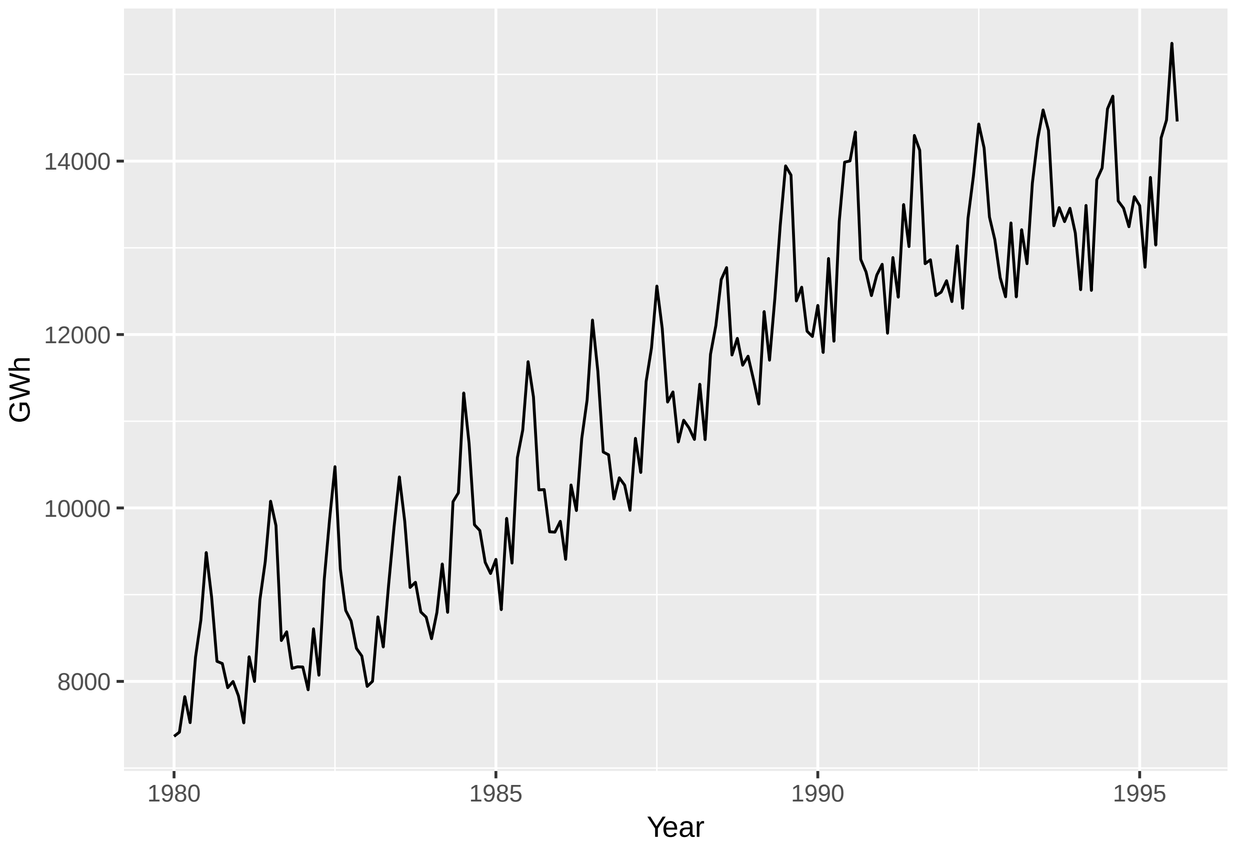
When there is a trend in data, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. As a result, the ACF is trended time series tend to have positive values that slowly decrease as the lags increase.

With seasonal data, the autocorrelations will be larger for seasonal lags (at multiples of the seasonal frequency) than for other lags.

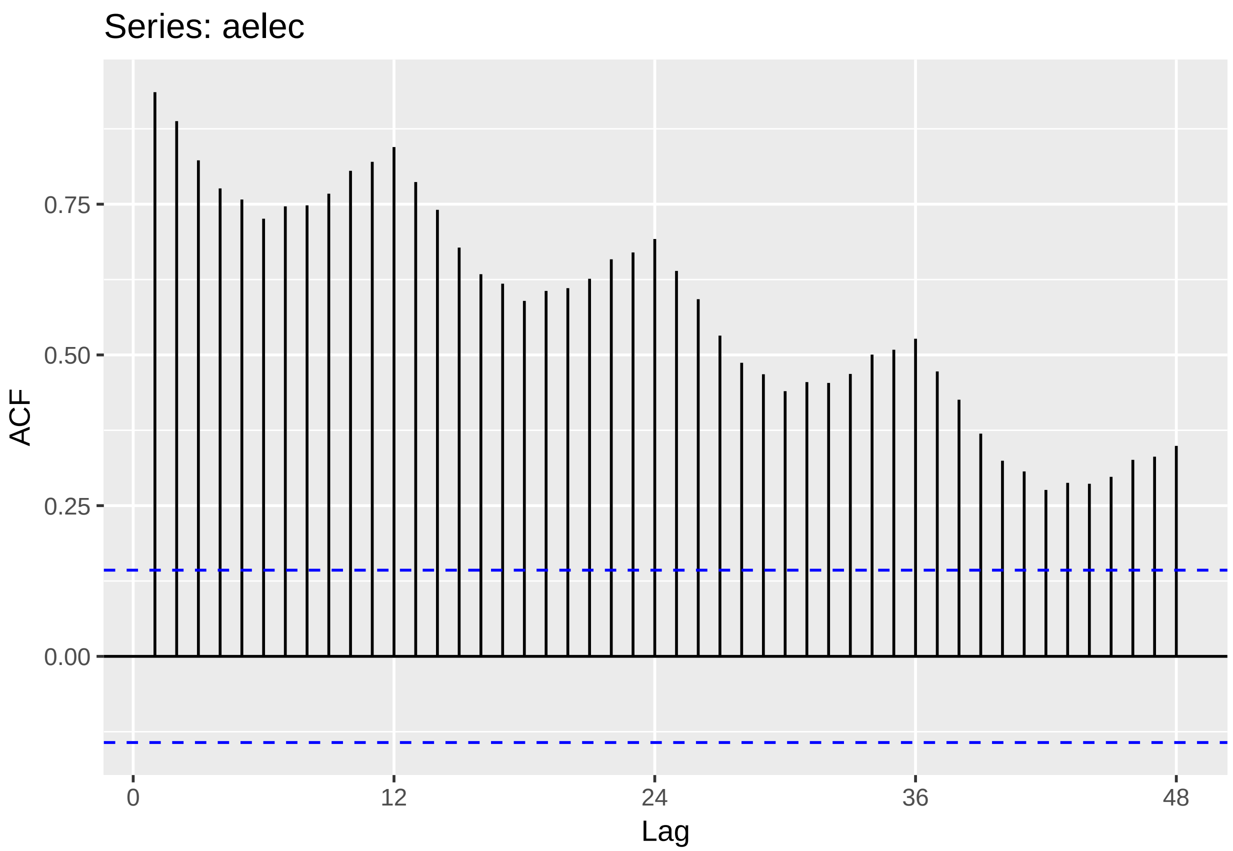
When data is both trended and seasonal, we see a combination.

In the plot below we see the Australian electricity demand series, showing both trend and seasonality

```r
aelec <- window(elec, start = 1980)
autoplot(aelec) + xlab("Year") + ylab("GWh")
```



```
ggAcf(aelec, lag = 48)
```



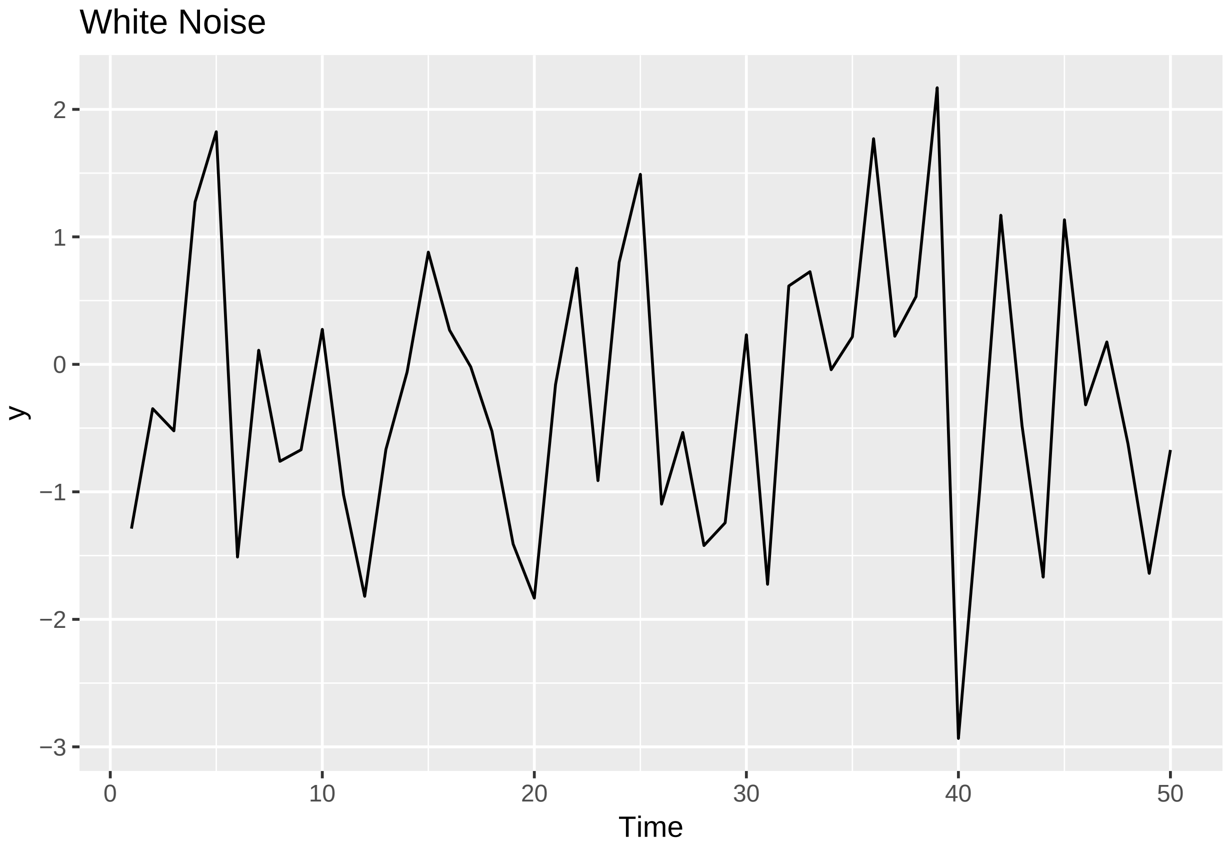
The slow decrease in the ACF as the lags increase is due to the trend, while the "scalloped" shape is due to the seasonality.

2.9 | White Noise

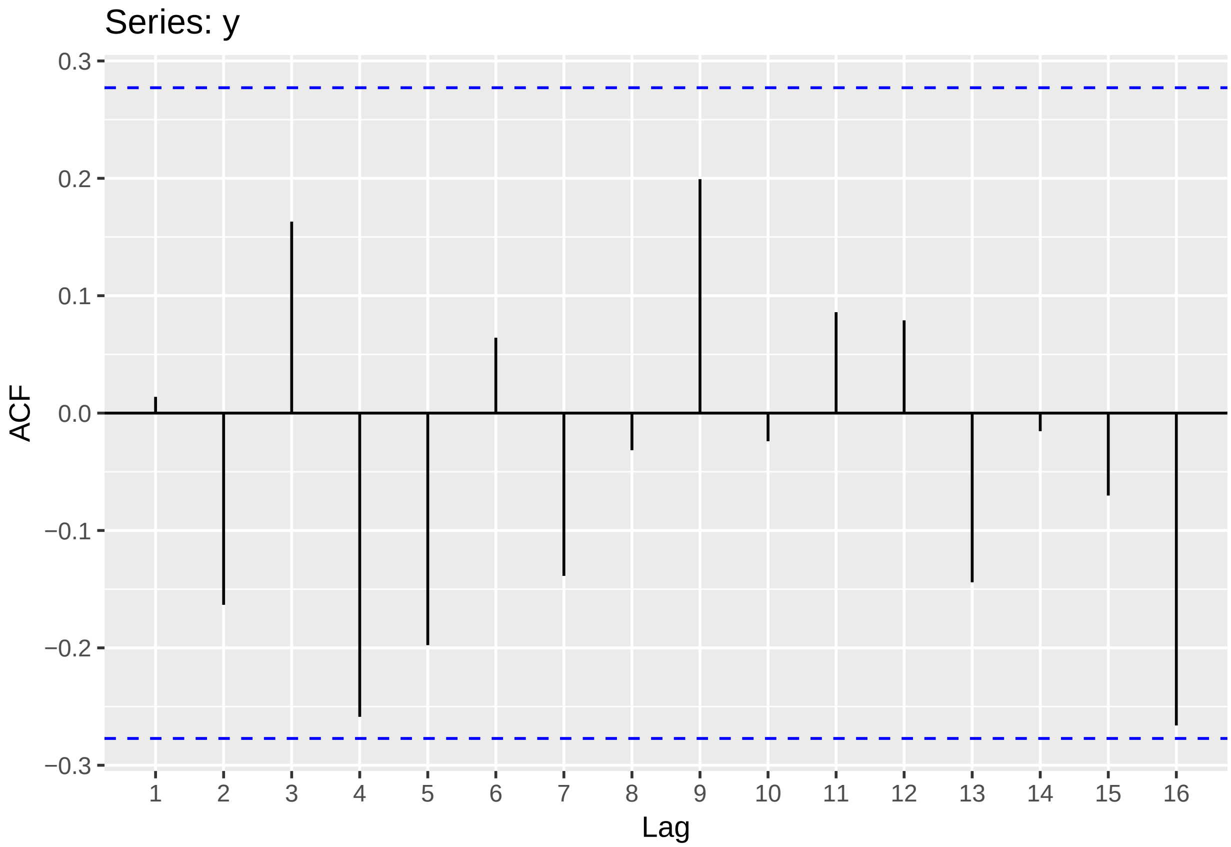
Time series with no autocorrelation are considered **white noise**.

Example:

```r
set.seed(30)
y <- ts(rnorm(50))
autoplot(y) + ggtitle("White Noise")
```



```
ggAcf(y)
```



We can see if a series is white noise by checking that 95% of the spikes in the ACF lie within +- 2 / $sqrt(T)$ where T is the length of the time series. In this example, $T = 50$, so the bounds are $+-2/\sqrt(50) = +-0.28$. All of the coefficients lie within these limits, confirming the data is white noise.