Ch2 | Time Series Graphics

Michael Rose

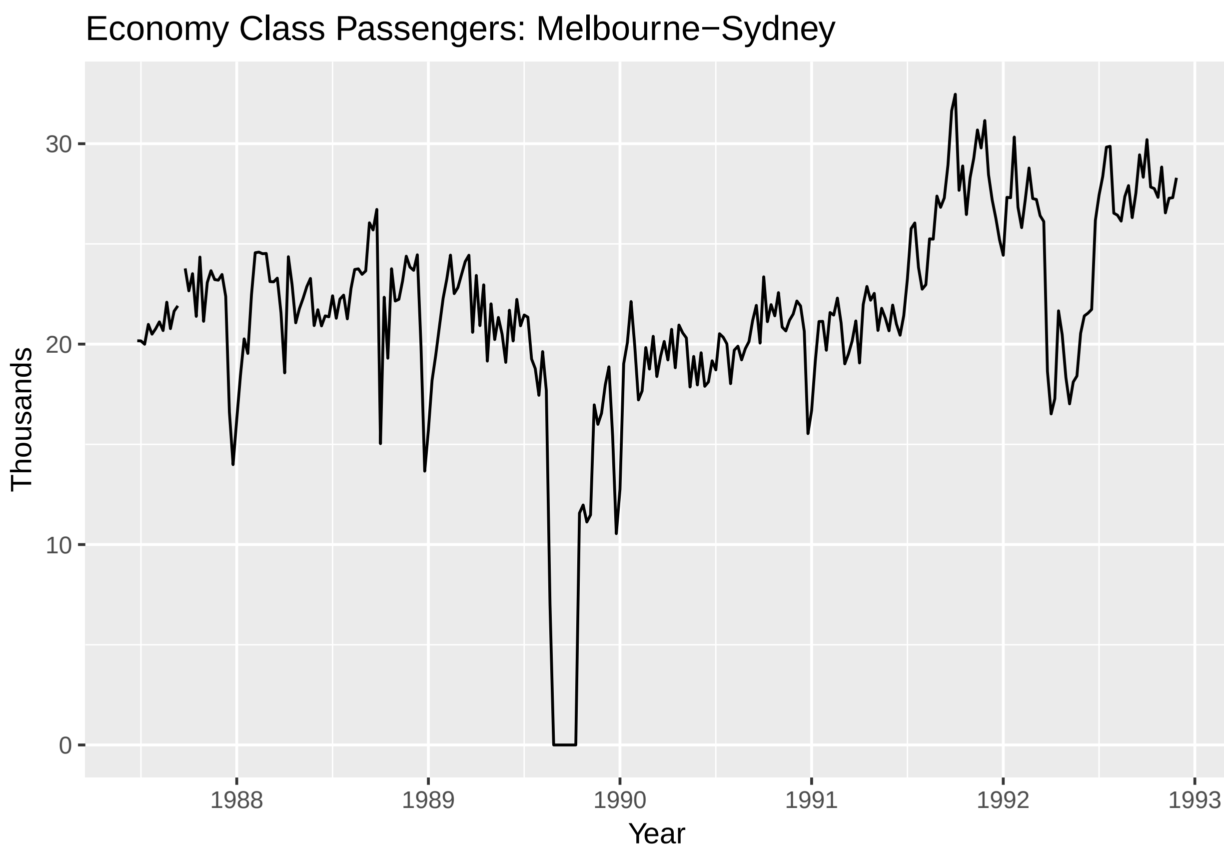
October 27, 2018

2.1 | ts Objects

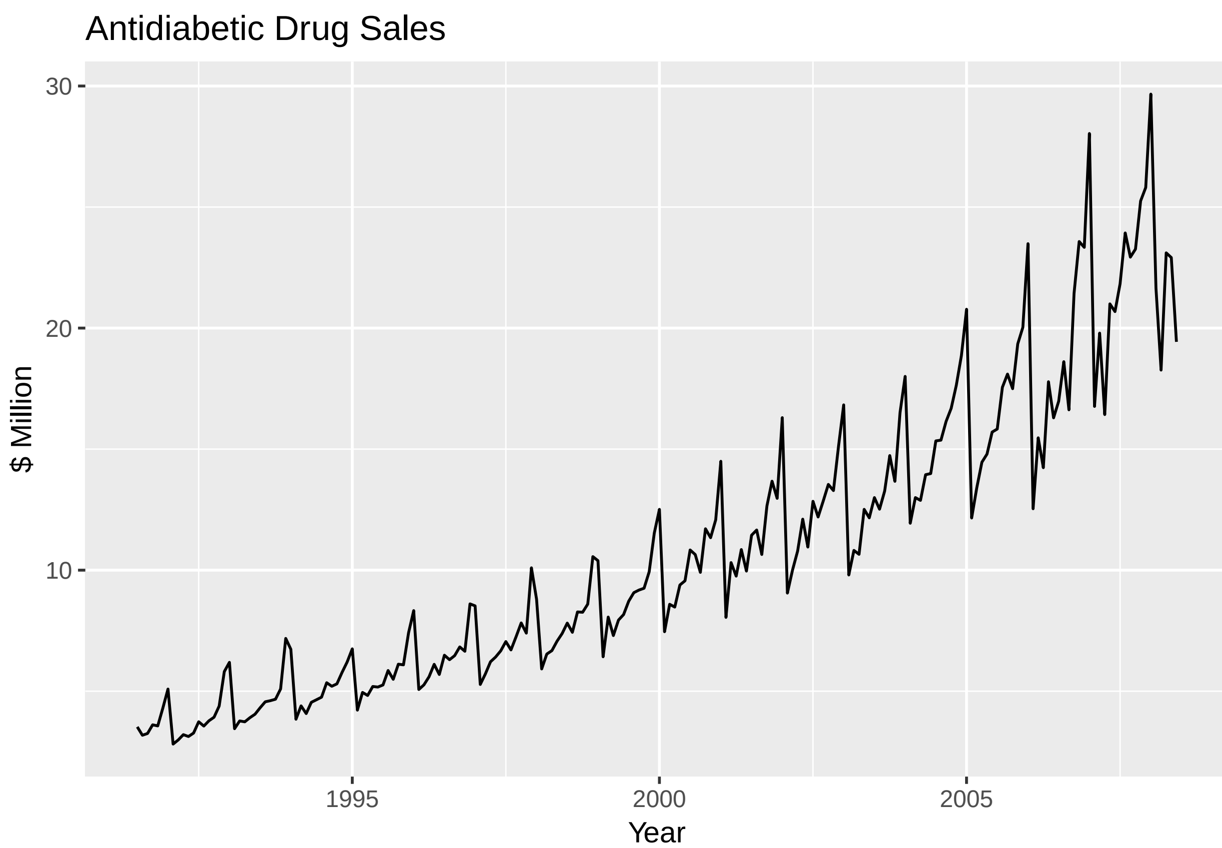
```
# create a time series object
y <- ts(c(123, 39, 78, 52, 110), start = 2012)

# 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

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




```r
# 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

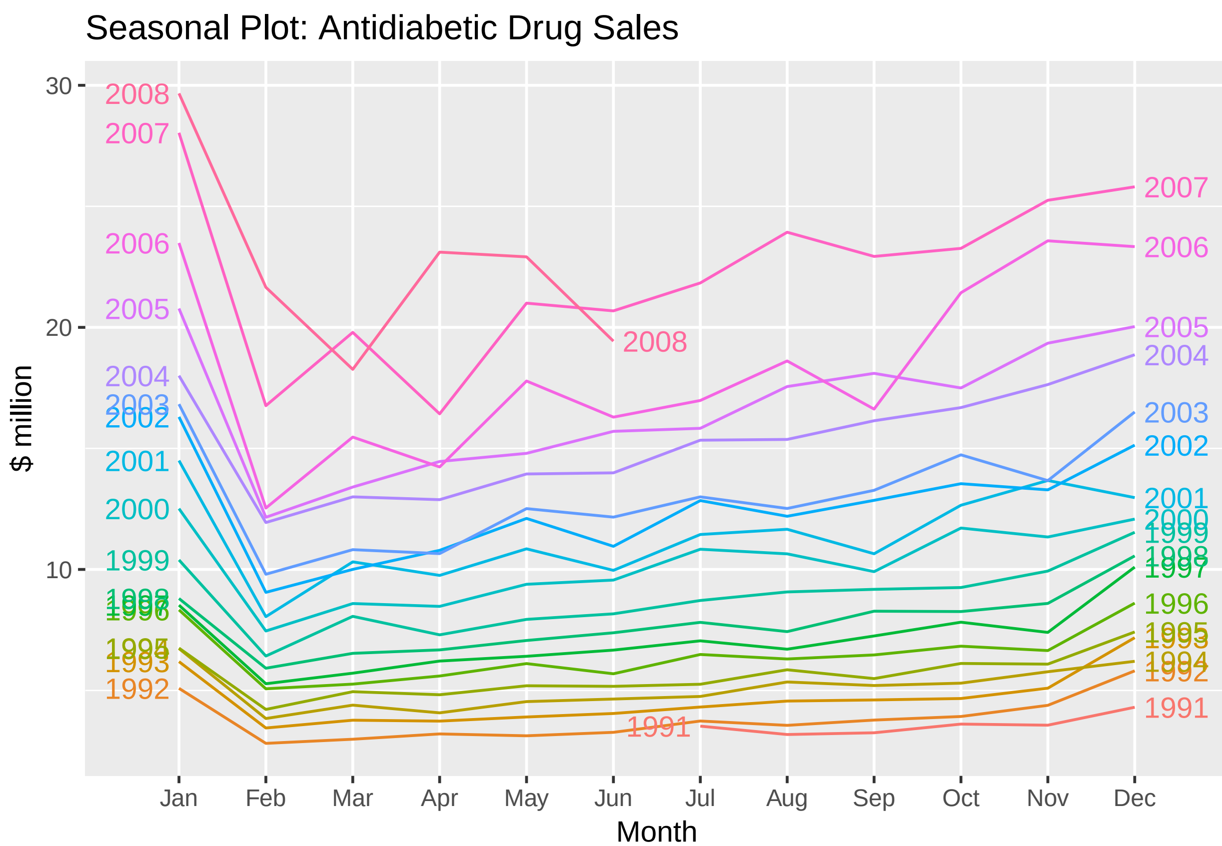
A *seasonal* pattern occurs when a time series is affected by seasonal factors such as the time of the year or day of the week. Seasonality is always of a fixed and known frequency.

Cyclic

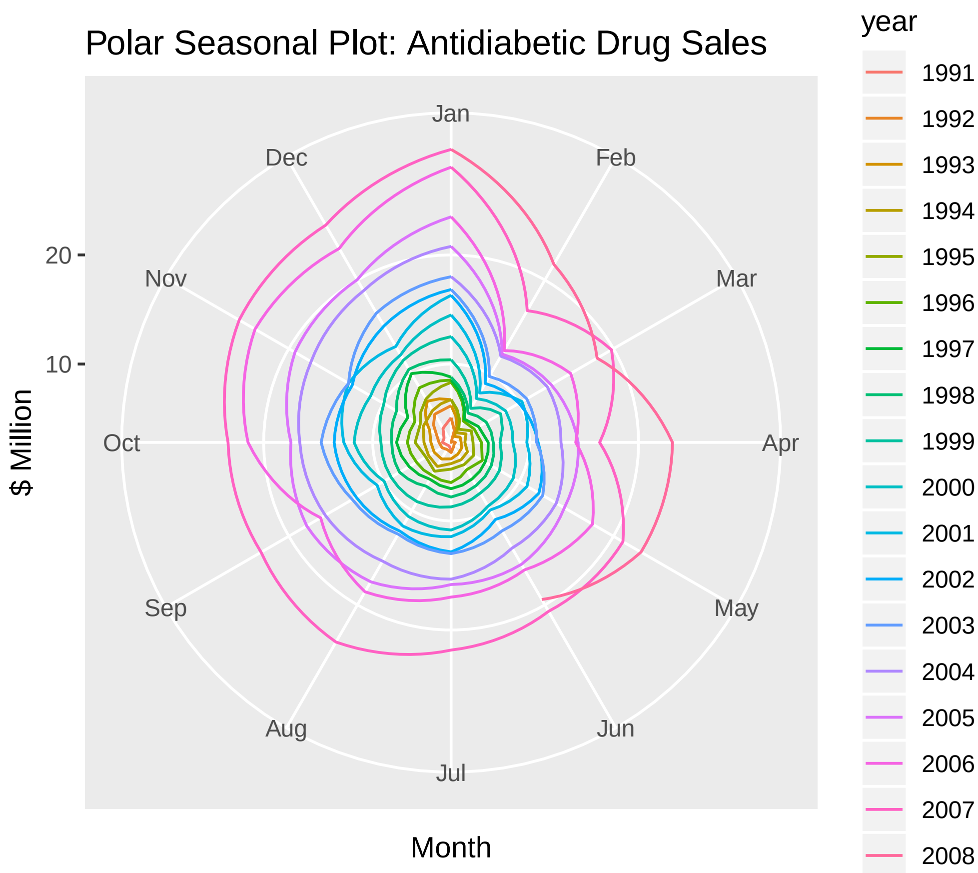
A *cycle* occurs when the data exhibit rises and falls that are not of a fixed frequency. These fluctuations are usually due to economic conditions, and are often related to the business cycle.

2.4 | Seasonal Plots

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



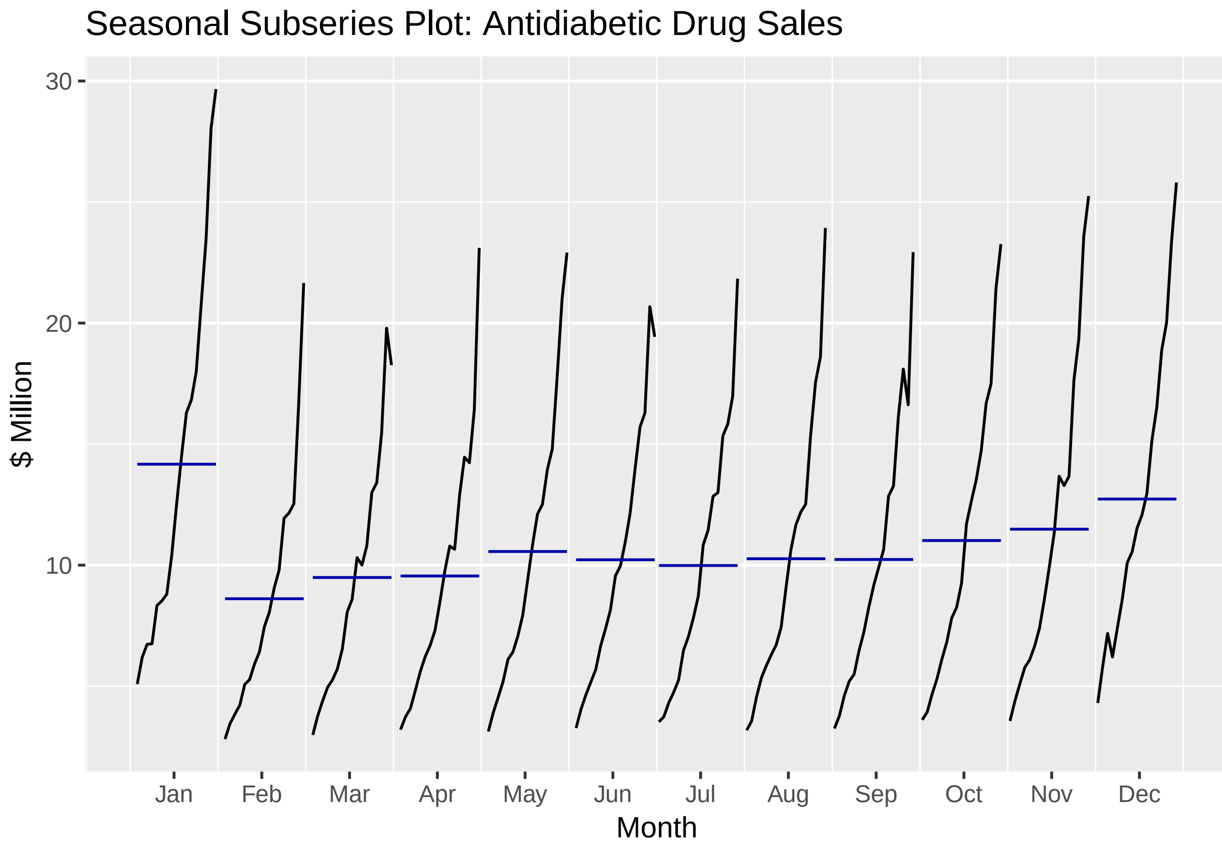

```r
ggseasonplot(a10, polar=TRUE) +
  ylab("$ Million") +
  ggtitle("Polar Seasonal Plot: Antidiabetic Drug Sales")
```



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

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



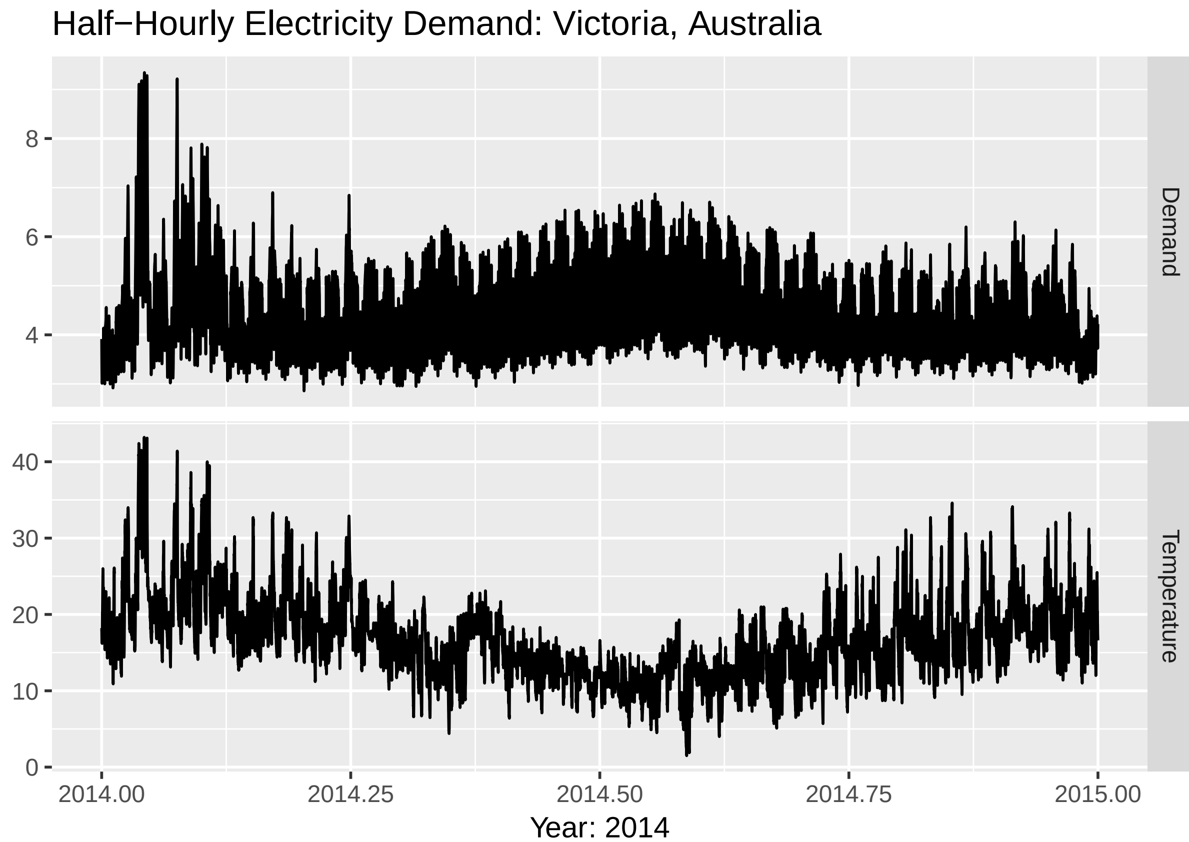
The horizontal lines indicate the means for each month. This plot allows the underlying seasonal pattern to be seen clearly, and shows the change in seasonality over time.

2.6 | Scatterplots

The plots below show two time serieS:

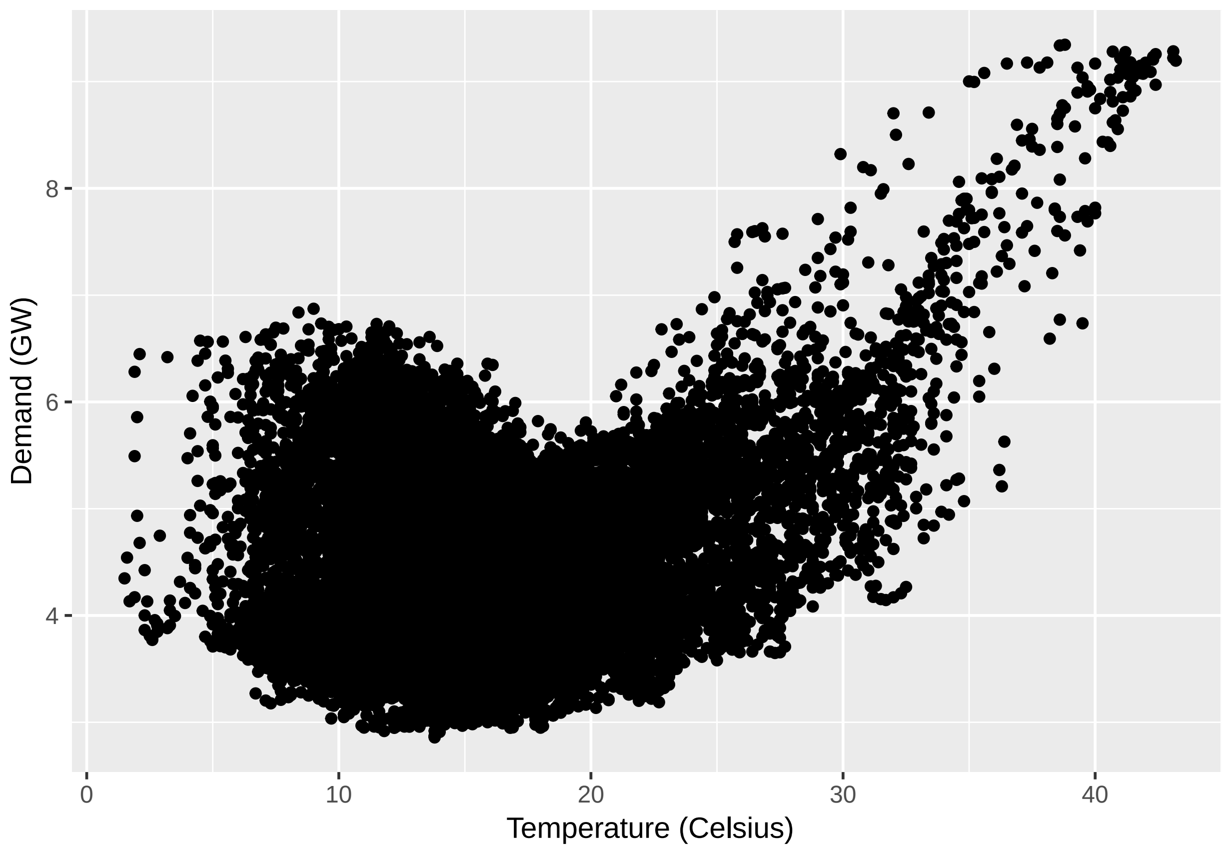
The half hourly electricity demand (in gigawatts) and temperature(in degrees celsius) for 2014 in victoria, australia.

```
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

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

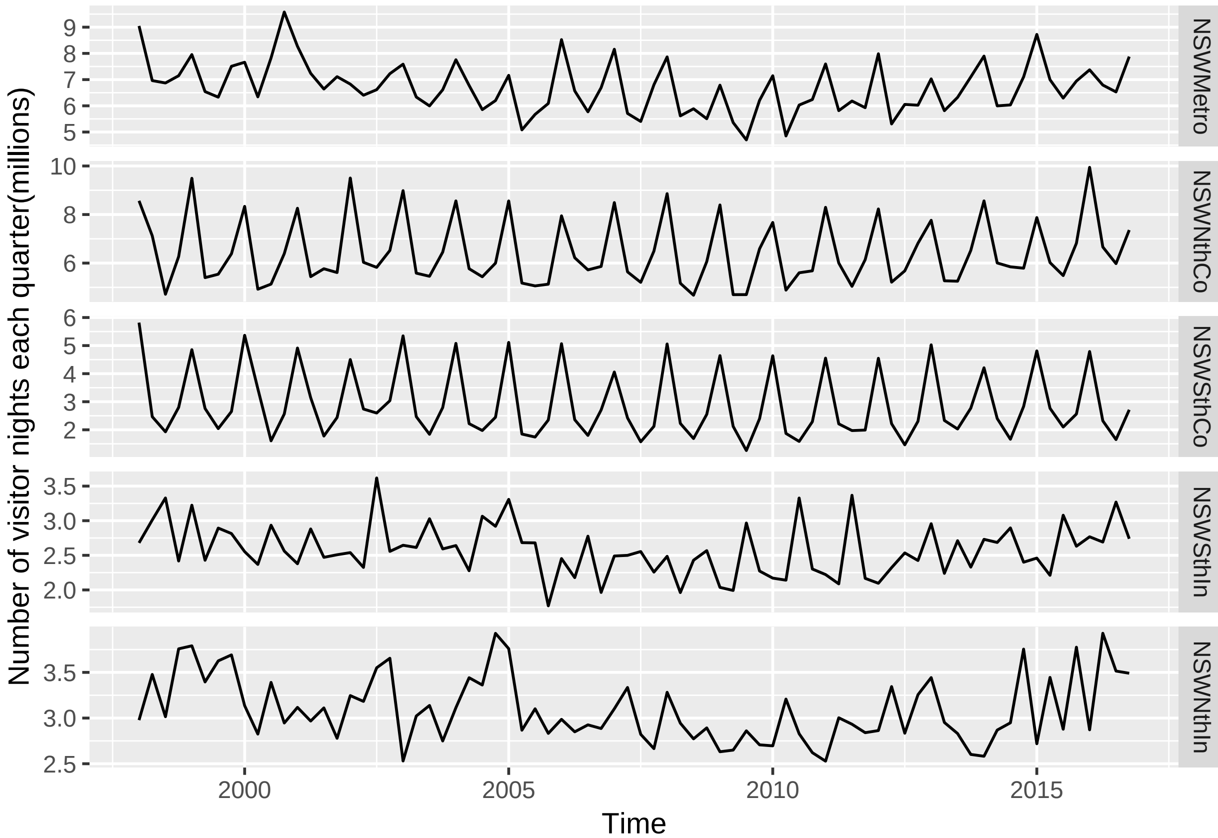
The correlation coefficient $r \in [-1, 1]$ with negative values representing a negative relationship and positive values showing a positive relationship. An important thing to note is that r only measures the strength of the *linear* coefficient.

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

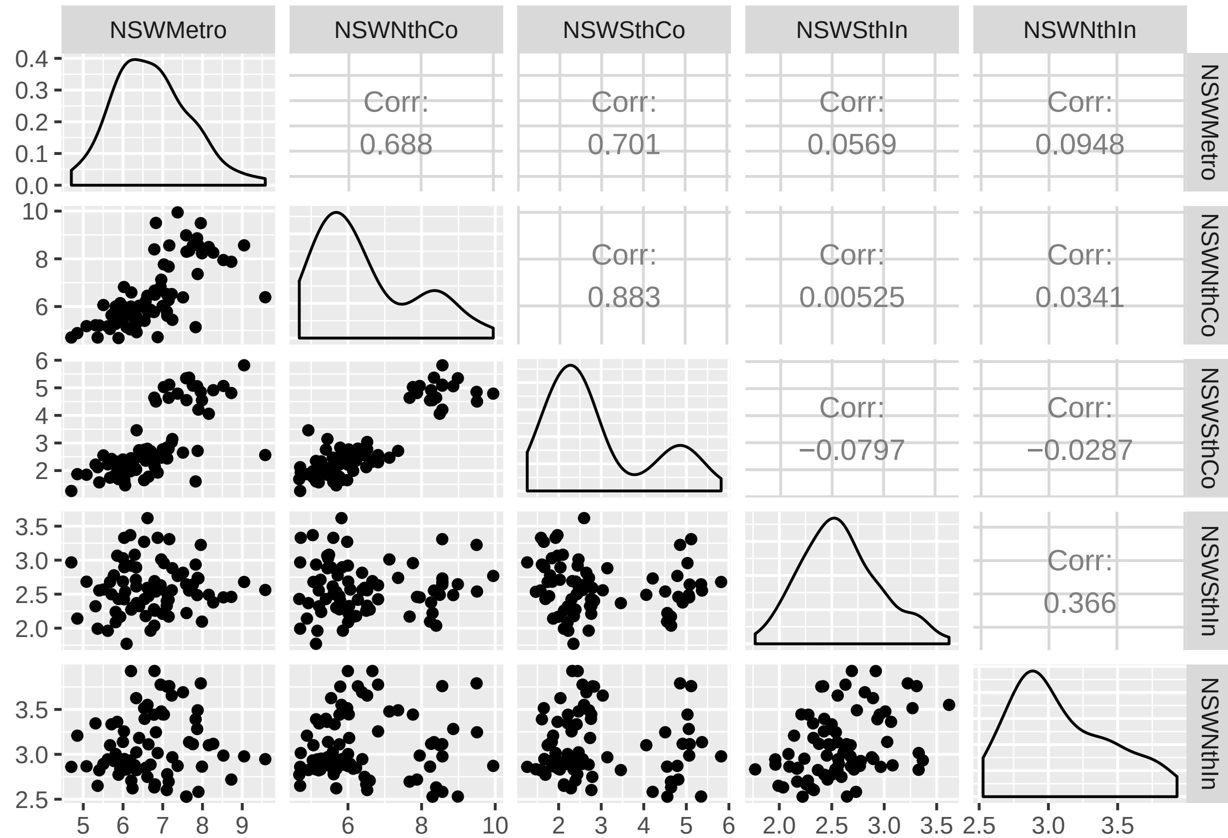
```
autoplot(visnights[, 1:5], facets = TRUE) +
  ylab("Number of visitor nights each quarter(millions)")
```



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