Week 1: Introduction to Time Series Analytics

*Forecasting: Principles and Practice (3rd ed), Rob J Hyndman and George Athanasopoulos, Monash University, Australia:* [*https://otexts.com/fpp3/*](https://otexts.com/fpp3/)

1. Chapter 1+ – Approaching Forecasting

## Introduction to Time Series Analytics

Many statistical methods relate to data which are independent, or at least uncorrelated. There are many practical situations where data might be correlated. This is particularly so where repeated observations on a given system are made sequentially in time.

Data gathered sequentially in time are called a time series.

The simplest form of data is a long series of continuous measurements at equally spaced time points.

That is:

* observations are made at distinct points in time, these time points being equally spaced
* and, the observations may take values from a continuous distribution.

The above setup could be easily generalized: for example, the times of observation need not be equally spaced in time, the observations may only take values from a discrete distribution, . . .

If we repeatedly observe a given system at regular time intervals, it is very likely that the observations we make will be correlated. So, we cannot assume that the data constitute a random sample. The time-order in which the observations are made is vital.

The analysis of experimental data that have been observed at different points of time leads to new and unique problems in statistical modeling and inference. The obvious correlation introduced by the sampling of adjacent points in time can severely restrict the applicability of many conventional statistical methods traditionally dependent on the assumption that these adjacent observations are *independent and identically distributed* (iid). The systematic approach by which one goes about answering the mathematical and statistical questions posed by these time correlations is commonly referred to as time series analytics.

## Objectives of time series analytics:

* description - summary statistics, graphs
* analysis and interpretation - find a model to describe the time dependence in the data, can we interpret the model?
* forecasting or prediction - given a sample from the series, forecast the next value, or the next few values
* control - adjust various control parameters to make the series fit closer to a target
* adjustment - in a linear model, the errors could form a time series of correlated observations, and we might want to adjust estimated variances to allow for this to be minimize

## Definitions and Notations

Assume that the series runs throughout time, that is *t = 0, ±1, ±2,...,* but is only observed at times *t = 1,...,n*.

We denote the forecast value at the time period t as and the error or residual at time period t as *et*where it is equal to . The following table summarizes the forecasting notations:

|  |  |
| --- | --- |
| Notation | Explanation |
| t=1,2,3, ... | An index denoting the time period of interest.*t = 1* is the first period in a series |
|  | A series of n values measured over n time periods, where denotes the value of the series at time period *t* |
|  | The forecasted value for time period *t* |
|  | The k-step-ahead forecast when the forecasting time is*t*. If we are currently at time period *t* , the forecast for the next time period (*t+1*) is denoted Ft+1 |
|  | The forecast error for time period t, which is the difference between the actual value and the forecast at time t, and equal to |

And the following table explains these notations meaning in an example:

|  |
| --- |
| Notation and Meaning with example |
| t = 1,2,3,… denotes day 1, day 2, and day 3, etc. |
| denote the temperatures on days 1,2, and 3 |
| *F4* denotes the forecasted average daily temperature on day 4 |
| *e4* denotes the forecast error for *F4* |

# 

## Initial Step

The first step in any time series investigations always involves careful examination of the recorded data plotted over time. This inspection often suggests the method of analysis as well as statistics that will be of use in summarizing the information in the data.

Before looking more closely at the particular statistical methods, it is appropriate to indicate that two separate, but not mutually exclusive, approaches to time series analysis exist,

* Time domain
* Frequency domain

The time domain approach views the investigation of lagged relationships as most important (that means, how does “what happened today” affect “what will happen tomorrow”).

The frequency domain approach views the investigation of cycles as most important (that means, what is the economic cycle through period of expansion and recession)

## Time Series Data – Statistical Understanding

Some of the problems and questions of importance to the future time series analyst can be exposed by considering real experimental data taken from different subject areas. In this section, I provided a set of time series datasets from various subject areas. You can create their visualizations and identify some of the statistical questions that might be asked about such datasets. All these datasets are available to use from the R “astsa” library. You just need to install the *astas* package and activate the library. The visualizations of these datasets are available in the following sections.

Sample Time Series Datasets

1. Johnson & Johnson Quarterly Earnings dataset: “jj”
2. Global warming dataset: “globaltemp”
3. Speech dataset: “speech”
4. Dow Johns industrial average return. The dataset “djia” is in “xts” library and has data on opening, Highest, lowest, and closing and volume. The following is a sample of that dataset.

Open High Low Close Volume

2006-04-20 11278.53 11384.11 11275.05 11342.89 336420000

2006-04-21 11343.45 11405.88 11316.79 11347.45 325090000

2006-04-24 11346.81 11359.70 11305.83 11336.32 232000000

2006-04-25 11336.56 11355.37 11260.84 11283.25 289230000

2006-04-26 11283.25 11379.87 11282.77 11354.49 270270000

To compute the *return,* we can use the following equation:

*Taking logarithm of the both sides but*

*In this formula,*

1. El Nino and fish population

The monthly values of an environment series called the Southern Oscillation index is in the “soi” dataset and associated recruitment (number of new fish) is in “rec” dataset. Will be interesting to plot these two datasets together and see the impact of the first over the second.

1. A fundamental problem in classical statistics occurs when we are given a collection of independent series or vectors of series, generated under varying experimental condition or treatment configurations. As an example, consider data collected from various locations in the brain via fundamental magnetic resonance imaging (dataset: fMRI). In this example, five subjects were given periodic brushing on the hand. The stimulus was applied for 32 seconds and then stopped for 32 seconds; therefore, the signal period is 64 seconds. The sampling rate was as one observation every 2 seconds for 256 seconds. For this example, the results are averaged over subjects. Generate the visualizations for dataset “fmri1”. One visualization for Cortex data which is stored in columns 2 thru 5 and one observation for Thalamus & Cerebellum data which is stored in columns 6 thru 9. Display these visualizations in one frame.

## Time Series Statistical Models

The primary objective of time series analysis is to develop mathematical models that provide credible descriptions for sample data. We assume a time series can be defined as a collection of random variables (a variable’s values in this context) indexed according to the order they are obtained in time.

The fundamental visual characteristics distinguishing the different series is their differing degrees of smoothness. One possible explanation for this smoothness is that it is being induced by the supposition that adjacent points in time are correlated! So, the value of the series at time t, say, *xt*, depends in some way on the past values of *xt-1*,

*Xt-2, ….*

It is conventional to display a sample time series graphically by linearly plotting the values of the random variables on the vertical axis with the time on the horizontal axis. This method gives hypothetical reconstruction of the time series in continuous format.

White Noise σ

A simple kind of generated series might be a collection of uncorrelated random variables (values in this context), with means 0 and finite variance . The time series generated from uncorrelated variables is used as a model for noise in engineering applications, where it is called *white noise*: we shall denote this process as .

(the symbol means similarity)

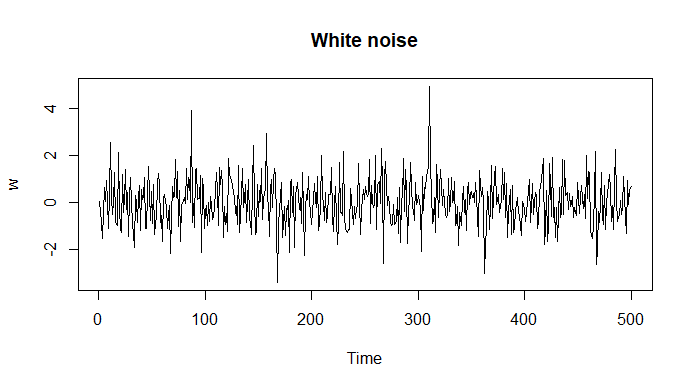
We will sometime require the noise to be *independent* and *identically distributed* *()* random variables with mean equal to 0 and variance of . We distinguish this by writing . A particularly useful white noise series is Gaussian white noise where the are *independent normal random variables* The following figure (fig. 1) shows 500 random numbers which have the characteristics of white noise with close to mean = 0 and . These values are generated by R rnorm().

Fig. 1: Plot of 500 random Variables with

If the stochastic (*of or relating to a process involving a randomly determined sequence of observations each of which is considered as a sample of one element from a probability distribution*) behavior of all time-series could be explained in terms of the white noise model, classical statistical methods would be enough.

Two ways of introducing serial correlation and more smoothness into series models are given as follow:

1. Moving average and filtering. This method will be one of the first method we will learn in this course for time series forecasting.
2. Autoregressions. This method is a part of more complex time series forecasting and will be covered at the end of the course. I will cover briefly this method here and then it will be covered in detail later.

Autoregression

Suppose we consider the white noise series in the figure 1 as input and we calculate the output using the second order equation below:

We can say this equation, for t = 1, 2, …, 500 represents a regression or prediction of the current value of of a time series as a function of past two values of the series and therefore, the term autoregression is used for this model. For now, we ignore the problem of predicting values of times xt-1 and xt-2 and assume we already have them.

Here how we build this scenario in R

1. We generate 550 random numbers with normal distribution
2. The function filter uses zeros for the initial values. That is . In this case , and , then so on. The observation index t = {1, 2, 3, …, 550}

R codes:

wt<-rnorm(550,0, 1) #Generating 550 random numbers with ) characteristics

x<-filter(wt, filter = c(1, -0.9), method="recursive")[-(1:50)] #Apply the regression function and then filter out the first 50

plot.ts(x, main="Autoregression")# plot the result which is the predicted values for the original series

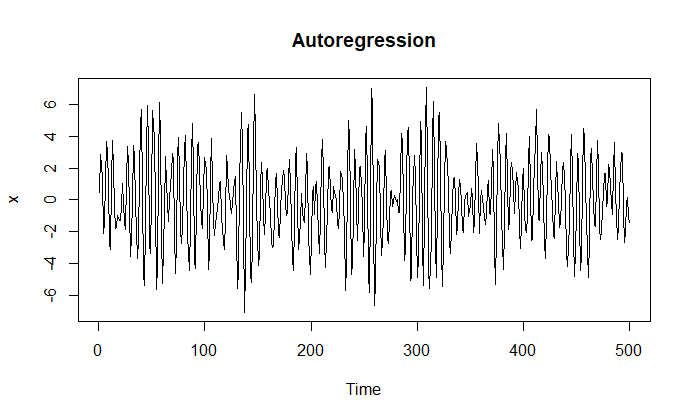


Fig. 2: plot of predicted behavior of the series

If you go back to number 3, in the “Time Series Data” section and the plot of speech series in this document Appendix, you will find similarities between figure 2 and the plot of speech series.

The Figure 3 shows the original white noise and predicted values by autoregression function .

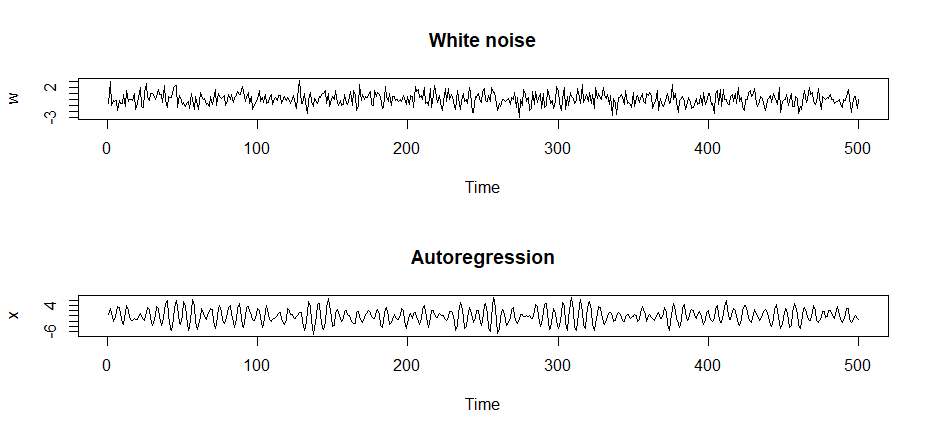


Fig. 3 original white noise and predicted values

Random Walk with Drift

A model for analyzing trend such as the one in global temperature data (Number 2 in the “Time Series Data” section) is the *random walk with drift* model given by , for with initial condition and where is white noise.

The constant is called *drift*, and when the equation above is simply called a *random walk.* The term random walk comes from the fact that, when the value of the time series at the time t is the value of the series at the time t – 1 plus a completely random movement determined by . We may rewrite the *random walk with drift* as a cumulative sum of white noise variables. That is:

**Question:** can you prove the above equation is the same as what we had in the first paragraph?

In the following example, we generate 200 values from the random walk model with and 0.2, and with . The following code generates a graph of random walk with and without drift. Could you identify with and without drift graphs?

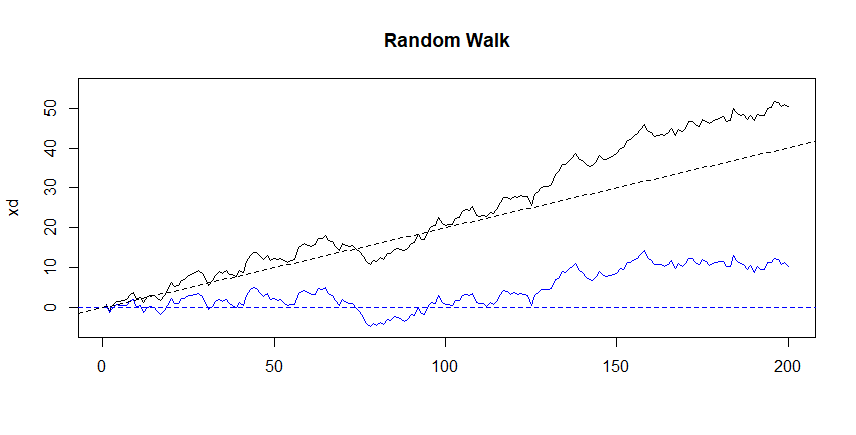


Fig 4. Random walk with and without drift

Here how we built our example in R.

set.seed(154)

w <-rnorm(200); x <- cumsum(w) #two commands in one line 😊

wd = w + 0.2; xd = cumsum(wd)

plot.ts(xd, ylim=c(-5,55), main="Random Walk", xlab='')

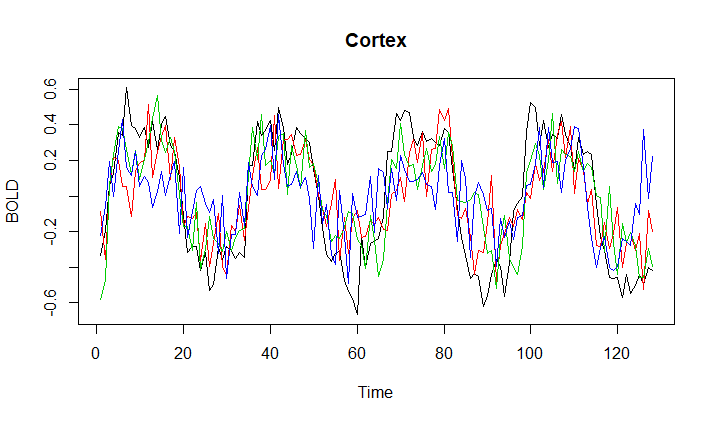
lines(x, col=4); abline(h=0, col=4, lty=2); abline(a=0, b=0.2, lty=2)

Signal in Noise

Many realistic models for generating time series assume an underlying signal with some consistent periodic variation contaminated by adding random noise. For example, it is easy to detect the regular cycle fMARI series (Number 6 in the “Time Series Data” section) displayed on the plot of Cortex data.

Consider the following sinusoidal model:

For the , where the first term {that is } is regarded as signal, in the linear plotting of fMRI Cortex data. I used the following code to plot the Cortex component of the fMRI to clarify this example.



ts.plot(fmri1[,2:5], col=1:4, ylab="BOLD", main="Cortex")

Fig 5. Cortex data

We note that the general sinusoidal waveform can be written as:

**Question**: Find out the amplitude A , oscillation frequency ω, and the phase shift in our example

If we add an additive white noise term with and another with drawn from normal distribution. Adding the two together obscures the signal. The R codes and three graphs in figure 6 shows this example.

w <-rnorm(500, 0, 1)# Generate 500 random values with mean = 0 and

cs<-2\*cos(2\*pi\*1:500/50 + 0.6\*pi)

par(mfrow=c(3, 1))#partition the plot area in three section one plot in every section

plot.ts(cs, main=expression(2\*cos(2\*pi\*t/50 + 0.6\*pi)))

plot.ts(cs+w, main=expression(2\*cos(2\*pi\*t/50 + 0.6\*pi)+N(0,1)))

plot.ts(cs+5\*w, main=expression(2\*cos(2\*pi\*t/50 + 0.6\*pi)+N(0,25)))

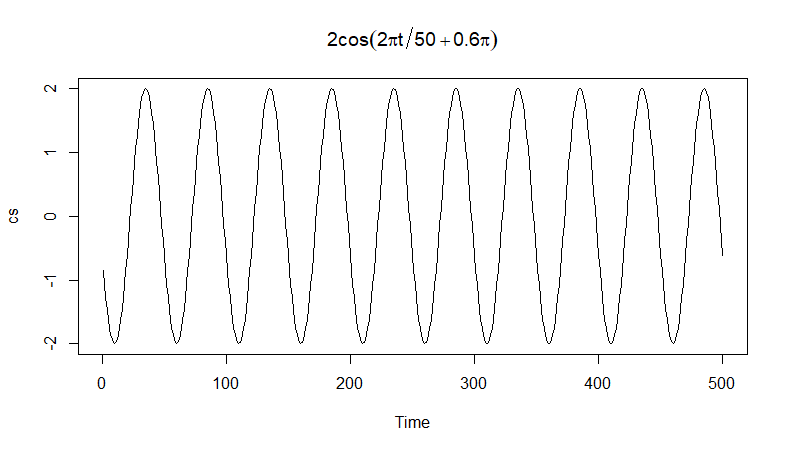
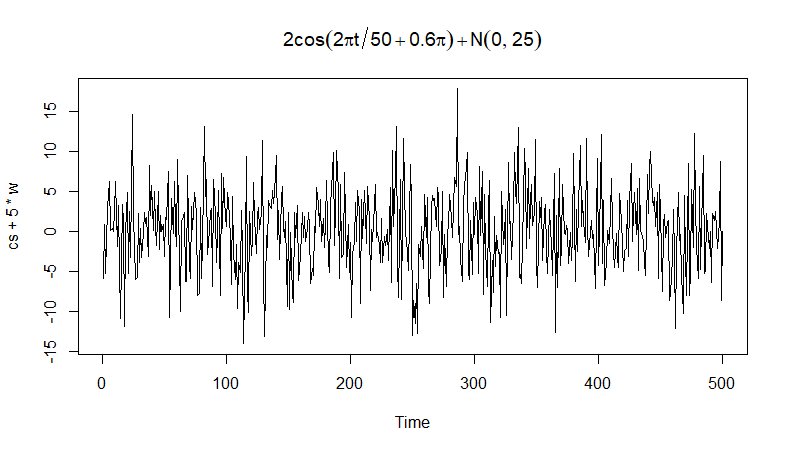
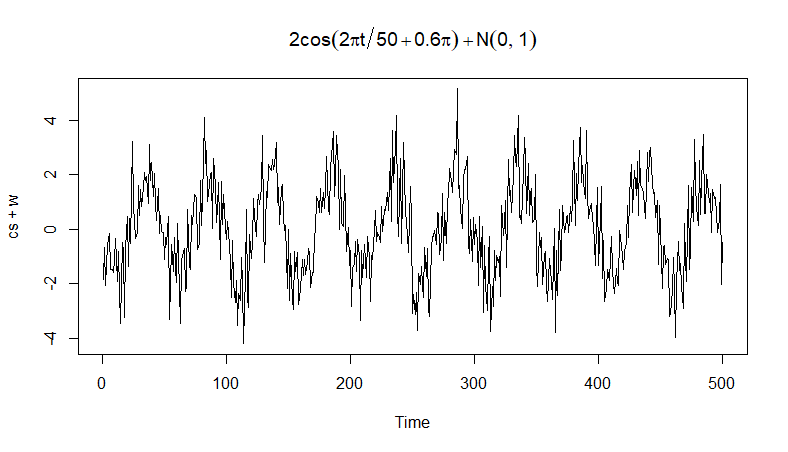


Fig. 6: Signal to Noise Example with different noise level and combined

**Detail figures for this section are in the appendix.**

## Stationary Time Series Data Characteristics

The preceding sections examples have suggested that a sort of regularity may exist over time in the behavior of a time series. This notion of regularity may be introduced using a concept called stationarity.

Strictly Stationary

**Strictly stationary** time series is one for which the probabilistic behavior of every collection of values is identical to that of the time shifted set . That is:

*For all k = 1, 2, … all time points and all time shift*

The version of stationary in the above definition is too strong for most applications. Moreover, it is difficult to assess strict stationary from a single data set. Rather than imposing conditions on all possible distributions of a time series, we will use a milder version of that imposes conditions on the first two moments of a time series. Now we have the following definition:

A **weakly stationary** time series, , is a finite variance process such that

1. The mean value function , is constant and does not depends on time t, and
2. Autocovariance function depends on s and t only through their differences |s-t|.

Therefore, in this course we will use the term stationary to mean weakly stationary. The more in-depth discussion on strictly stationary is beyond the scope of this course.

Stationary time series, a simpler view

A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times. On the other hand, a white noise series is stationary — it does not matter when you observe it, it should look much the same at any point in time.

Some cases can be confusing — a time series with cyclic behavior (but with no trend or seasonality) is stationary. This is because the cycles are not of a fixed length, so before we observe the time series we cannot be sure where the peaks and troughs (or trench) of the cycles will be.

In general, a stationary time series will have no predictable patterns in the long-term. Time plots will show the series to be roughly horizontal (although some cyclic behavior is possible), with constant variance.

The nine line-graphs in figure 7 show some stationary and non-stationary time series. Try to identify them before reading my solutions. The following is the time series graphs explanation:

1. Google stock price for 200 consecutive days;
2. Daily change in the Google stock price for 200 consecutive days;
3. Annual number of strikes in the US;
4. Monthly sales of new one-family houses sold in the US;
5. Annual price of a dozen eggs in the US (constant dollars);
6. Monthly total of pigs slaughtered in Victoria, Australia;
7. Annual total of lynx trapped in the McKenzie River district of north-west Canada;
8. Monthly Australian beer production;
9. Monthly Australian electricity production.

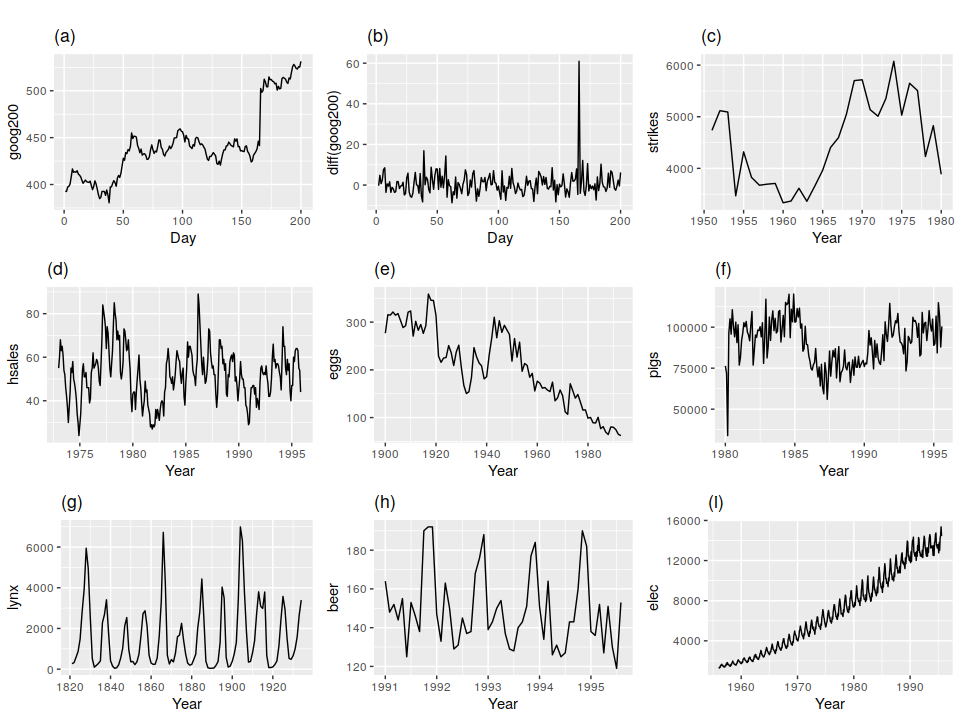


Fig. 7: Stationary and non-stationary time series. Which of these series are stationary?

Obvious seasonality rules out series (d), (h) and (i). Trends and changing levels rule out series (a), (c), (e), (f) and (i). Increasing variance also rules out (i). That leaves only (b) and (g) as stationary series.

At first glance, the strong cycles in series (g) might appear to make it non- stationary. But these cycles are aperiodic — they are caused when the lynx cat population becomes too large for the available feed, so that they stop breeding and the population falls to low numbers, then the regeneration of their food sources allows the population to grow again, and so on. In the long-term, the timing of these cycles is not predictable. Hence the series is stationary.

Note: White Noise is stationary and random walk is not. Proof is beyond the scope of this course.

1. Chapter 2 – Time series Data Exploration

## Basic Information

1. Data Collection
2. Data quality
   1. Measurement Accuracy
   2. Missing Values
   3. Corrupted Data
   4. Data Entry Errors
3. Temporal Frequency
4. Series Granularity
5. Domain Expertise
6. Time Series Components
   1. Level
   2. Trend
   3. Seasonality
   4. Noise

## The seasonal period

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.

Some common periods for different time intervals are shown in the table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data** | **Minute** | **Hour** | **Day** | **Week** | **Year** |
| Quarters |  |  |  |  | 4 |
| Months |  |  |  |  | 12 |
| Weeks |  |  |  |  | 52 |
| Days |  |  |  | 7 | 365.25 |
| Hours |  |  | 24 | 168 | 8766 |
| Minutes |  | 60 | 1440 | 10080 | 525960 |
| Seconds | 60 | 3600 | 86400 | 604800 | 31557600 |

If we are dealing with data which we do not know the seasonality period but want to create its timeseries, we just use 1 as the frequency or we ignore this parameter.

These components together, at the time t make .

If these components in the additive nature, then:

and with multiplicative nature:

1. Visualizing Time Series
2. Data Pre-Processing
   1. Solution with Missing value issue
   2. Unequally Spaced Series
   3. Extreme Values
   4. Choice of Time Span

# R Code for creating the time plot of Amtrak ridership (Textbook: fig 2.2)

Amtrak.data <- read.csv("Amtrak data.csv")

ridership.ts <- ts(Amtrak.data$Ridership, start = c(1991,1), end = c(2004, 3), freq = 12)

plot(ridership.ts, xlab = "Time", ylab = "Ridership", ylim = c(1300, 2300), bty = "l")

## Seasonal plots

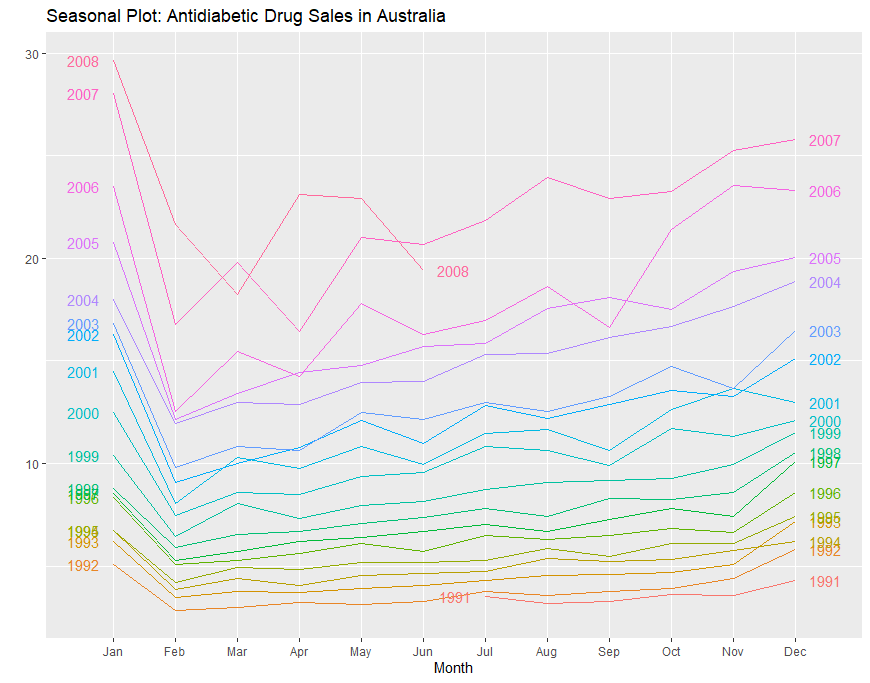
A seasonal plot is similar to a time plot except that the data are plotted against the individual “seasons” in which the data were observed. An example is given below (Figure 8) showing the antidiabetic drug sales.

The following R codes and packages are used to create the most plots in this section

*install.packages("fpp"), install.packages("fpp2")*

*install.packages('forecast', dependencies = TRUE)#to get seasonal plot functions*

*library(fpp), library(fpp2) #to get datasets, library(ggplot2), library(forecast), data(a10) #get the antidiabetic drug sales data*

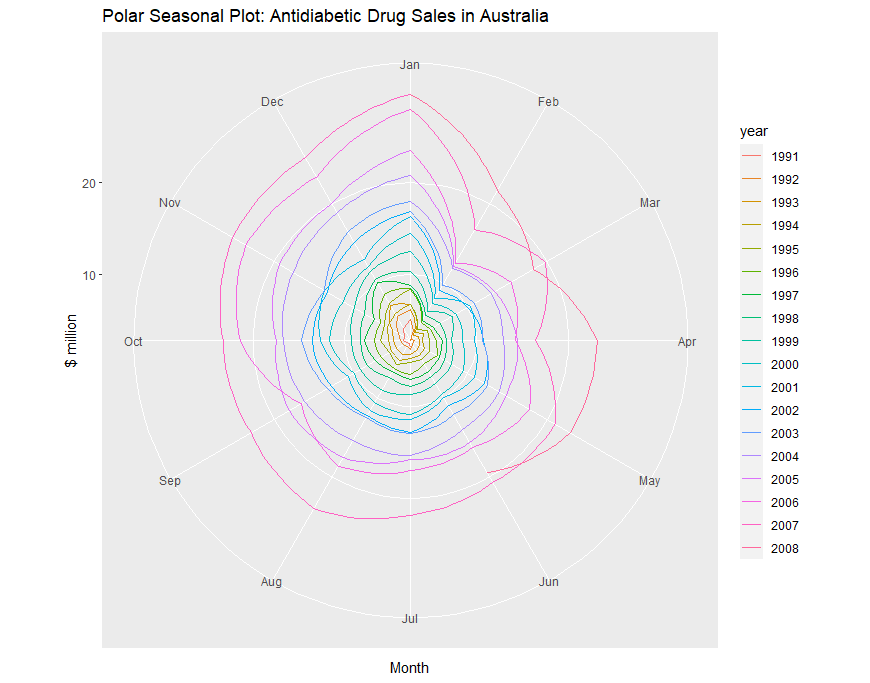
Figure 8:

*ggseasonplot(a10, year.labels = TRUE, year.labels.left = TRUE) + labs(ylab = "$million") +*

*ggtitle("Seasonal Plot: Antidiabetic Drug Sales in Australia")*

It is clear that there is a large jump in sales in January each year. Actually, these are probably sales in late December as customers stockpile before the end of the calendar year, but the sales are not registered with the government until a week or two later. The graph also shows that there was an unusually small number of sales in March 2008 (most other years show an increase between February and March). The small number of sales in June 2008 is probably due to incomplete counting of sales at the time the data were collected.

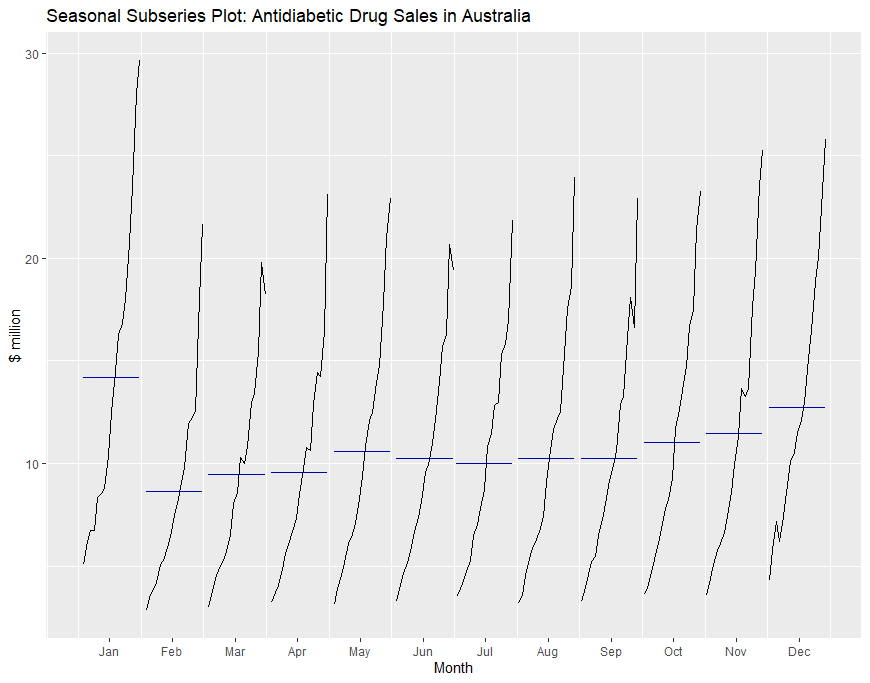
A useful variation on the seasonal plot uses polar coordinates. Setting polar=TRUE makes the time series axis circular rather than horizontal, as shown below (Figure 9).



*ggseasonplot(a10, polar = TRUE) + ylab("$ million") + ggtitle("Polar Seasonal Plot: Antidiabetic Drug Sales in Australia")*

Seasonal subseries plots

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

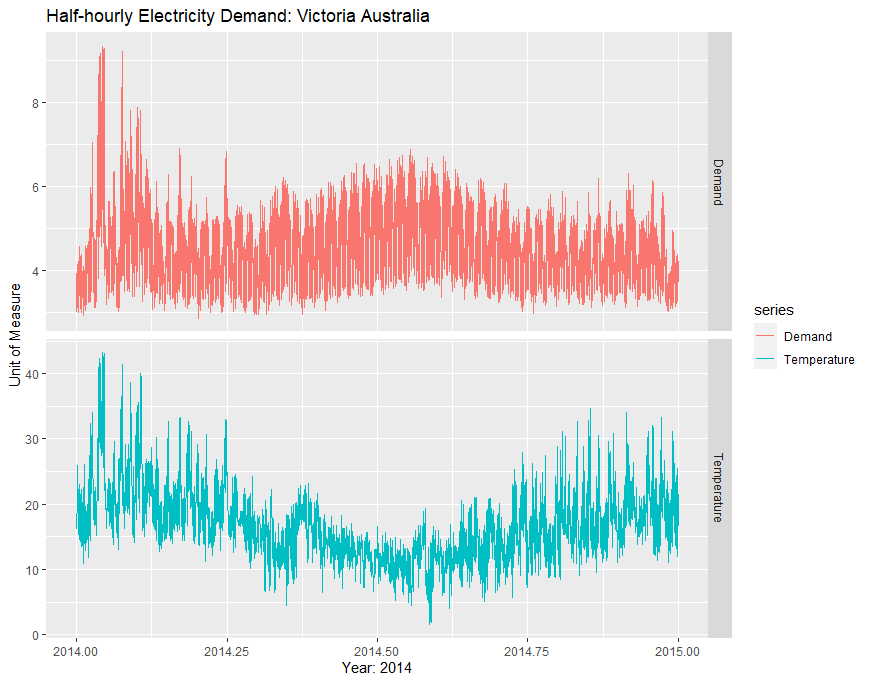


ggsubseriesplot(a10) + ylab("$ million") + ggtitle("Seasonal Subseries Plot: Antidiabetic Drug Sales in Australia")

The 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.

Scatterplots

The graphs discussed so far are useful for visualizing individual time series. It will be useful to explore relationships between time series.

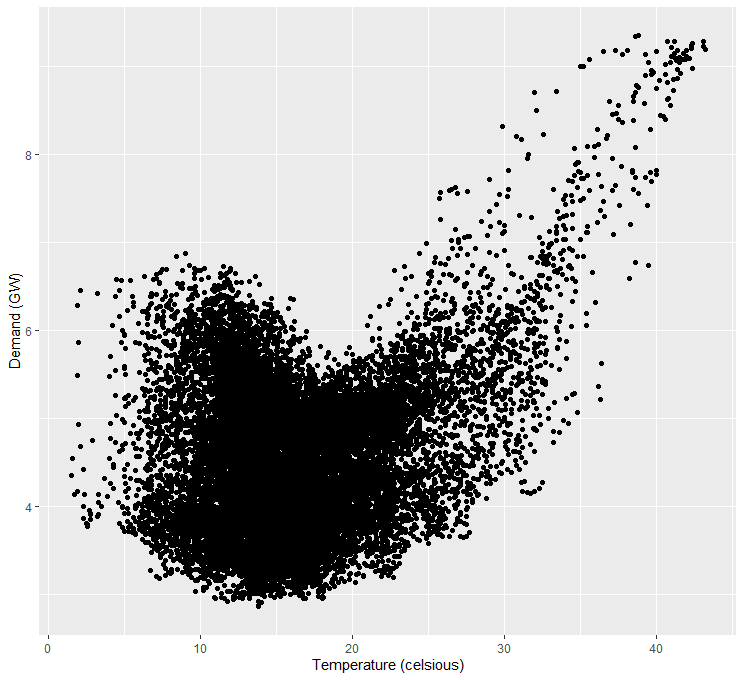
Figure below shows two time series: half-hourly electricity demand (in Gigawatts) and temperature (in degrees Celsius), for 2014 in Victoria, Australia. The temperatures are for Melbourne, the largest city in Victoria, while the demand values are for the entire state.

data(elecdemand) # get the electricity and temperature data

autoplot(elecdemand[,c("Demand", "Temperature")], colour = TRUE, facets= TRUE) + xlab("Year: 2014") + ylab("Unit of Measure") +

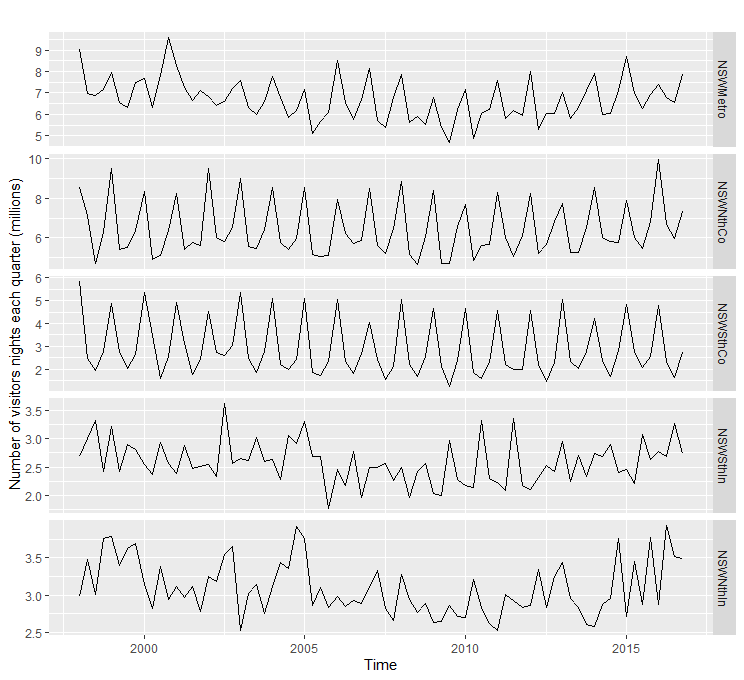
ggtitle("Half-hourly Electricity Demand: Victoria Australia")

Also, we can study the relationship between demand and temperature by plotting one series against the other.



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

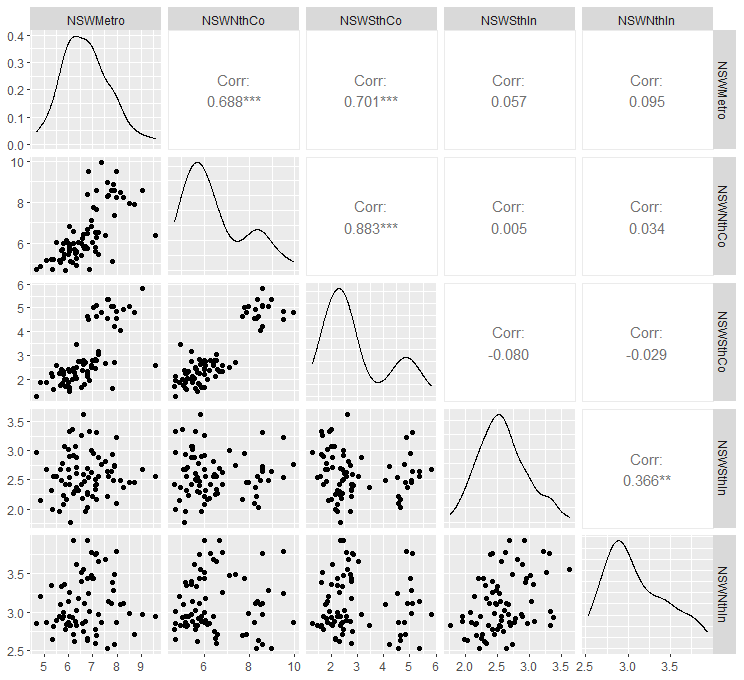
Scatterplot matrices

When there are several potential predictor variables, it is useful to plot each variable against each other variable. Consider the five time series shown in Figure below, showing quarterly visitor numbers for five regions of New South Wales, Australia.

data("visnights")

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

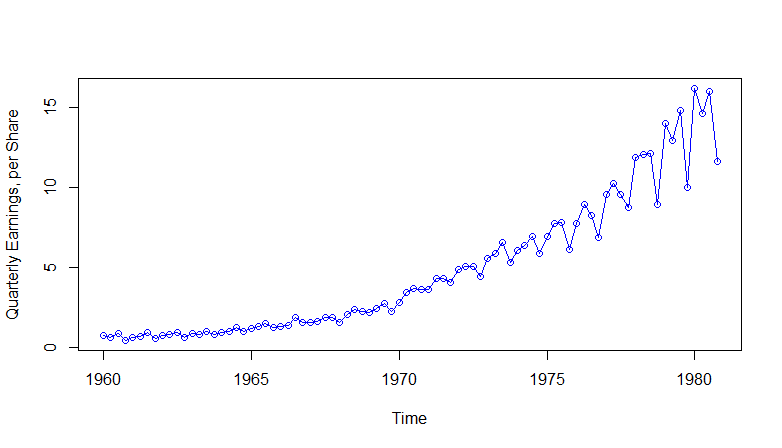
To see the relationships between these five time series, we can plot each time series against the others. These plots can be arranged in a scatterplot matrix, as shown in below. (This plot requires the GGally package to be installed.)



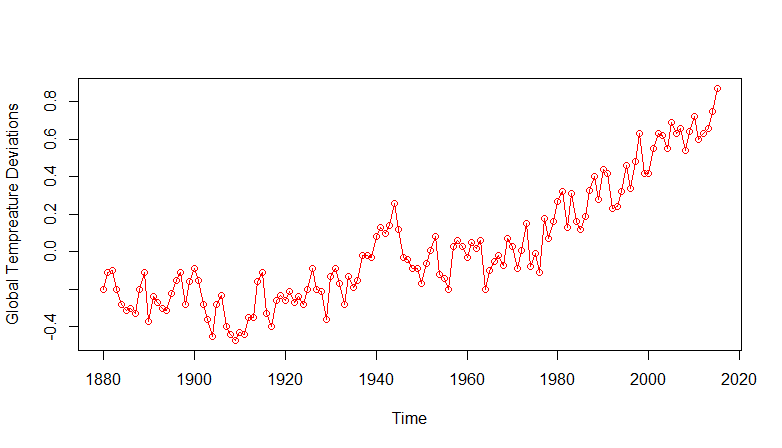
install.packages("GGally")

library(GGally)

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

1. Appendix

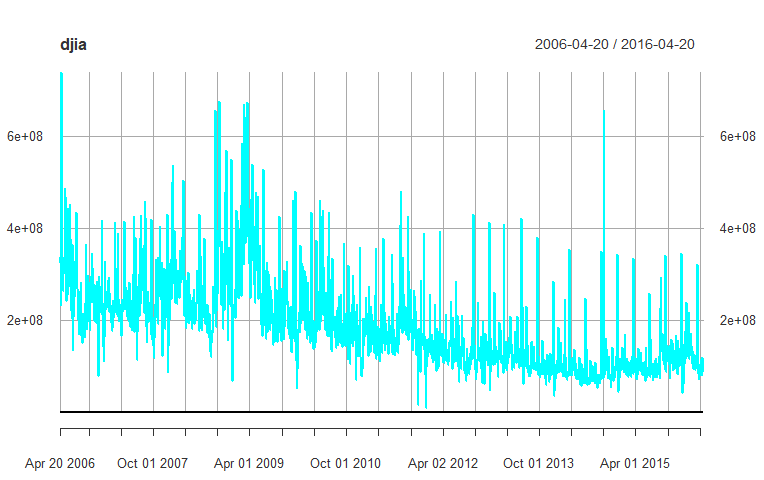
# Johnson & Johnson Quarterly Earnings dataset: “jj”

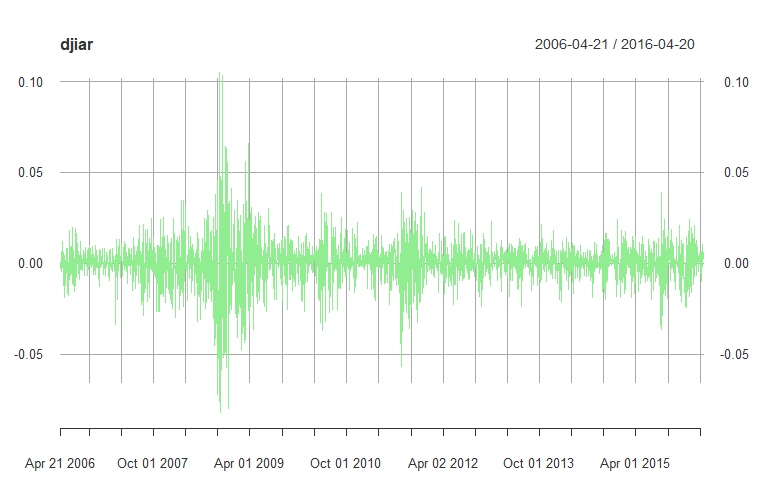


# Global warming dataset: “globaltemp”

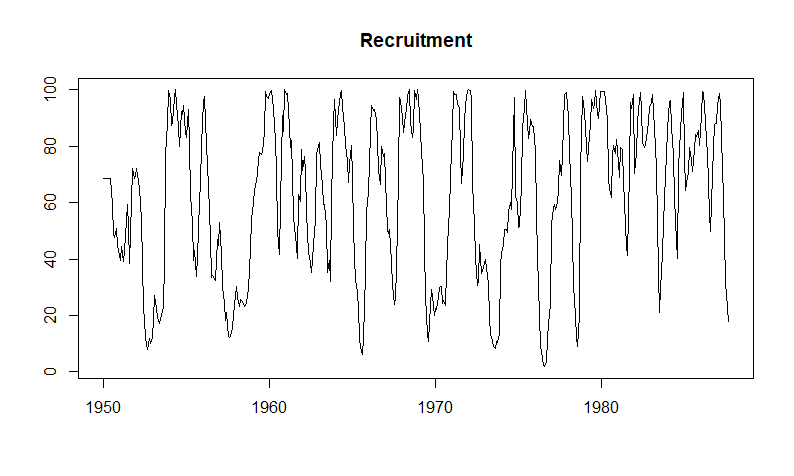
# Speech dataset: “speech”

# Dow Johns industrial average return dataset: “djia” (Dow John at closing and average return)





# El Nino and fish population datasets: “soi” and “rec”.



# Fundamental magnetic resonance imaging dataset: “fMRI”.

