

Introduction

Examples of time series

Annual Rainfall in Los Angeles

An Industrial Chemical Process

Annual Abundance of Canadian Hare

Monthly Average Temperatures in Dubuque, Iowa

Monthly Oil Filter Sales

A Model-Building Strategy

Time Series and Stochastic Processes

Means, Variances, and Covariances

The Random Walk

A Moving Average

Stationarity

White Noise

Random Cosine Wave

Summary

# Module 1 - Basic Plots, Examples, and Fundamental Concepts

## MATH1318 Time Series Analysis

Prepared by Dr Haydar Demirhan based on the textbook by Cryer and Chan, Time Series Analysis with R, Springer.

## Introduction

Time series data is composed of observations collected sequentially over time. This type of data is frequently seen and analysed in data analytics applications. In business, we observe

- weekly interest rates,
- daily closing stock prices,
- monthly price indices,
- yearly sales figures, and so forth.

In meteorology, we observe

- daily high and low temperatures,

- annual precipitation and drought indices, and
- hourly wind speeds.

In agriculture, we record

- annual figures for crop and livestock production,
- soil erosion, and export sales.

In the biological sciences, we observe the electrical activity of the heart at millisecond intervals. In ecology, we record the abundance of an animal species.

As a recent example from time series research, Miwa Fukino, Yoshito Hirata, and Kazuyuki Aihara from University of Tokyo (<https://sinews.siam.org/Details-Page/music-visualized-by-nonlinear-time-series-analysis>) proposed a new method based on nonlinear time series analysis analyse music's emotional effect.

In this module,

- we will start with examples of time series and visualise time series data with the most frequently used plots.
- Introduce the model-building strategy that we will follow through the semester.
- We will focus on the fundamental concepts in terms of
  - Time Series and Stochastic Processes,
  - Means, Variances, and Covariances, and
  - Stationarity.

## Examples of time series

In this section, we will see some examples of time series and basic visualisations to make inferences about the basic characteristics of time series data. In all of the examples, we use the R package `TSA`. Therefore you need to install the `TSA` packages calling

```
install.packages("TSA")
```

## Annual Rainfall in Los Angeles

In this example, the data is composed of the annual rainfall amounts recorded in Los Angeles, California, over more than 100 years. The time series plot for this data is generated with the following code chunk. Here, the dataset `larain` contains the annual rainfall data. Please note that throughout the codes given in this module, you can also use `win.graph()` function to open a new graphics window with the settings determined by the arguments of this function.

```
library(TSA)
```

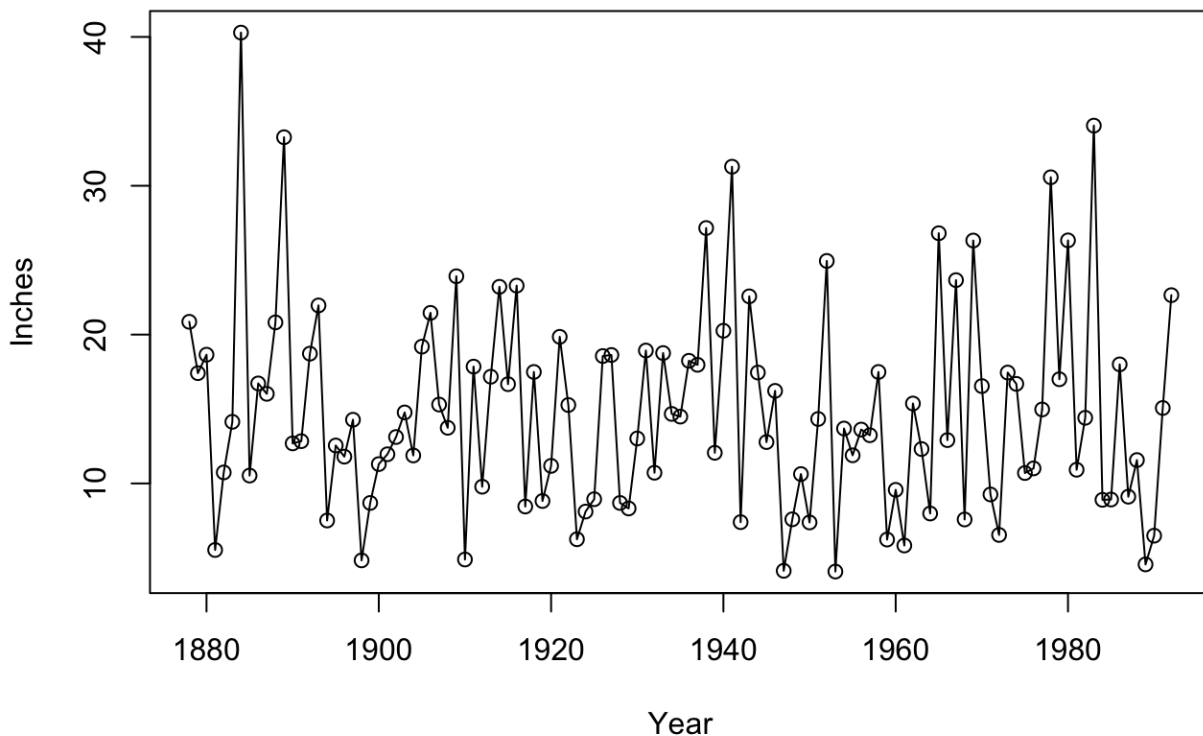
```
##  
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':  
##  
##   acf, arima
```

```
## The following object is masked from 'package:utils':  
##  
##   tar
```

```
# win.graph(width=4.875, height=2.5,pointsize=8)  
data(larain)  
plot(larain,ylab='Inches',xlab='Year',type='o', main = "Figure 1. Time series plot of LA rain series.")
```

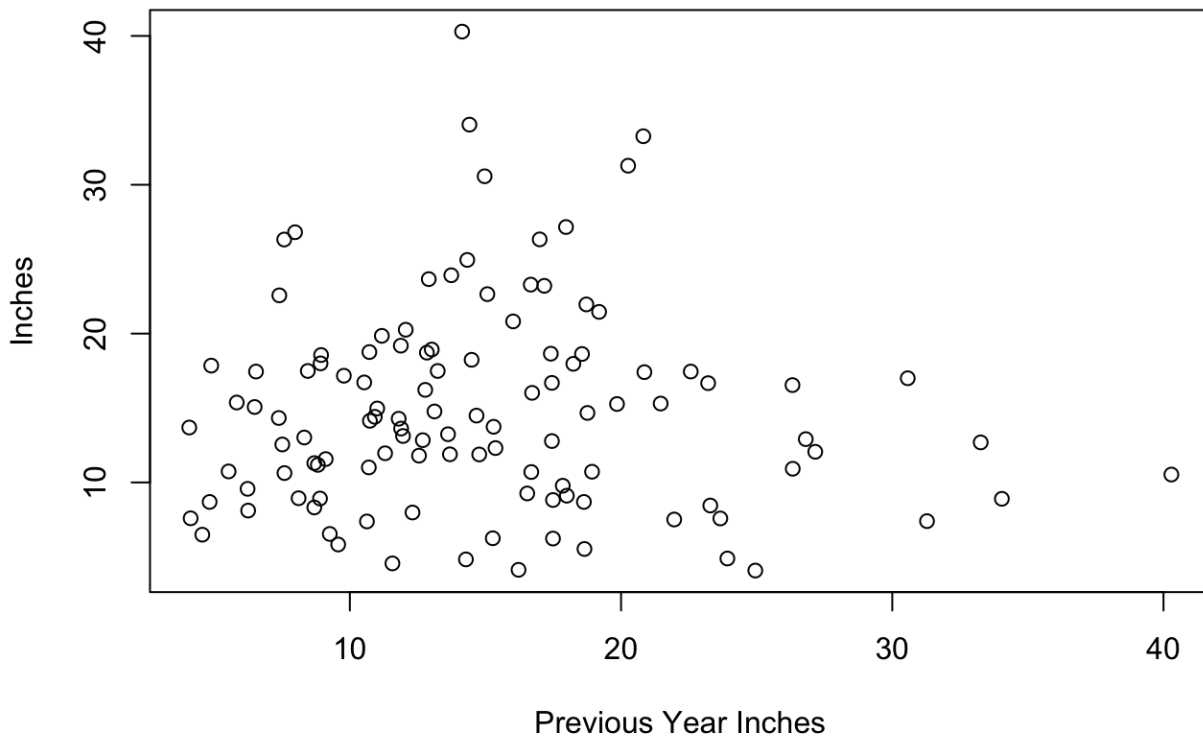
**Figure 1. Time series plot of LA rain series.**



In Figure 1, we observe considerable variation in rainfall amount over the years from this time series plot. The year 1883 was an exceptionally wet year for Los Angeles, while 1983 was quite dry. For the analysis, we are interested in whether or not consecutive years are related in some way. If so, we might be able to use one year's rainfall value to help forecast next year's rainfall amount. The following code chunk generates a scatter plot to investigate the relationship between pairs of consecutive rainfall values:

```
# win.graph(width=3,height=3,pointsize=8)  
plot(y=larain,x=zlag(larain),ylab='Inches', xlab='Previous Year Inches', main = "Figure 2. Scatter plot of rainfall in consecutive years.")
```

**Figure 2. Scatter plot of rainfall in consecutive years.**



The plots show no “trends” and no general tendencies. In Figure 2, the point plotted near the lower right-hand corner shows that the year of extremely high rainfall, 40 inches in 1883, was followed by an amount nearly 12 inches in 1884. The point near the top of the display shows that the 40-inch year was preceded by a much more typical year of about 15 inches. We observe from this plot that there is little information about this year’s rainfall amount from last year’s amount. The correlation between last year’s rainfall amount and this year’s amount which can be calculated by the following code chunk:

```
y = larin          # Read the annual rainfall data into y
x = zlag(larin)     # Generate first lag of the annual rainfall series
index = 2:length(x) # Create an index to get rid of the first NA value i
                  n x
cor(y[index],x[index]) # Calculate correlation between numerical values in
                  x and y
```

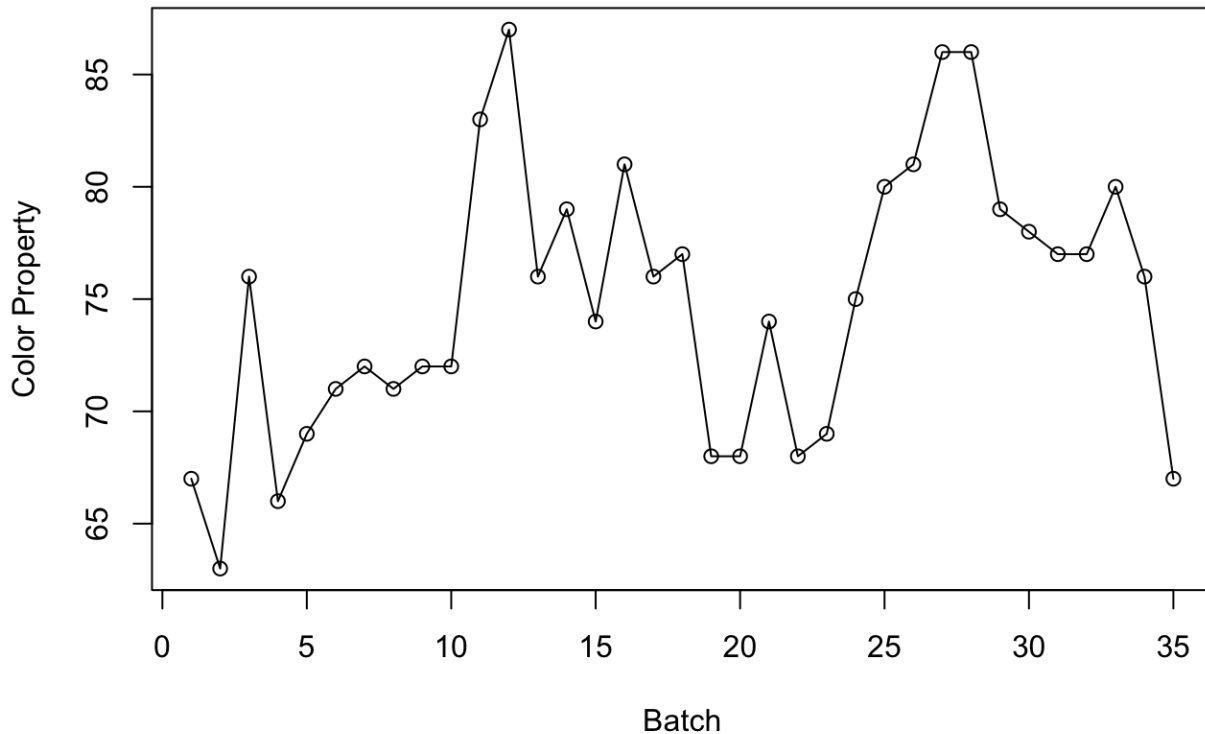
```
## [1] -0.03308892
```

## An Industrial Chemical Process

The second example is on a time series from an industrial chemical process. A color property from consecutive batches in the process is measured here and named `color` in the package. We plot the time series with the following code chunk:

```
# win.graph(width=4.875, height=2.5,pointsize=8)
data(color)
plot(color,ylab='Color Property',xlab='Batch',type='o', main = "Figure 3. Time series plot of the color property series.")
```

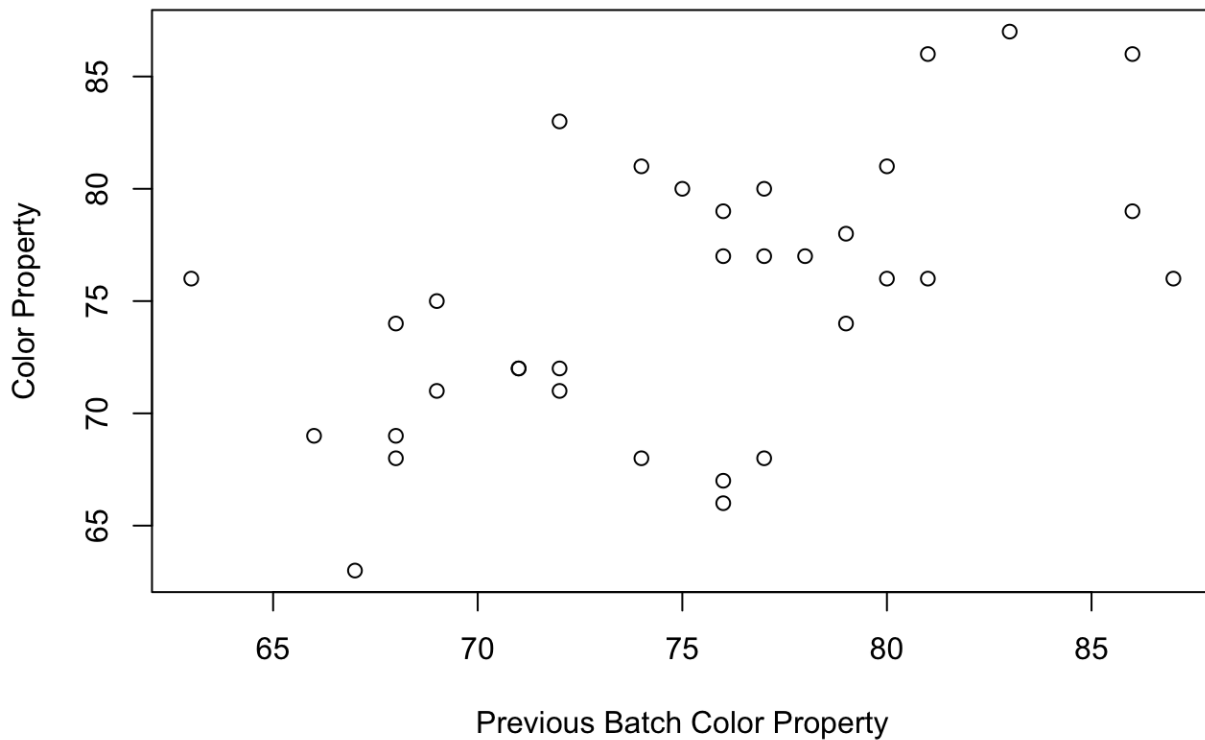
**Figure 3. Time series plot of the color property series.**



From Figure 3, it seems that succeeding measurements are related to one another as the values that are neighbors in time tend to be similar in size. This is more clearly seen in the following scatter plot of neighboring pairs:

```
# win.graph(width=3,height=3,pointsize=8)
plot(y=color,x=zlag(color),ylab='Color Property', xlab='Previous Batch Color Property', main = "Figure 4. Scatter plot of the color property in consecutive batches.")
```

**Figure 4. Scatter plot of the color property in consecutive batches.**



In Figure 4, we see a slight upward trend, which is apparent but not terribly strong. The correlation in this scatter plot is about 0.6:

```
y = color          # Read the color data into y
x = zlag(color)     # Generate first lag of the color series
index = 2:length(x) # Create an index to get rid of the first NA value i
                  n x
cor(y[index],x[index]) # Calculate correlation between numerical values in
                  x and y
```

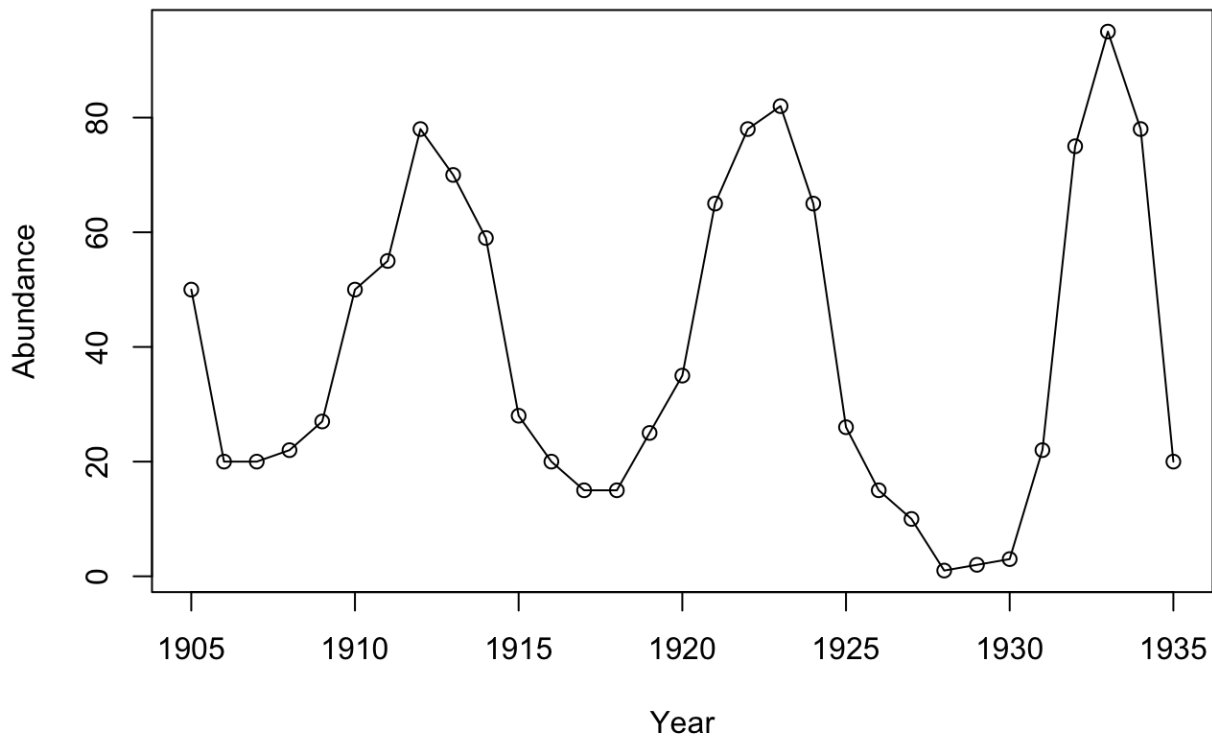
```
## [1] 0.554917
```

## Annual Abundance of Canadian Hare

The third example concerns the annual abundance of Canadian hare. The data consists of their abundance over about 30 years. The following is the time series plot of annual abundance of Canadian hare for 30 years:

```
# win.graph(width=4.875, height=2.5,pointsize=8)
data(hare)
plot(hare,ylab='Abundance',xlab='Year',type='o', main = "Figure 5. Time ser
ies plot of annual abundance of Canadian hare.")
```

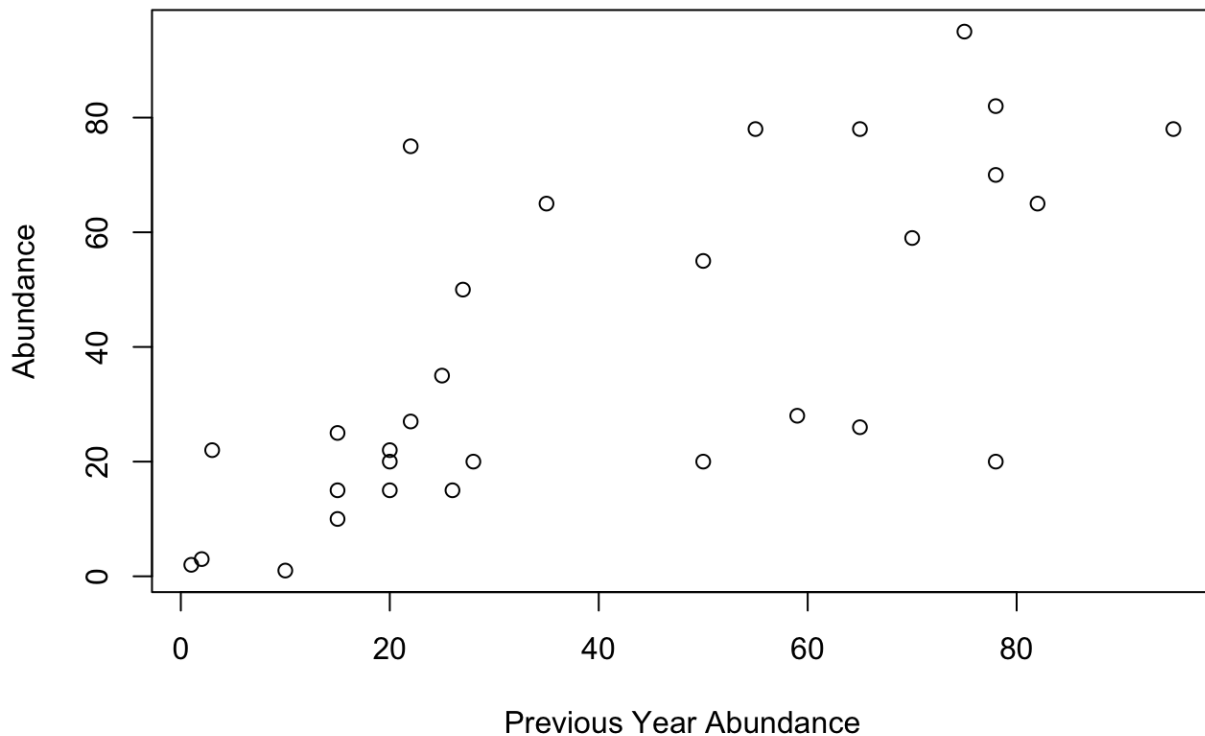
**Figure 5. Time series plot of annual abundance of Canadian hare.**



In Figure 5, neighboring values here are very closely related. Large changes in abundance do not occur from one year to the next. The following plot shows the correlation between abundance versus the previous year's abundance:

```
# win.graph(width=3, height=3,pointsize=8)
plot(y=hare,x=zl原因(hare),ylab='Abundance', xlab='Previous Year Abundance',
      main = "Figure 6. Scatter plot of abundance in consecutive years."
    )
```

**Figure 6. Scatter plot of abundance in consecutive years.**



In Figure 6, we see an upward trend in the plot-low values tend to be followed by low values in the next year, middle-sized values by middle-sized values, and high values by high values. The amount of correlation between neighboring abundance values is 0.70:

```
y = hare                # Read the abundance data into y
x = zlag(hare)           # Generate first lag of the abundance series
index = 2:length(x)      # Create an index to get rid of the first NA value i
                        n x
cor(y[index],x[index])  # Calculate correlation between numerical values in
                        x and y
```

```
## [1] 0.7025777
```

## Monthly Average Temperatures in Dubuque, Iowa

The average monthly temperatures (in degrees Fahrenheit) over a number of years recorded in Dubuque, Iowa, are shown in the following plot:

```
# win.graph(width=4.875, height=2.5,pointsize=8)
data(tempdub)
plot(tempdub,ylab='Temperature',type='o', main = "Figure 7. Time series plo
t of average monthly temperature series.")
```



**Figure 7. Time series plot of average monthly temperature series.**

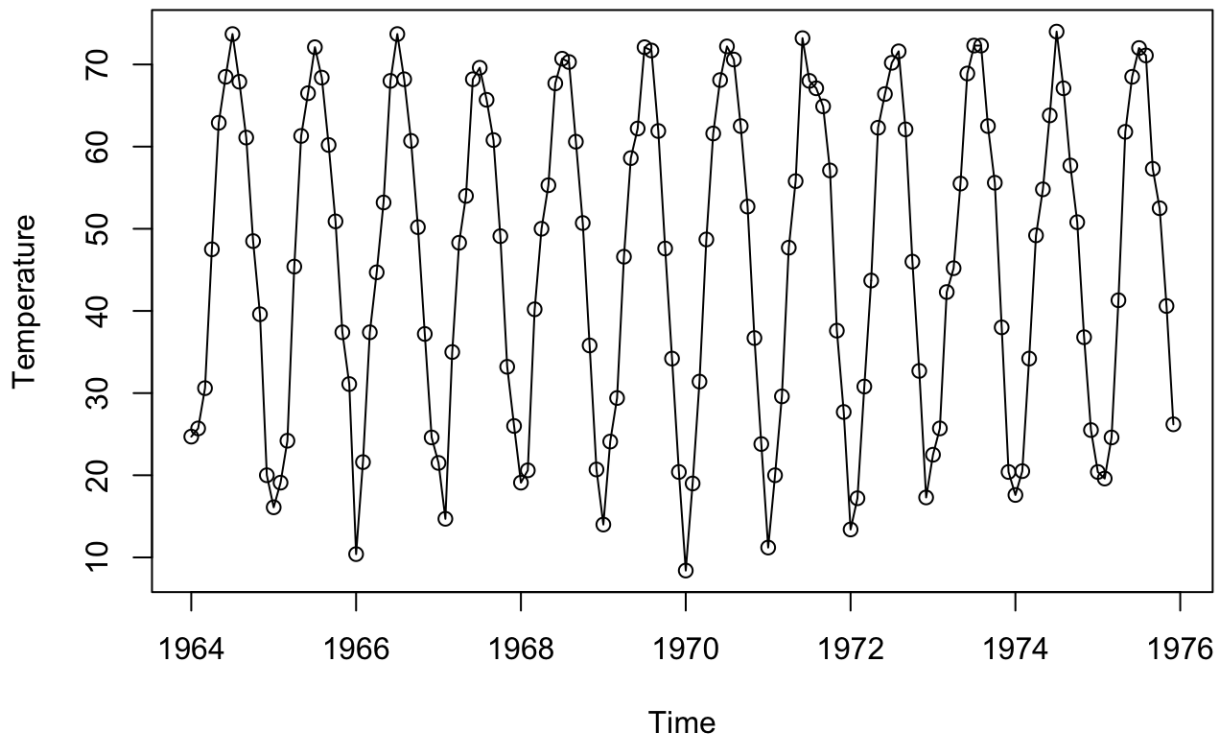
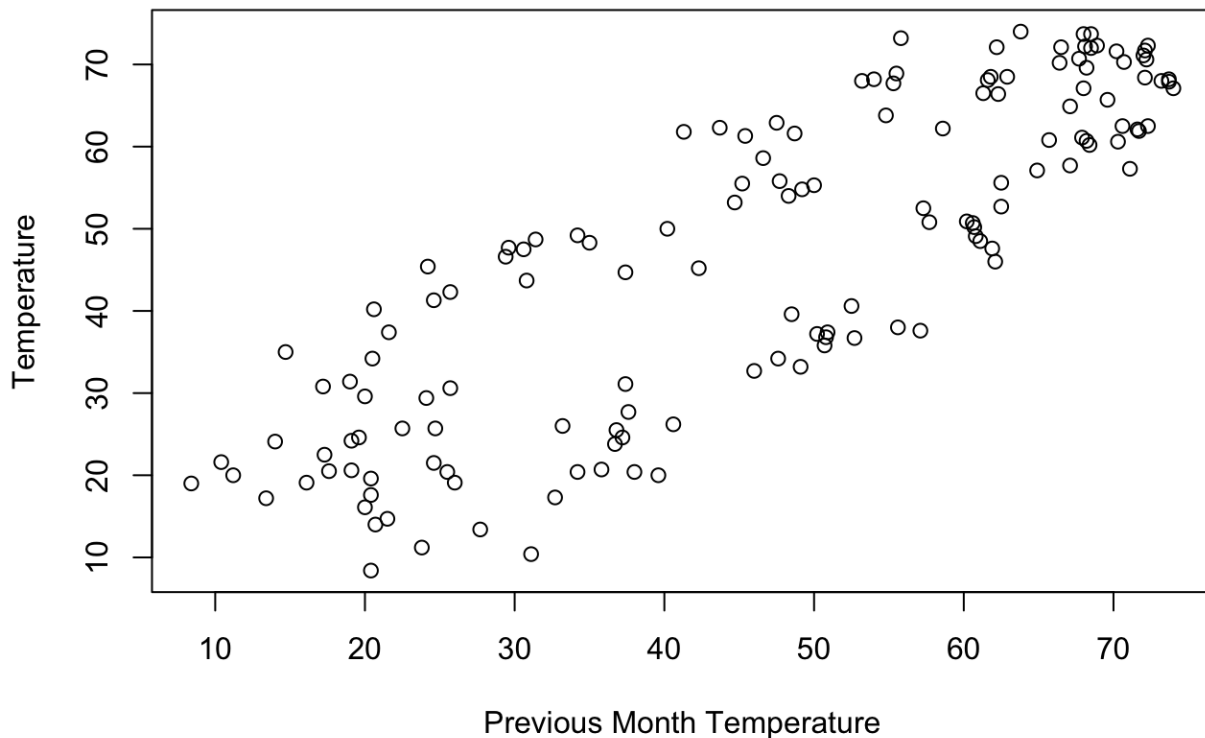


Figure 7 displays **seasonality**. Seasonality for monthly values occurs when observations twelve months apart are related in some manner or another. All Januarys and Februarys are quite cold but they are similar in value and different from the temperatures of the warmer months of June, July, and August, for example. There is still variation among the January values and variation among the June values. Models for such series must accommodate this variation while preserving the similarities. Here the reason for the seasonality is well understood-the Northern Hemisphere's changing inclination toward the sun.

The following shows the scatter plot of neighboring temperature measurements.

```
# win.graph(width=3, height=3,pointsize=8)
plot(y=tempdub,x=zlag(tempdub),ylab='Temperature', xlab='Previous Month Tem
      perature', main = "Figure 8. Scatter plot of average temperature i
      n consecutive months.")
```

**Figure 8. Scatter plot of average temperature in consecutive months.**



As expected, we observe a high correlation between temperatures of succeeding months from Figure 8. However, it is impossible to observe seasonality for this scatter plot.

```
y = tempdub          # Read the abundance data into y
x = zlag(tempdub)     # Generate first lag of the abundance series
index = 2:length(x)   # Create an index to get rid of the first NA value i
                     n x
cor(y[index],x[index]) # Calculate correlation between numerical values in
                     x and y
```

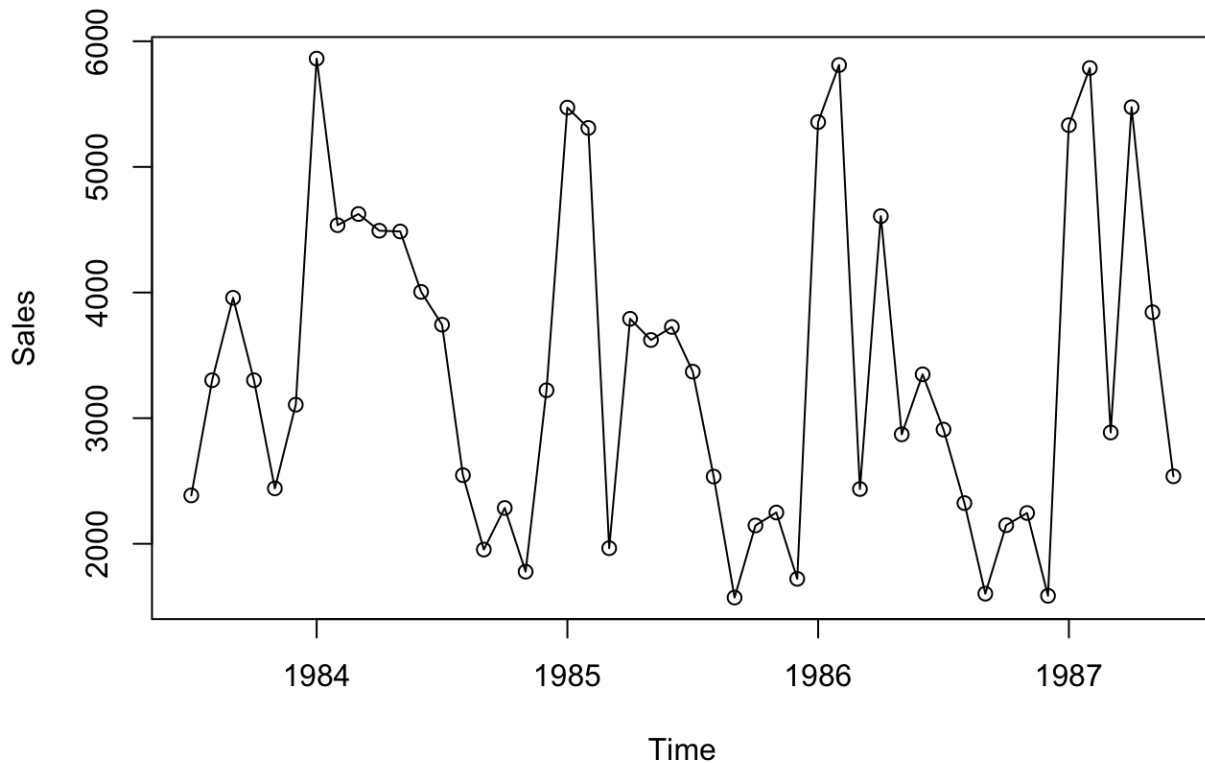
```
## [1] 0.8395499
```

## Monthly Oil Filter Sales

The last example is on the monthly sales to dealers of a specialty oil filter for construction equipment manufactured by John Deere. According to the authors of our textbook, when these data were first presented to one of the authors, the manager said, "There is no reason to believe that these sales are seasonal." Seasonality would be present if January values tended to be related to other January values, February values tended to be related to other February values, and so forth. The time series plot in Figure 9 is not designed to display seasonality especially well.

```
# win.graph(width=4.875, height=2.5,pointsize=8)
data(oilfilters)
plot(oilfilters,type='o',ylab='Sales', main = "Figure 9. Time series plot o
      f monthly oil filter sales series.")
```

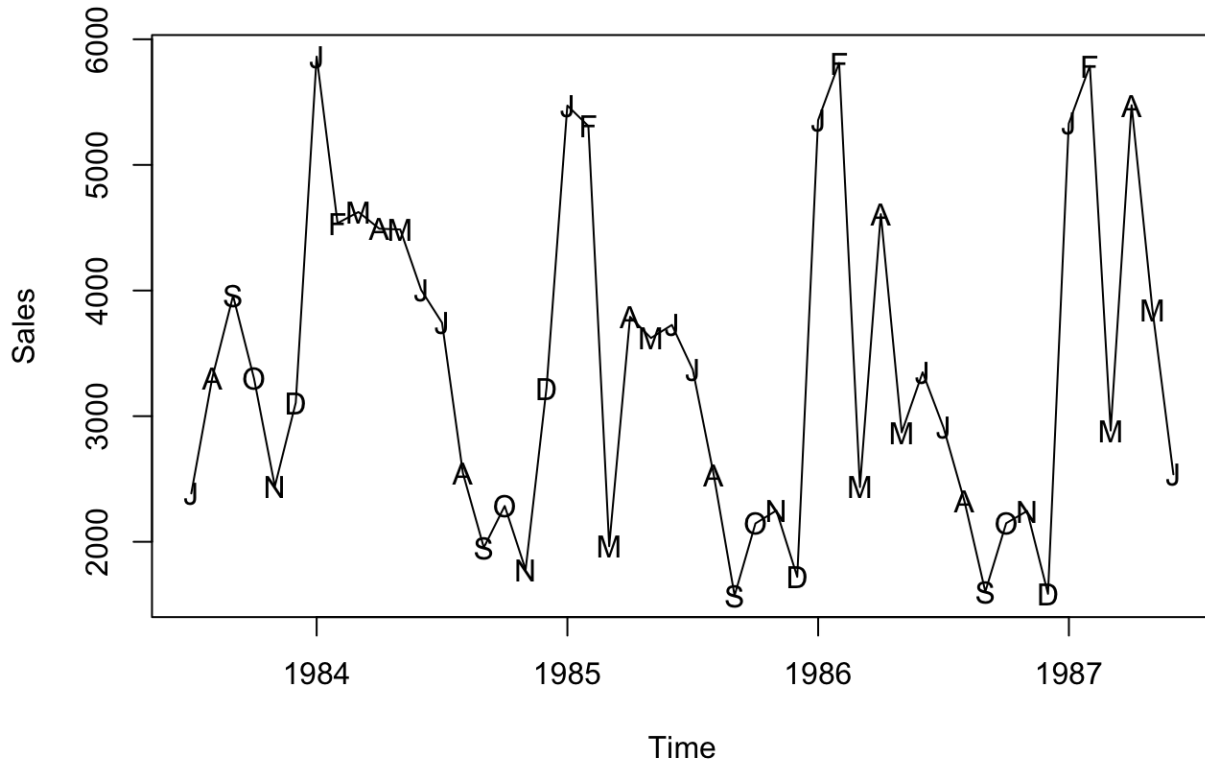
**Figure 9. Time series plot of monthly oil filter sales series.**



To exhibit the seasonality, we can build a more informative plot labeling the months with the first letters of their names. In this version, all January values are plotted with the character J, all Februarys with F, all Marches with M, and so forth. With these plotting symbols, it is much easier to see that sales for the winter months of January and February all tend to be high, while sales in September, October, November, and December are generally quite low. The seasonality in the data is much easier to see from the modified time series plot in Figure 10.

```
plot(oilfilters,type='l',ylab='Sales', main = "Figure 10. Time series plot
      of monthly oil filter sales series with labels.")
points(y=oilfilters,x=time(oilfilters), pch=as.vector(season(oilfilters)))
```

**Figure 10. Time series plot of monthly oil filter sales series with labels**



Here, the command `time(oilfilters)` puts continuous time points on each measurement such that

```
##           Jan           Feb           Mar           Apr           May           Jun           Jul
Aug
## 1983                                     1983.500 198
3.583
## 1984 1984.000 1984.083 1984.167 1984.250 1984.333 1984.417 1984.500 198
4.583
## 1985 1985.000 1985.083 1985.167 1985.250 1985.333 1985.417 1985.500 198
5.583
## 1986 1986.000 1986.083 1986.167 1986.250 1986.333 1986.417 1986.500 198
6.583
## 1987 1987.000 1987.083 1987.167 1987.250 1987.333 1987.417
##           Sep           Oct           Nov           Dec
## 1983 1983.667 1983.750 1983.833 1983.917
## 1984 1984.667 1984.750 1984.833 1984.917
## 1985 1985.667 1985.750 1985.833 1985.917
## 1986 1986.667 1986.750 1986.833 1986.917
## 1987
```

The command `as.vector(season(oilfilters))` creates a vector puts months on each measurement such that

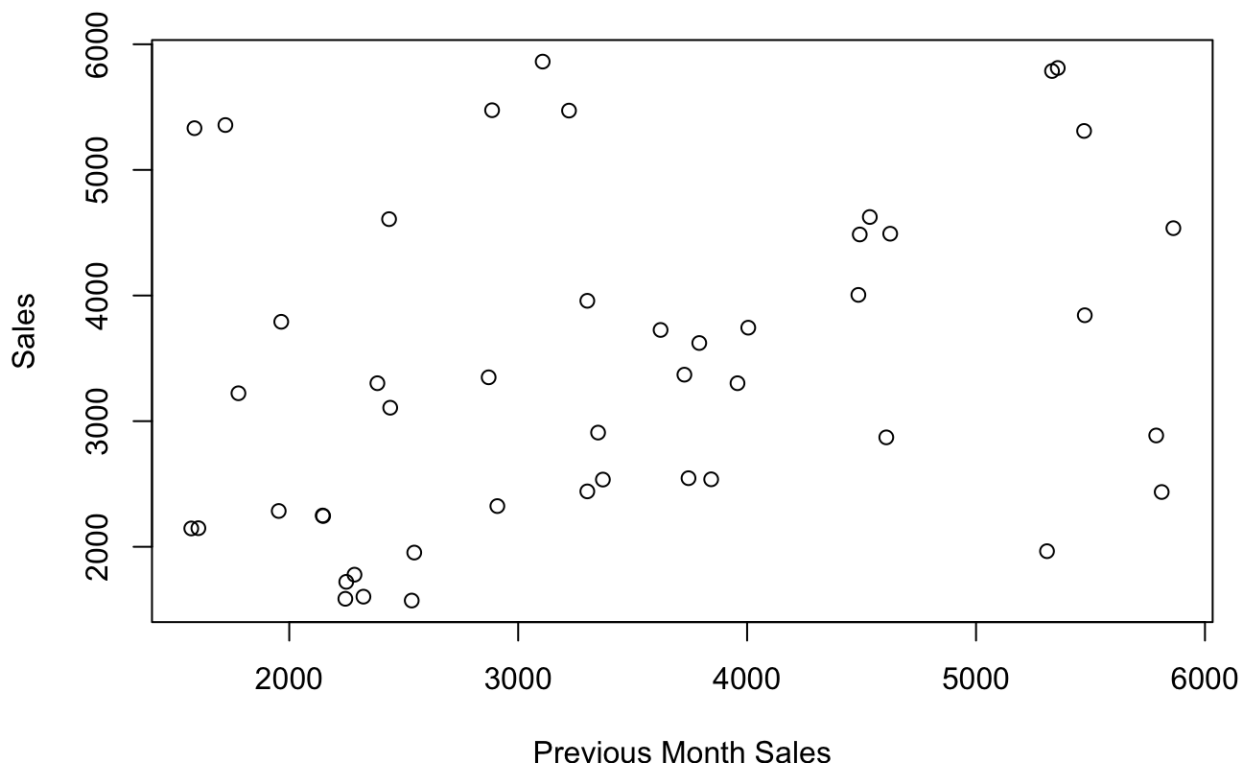
```
## [1] "July"      "August"    "September" "October"   "November"  "Decemb
er"
## [7] "January"   "February"  "March"     "April"    "May"       "June"
## [13] "July"      "August"    "September" "October"   "November"  "Decemb
er"
## [19] "January"   "February"  "March"     "April"    "May"       "June"
## [25] "July"      "August"    "September" "October"   "November"  "Decemb
er"
## [31] "January"   "February"  "March"     "April"    "May"       "June"
## [37] "July"      "August"    "September" "October"   "November"  "Decemb
er"
## [43] "January"   "February"  "March"     "April"    "May"       "June"
```

Then, the `points()` function (<https://stat.ethz.ch/R-manual/R-patched/library/graphics/html/points.html>) writes the first letter of each month over the time series plot generated by `plot()` function.

Figure 11 shows the scatter plot of neighboring monthly sales of oil filters.

```
# win.graph(width=3, height=3,pointsize=8)
plot(y=oilfilters,x=zlag(oilfilters),ylab='Sales', xlab='Previous Month Sal
es',main = "Figure 11. Scatter plot of oil filter sales in consecu
tive months.")
```

**Figure 11. Scatter plot of oil filter sales in consecutive months.**



And the amount of correlation between sales of succeeding months is

```
y = oilfilters          # Read the abundance data into y
x = zlag(oilfilters)     # Generate first lag of the abundance series
index = 2:length(x)      # Create an index to get rid of the first NA value i
                        n x
cor(y[index],x[index]) # Calculate correlation between numerical values in
                        x and y
```

```
## [1] 0.3142145
```

Although we observe a seasonal effect, autocorrelation between monthly sales is slightly correlated. So, we should note that having seasonality in time series data does not imply a significant correlation.

## A Model-Building Strategy

It is not a straightforward task to find appropriate models for time series data. There are several components that a data scientist should deal with. Throughout the course we will follow a multistage model-building strategy with the following three steps:

- **model specification**, (or identification)
- **model fitting**, and
- **model diagnostics**.

In model specification (or identification), the classes of time series models are selected that may be appropriate for a given observed series. In choosing a model, we shall attempt to adhere to the **principle of parsimony**; that is, the model used should require the smallest number of parameters that will adequately represent the time series.

Albert Einstein is quoted in Parzen (1982, p. 68) as remarking that “everything should be made as simple as possible but not simpler.”

The model fitting consists of finding the best possible estimates of a number of parameters involved in the model using a statistical estimation method such as least squares, maximum likelihood estimation, or Bayesian estimation.

Model diagnostics is concerned with assessing the quality of the model that we have specified and estimated. We try to answer the questions:

- How well does the model fit the data?
- Are the assumptions of the model reasonably well satisfied?

If no inadequacies are found, the modeling may be assumed to be complete, and the model may be used, for example, to forecast future values. Otherwise, we choose another model in the light of the inadequacies found; that is, we return to the model specification step. In this way, we cycle through the three steps until an acceptable model is found.

## Time Series and Stochastic Processes

The sequence of random variables  $\{Y_t : t = 0, \pm 1, \pm 2, \pm 3, \dots\}$  is called a *stochastic process* and serves as a model for an observed time series. It is known that the complete probabilistic structure of such a process is determined by the set of distributions of all finite collections of the  $Y$ 's. Much of the information in these joint distributions can be described in terms of *means, variances, and covariances*. So, we will focus on the measures mostly based on the first and second moments.

## Means, Variances, and Covariances

For a stochastic process  $\{Y_t : t = 0, \pm 1, \pm 2, \pm 3, \dots\}$ , the *mean function* is defined by

$$\mu_t = E(Y_t), \text{ for } t = 0, \pm 1, \pm 2, \pm 3, \dots$$

Actually,  $\mu_t$  is just the expected value of the process at time  $t$ .

The *autocovariance function*,  $\gamma_{t,s}$  is defined as

$$\gamma_{t,s} = \text{Cov}(Y_t, Y_s), \text{ for } t, s = 0, \pm 1, \pm 2, \pm 3, \dots$$

where  $\text{Cov}(Y_t, Y_s) = E[(Y_t - \mu_t)(Y_s - \mu_s)] = E(Y_t Y_s) - \mu_t \mu_s$ .

The *autocorrelation function*,  $\rho_{t,s}$ , is given by

$$\rho_{t,s} = \text{Corr}(Y_t, Y_s), \text{ for } t, s = 0, \pm 1, \pm 2, \pm 3, \dots$$

where

$$\text{Corr}(Y_t, Y_s) = \frac{\text{Cov}(Y_t, Y_s)}{\sqrt{\text{Var}(Y_t)\text{Var}(Y_s)}} = \frac{\gamma_{t,s}}{\sqrt{\gamma_{t,t}\gamma_{s,s}}}.$$

Please note that

$$\gamma_{t,t} = \text{Cov}(Y_t, Y_t) = E[(Y_t - \mu_t)(Y_t - \mu_t)] = E[(Y_t - \mu_t)^2] = E(Y_t^2) - \mu_t^2 = \text{Var}(Y_t)$$

and

$$\gamma_{s,s} = \text{Cov}(Y_s, Y_s) = E[(Y_s - \mu_s)(Y_s - \mu_s)] = E[(Y_s - \mu_s)^2] = E(Y_s^2) - \mu_s^2 = \text{Var}(Y_s).$$

Consequently,  $\rho_{t,t} = \rho_{s,s} = 1$ .

Because  $\gamma_{t,s} = \gamma_{s,t}$ , we have  $\rho_{t,s} = \rho_{s,t}$ . And because  $|\gamma_{t,s}| \leq \sqrt{\gamma_{t,t}\gamma_{s,s}}$ , we have  $|\rho_{t,s}| \leq 1$ .

Values of  $\rho_{t,s}$  near  $\pm 1$  indicate strong (linear) dependence, whereas values near zero indicate weak (linear) dependence. If  $\rho_{t,s} = 0$ , we say that  $Y_t$  and  $Y_s$  are **uncorrelated**.

Because most of the context is based on the covariance and correlations, we will go a little bit further. To investigate the covariance properties of various time series models, the following result will be used repeatedly: If  $c_1, c_2, \dots, c_m$  and  $d_1, d_2, \dots, d_n$  are constants and  $t_1, t_2, \dots, t_m$  and  $s_1, s_2, \dots, s_n$  are time points, then the contrivance of two linear combinations of time series observations is calculated by

$$Cov\left[\sum_{i=1}^m c_i Y_{t_i}, \sum_{j=1}^n d_j Y_{s_j}\right] = \sum_{i=1}^m \sum_{j=1}^n c_i d_j Cov(Y_{t_i}, Y_{s_j}).$$

As a special case, we obtain the variance of a linear combination of time series observations as

$$Var\left[\sum_{i=1}^m c_i Y_{t_i}\right] = \sum_{i=1}^m c_i^2 Var(Y_{t_i}) + 2 \sum_{i < j} c_i c_j Cov(Y_{t_i}, Y_{t_j}).$$

## The Random Walk

Let  $e_1, e_2, \dots$  be a sequence of independent, identically distributed random variables each with zero mean and variance  $\sigma_e^2$ . The observed time series,  $\{Y_t : t = 1, 2, \dots\}$ , is constructed as follows:

$$\begin{aligned} Y_1 &= e_1 \\ Y_2 &= e_1 + e_2 \\ &\vdots \\ Y_t &= e_1 + e_2 + \dots + e_t \end{aligned}$$

Apparently, this system of equations can be written as

$$Y_t = Y_{t-1} + e_t$$

with *initial condition*  $Y_1 = e_1$ .

If the  $e$ 's are interpreted as the sizes of the "steps" taken (forward or backward) along a number line, then  $Y_t$  is the position of the **random walker** at time  $t$ .

We obtain the mean and variance of the random walk process as follows:

$$\mu_t = E(Y_t) = E(e_1 + e_2 + \dots + e_t) = E(e_1) + E(e_2) + \dots + E(e_t) = 0 + 0 + \dots + 0.$$

So,  $\mu_t = 0$  for all  $t$ . As for the variance,

$$Var(Y_t) = Var(e_1 + e_2 + \dots + e_t) = Var(e_1) + Var(e_2) + \dots + Var(e_t) = \sigma_e^2 + \sigma_e^2 + \dots + \sigma_e^2.$$

So we have a linearly increasing variance  $Var(Y_t) = t\sigma_e^2$  with time.

For  $1 \leq t \leq s$ , we have the following autocorrelation function,

$$\gamma_{t,s} = t\sigma_e^2.$$

Autocorrelation function is straightforwardly obtained as follows:

$$\rho_{t,s} = \sqrt{\frac{t}{s}}$$

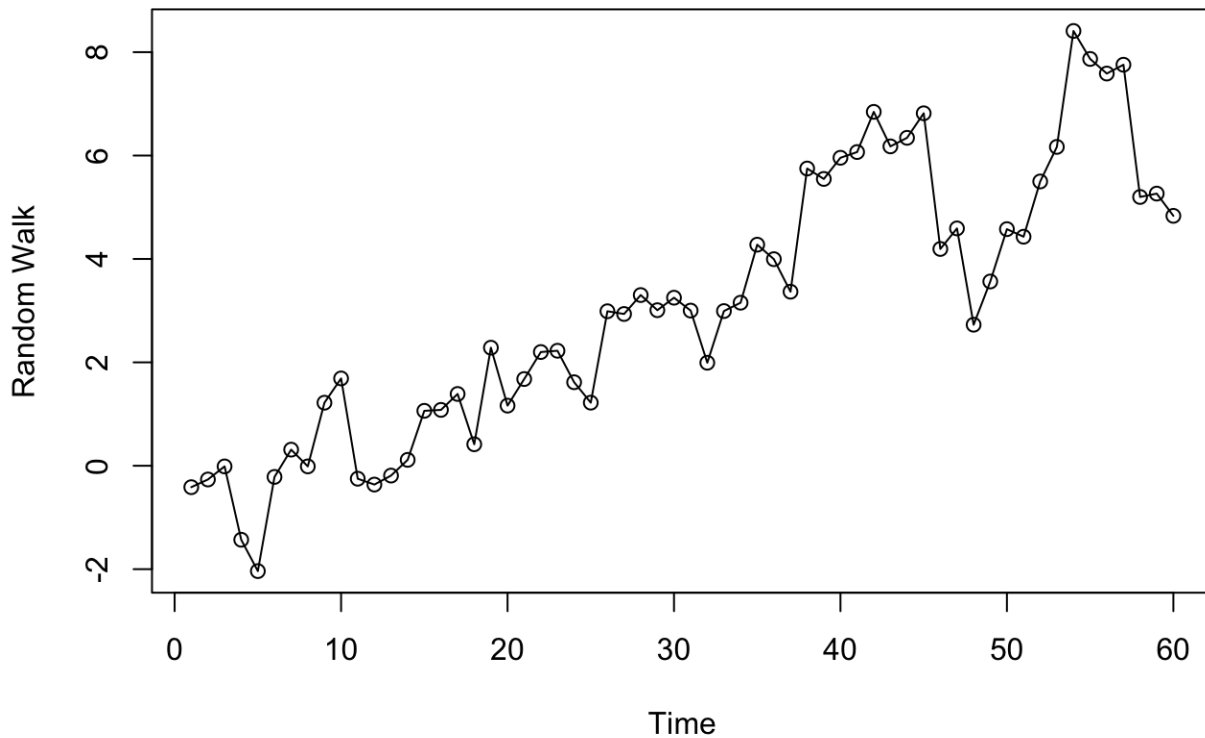


for  $1 \leq t \leq s$ . For example, we have the following autocorrelation values  $\rho_{1,2} = 0.707$ ,  $\rho_{8,9} = 0.943$ ,  $\rho_{24,25} = 0.980$ , and  $\rho_{1,25} = 0.200$ . The neighboring time points are more correlated than those distant from each other.

Figure 12 shows a simulated random walk:

```
data(rwalk) # rwalk contains a simulated random walk
plot(rwalk,type='o',ylab='Random Walk', main = "Figure 12. Time series plot
of a simulated random walk.")
```

**Figure 12. Time series plot of a simulated random walk.**



## A Moving Average

Suppose that  $\{Y_t\}$  is constructed as

$$Y_t = \frac{e_t + e_{t-1}}{2}$$

where  $e$ 's are i.i.d. with zero mean and variance  $\sigma_e^2$ . Consequently, we have the following:

$$\mu_t = 0$$

$$\text{Var}(Y_t) = 0.5\sigma_e^2$$

$$\text{Cov}(Y_t, Y_{t-1}) = 0.25\sigma_e^2 \text{ and } \text{Cov}(Y_t, Y_{t-k}) = 0 \text{ for } k > 1, \text{ and}$$

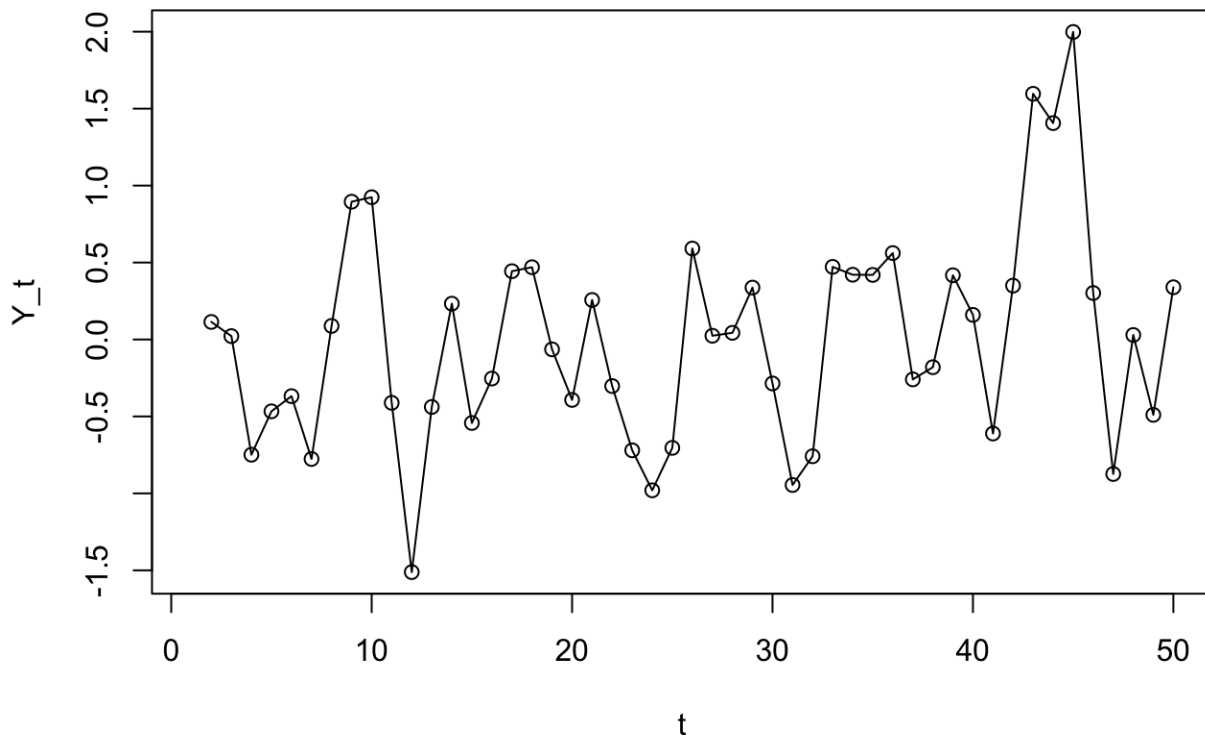
$$\rho_{t,s} = 0.5 \text{ and } \rho_{t,t-k} = 0 \text{ for } k > 1.$$

Values of  $Y$  precisely one-time unit apart have exactly the same correlation no matter where they occur in time and, more generally,  $\rho_{t,t-k}$  is the same for all values of  $t$ . This leads us to the important concept of stationarity.

The following code chunk generates a random realisation of a moving average series in Figure 13.

```
e = rnorm(50,0,1) # Generate standard normal distributed e_t series
Y = array(NA, 50)
for ( t in 2:50){
  Y[t] = (e[t] + e[t-1])/2
}
plot(ts(Y), type = "o", ylab="Y_t", xlab = "t", main = "Figure 13. Time series plot of a moving average series.")
```

**Figure 13. Time series plot of a moving average series.**



## Stationarity

The basic idea of stationarity is that the probability laws that govern the behavior of the process do not change over time. the assumption of stationarity allows us to make statistical inferences about the structure of a stochastic process on the basis of an observed record of that process. Specifically, a process  $\{Y_t\}$  is said to be strictly stationary if the joint distribution of  $Y_{t_1}, Y_{t_2}, \dots, Y_{t_n}$  is the same as the joint distribution of  $Y_{t_1-k}, Y_{t_2-k}, \dots, Y_{t_n-k}$  for all choices of time points  $t_1, t_2, \dots, t_n$  and all choices of time lag  $k$ .

It then follows that both the mean and variance functions are constant for all time:

$$E(Y_t) = E(Y_{t-k}) \text{ and } Var(Y_t) = Var(Y_{t-k}) \text{ for all } t \text{ and } k.$$

The covariance between  $Y_t$  and  $Y_s$  depends on time only through the time difference  $|t - s|$  and not otherwise on the actual times  $t$  and  $s$ .

Thus, for a stationary process, to simplify our notation, we can write

$$\gamma_k = Cov(Y_t, Y_{t-k}) \text{ and } \rho_k = Corr(Y_t, Y_{t-k}) = \frac{\gamma_k}{\gamma_0}.$$

General properties of a stationary process are

$$\gamma_0 = Var(Y_t)$$

$$\gamma_k = \gamma_{-k}$$

$$|\gamma_k| \leq \gamma_0$$

and correspondingly,

$$\rho_0 = 1$$

$$\rho_k = \rho_{-k}$$

$$|\rho_k| \leq 1.$$

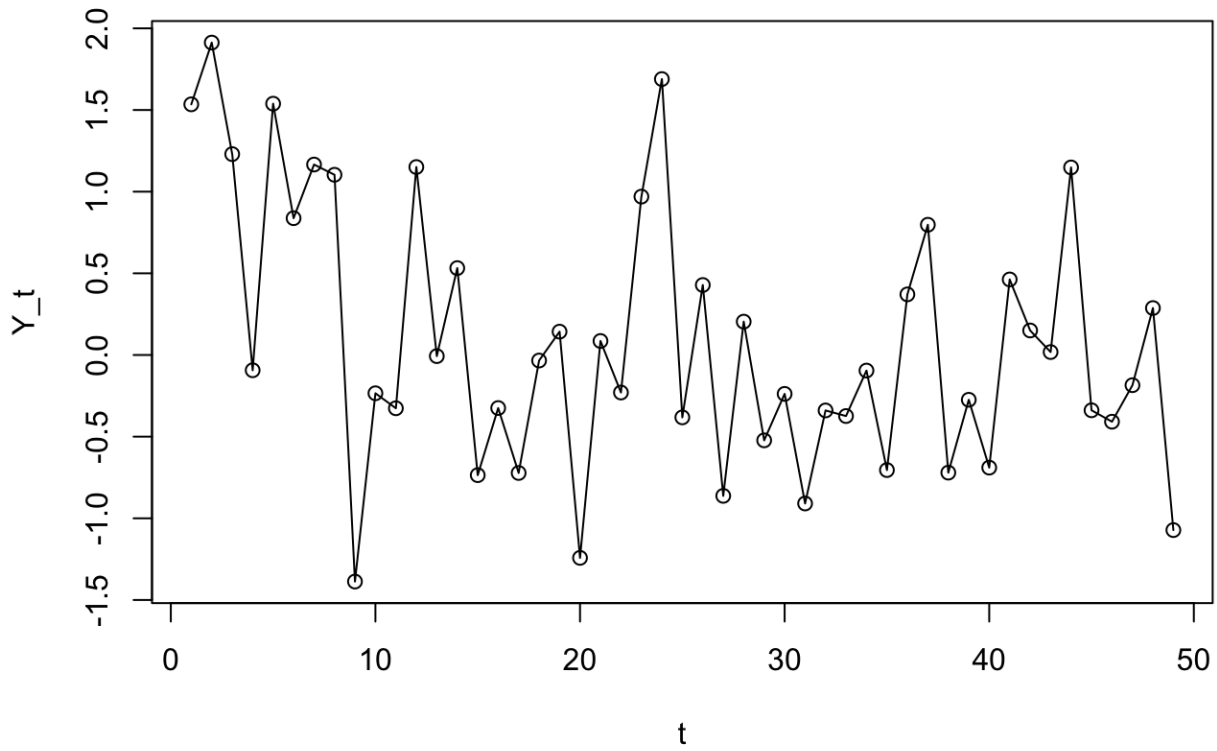
If a process is strictly stationary and has a finite variance, then the covariance function must depend only on the time lag. A definition that is similar to that of strict stationarity but is mathematically weaker is the following: A stochastic process  $\{Y_t\}$  is said to be weakly (or second-order) stationary if

1. The mean function is constant over time, and
2.  $\gamma_{t,t-k} = \gamma_{0,k}$  for all time  $t$  and lag  $k$ .

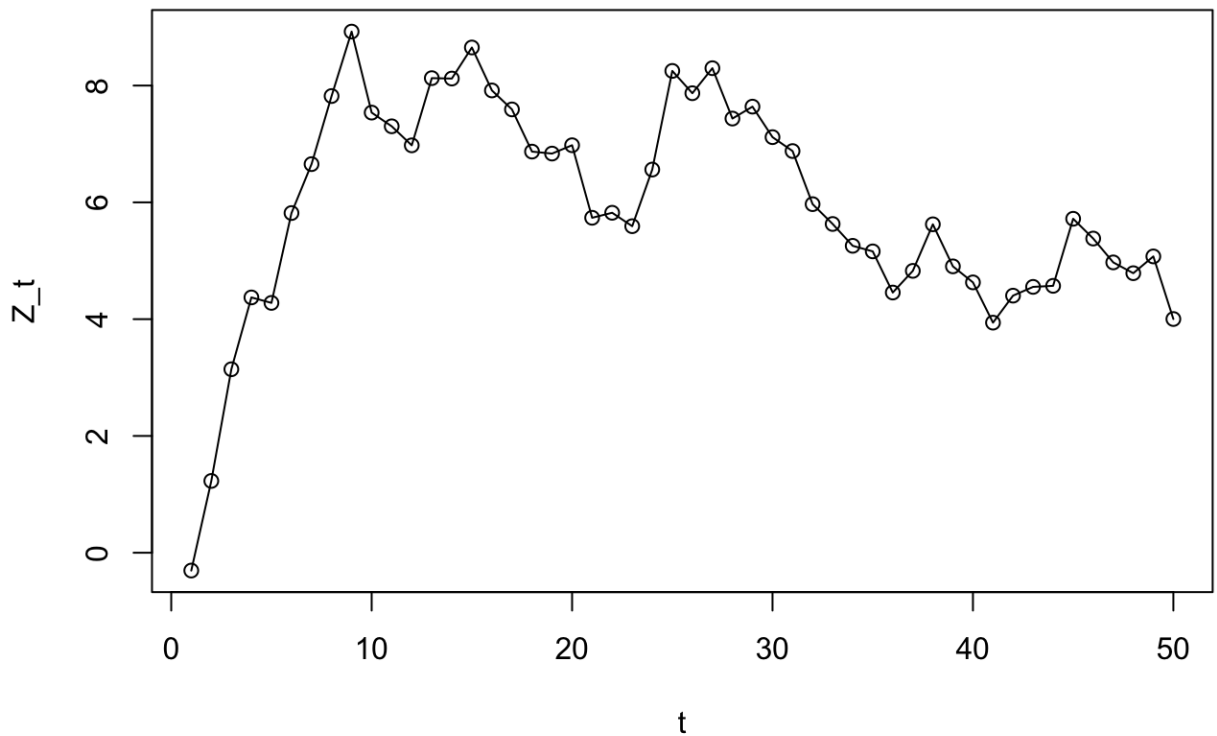
In this course, the term stationary when used alone will always refer to this weaker form of stationarity.

Figure 14 and 15 show a stationary and a nonstationary randomly generated series. Which one is for the non-stationary series?

**Figure 14. Time series plot of a randomly generated stationary series.**



**Figure 15. Time series plot of a randomly generated nonstationary series.**



White Noise

One of the important and mostly used stationary processes is called **white noise** process. A white noise process is actually a sequence of i.i.d. random variables  $\{e_t\}$ . many useful processes can be constructed from white noise.

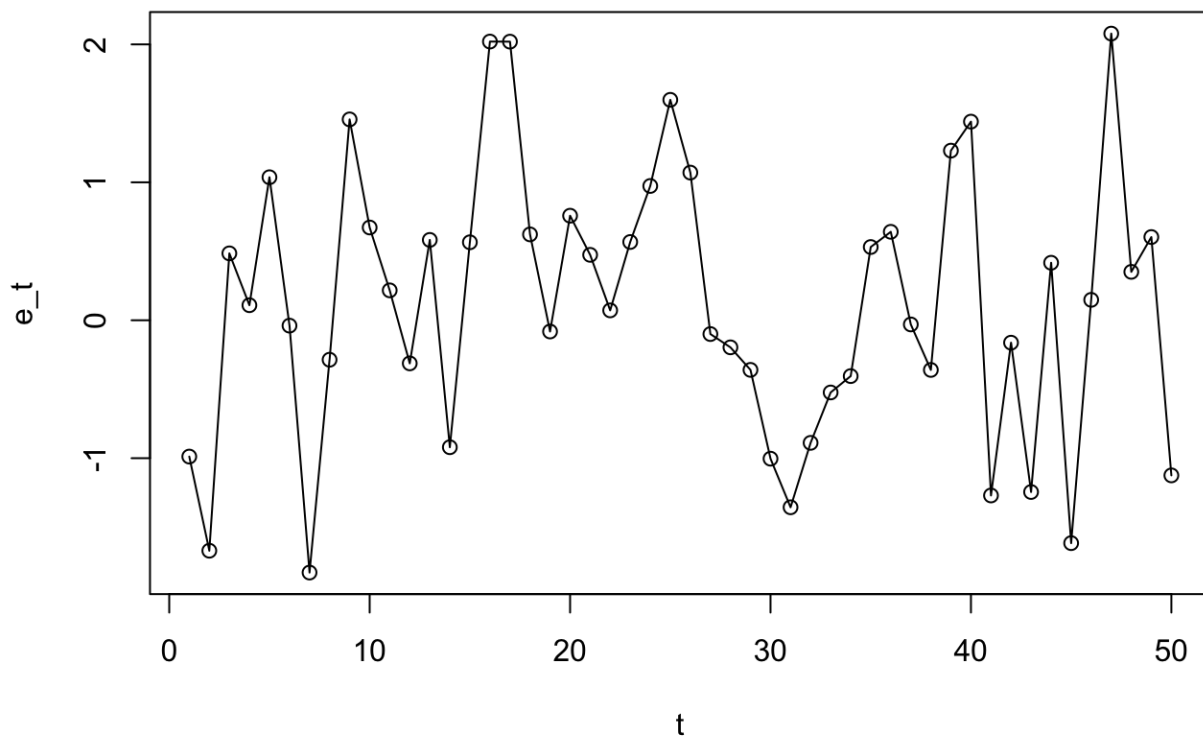
For a white noise process,  $\mu_t = E(e_t)$  is constant and  $\gamma_k = \text{Var}(e_t)$  for  $k = 0$  and  $\gamma_k = 0$  for  $k \neq 0$ . Thus, we can write  $\rho_k = 1$  for  $k = 0$  and  $\rho_k = 0$  for  $k \neq 0$ . We usually assume that the white noise process has mean zero and denote  $\text{Var}(e_t)$  by  $\sigma_e^2$ .

The moving average also defines a stationary process. For a moving average process,  $Y_t = (e_t + e_{t-1})/2$ , we have  $\rho_k = 1$  for  $k = 0$ ,  $\rho_k = 0.5$  for  $|k| = 1$ , and  $\rho_k = 0$  for  $|k| \geq 2$ .

Figure 16 shows an example of a white noise series:

```
e = rnorm(50,0,1)
plot(ts(e), type = "o", ylab="e_t", xlab = "t", main = "Figure 16. Time series plot of a white noise series.")
```

**Figure 16. Time series plot of a white noise series.**



## Random Cosine Wave

Let us define the following process:

$$Y_t = \cos \left[ 2\pi \left( \frac{t}{12} + \Phi \right) \right] \text{ for } t = 0, \pm 1, \pm 2, \dots$$

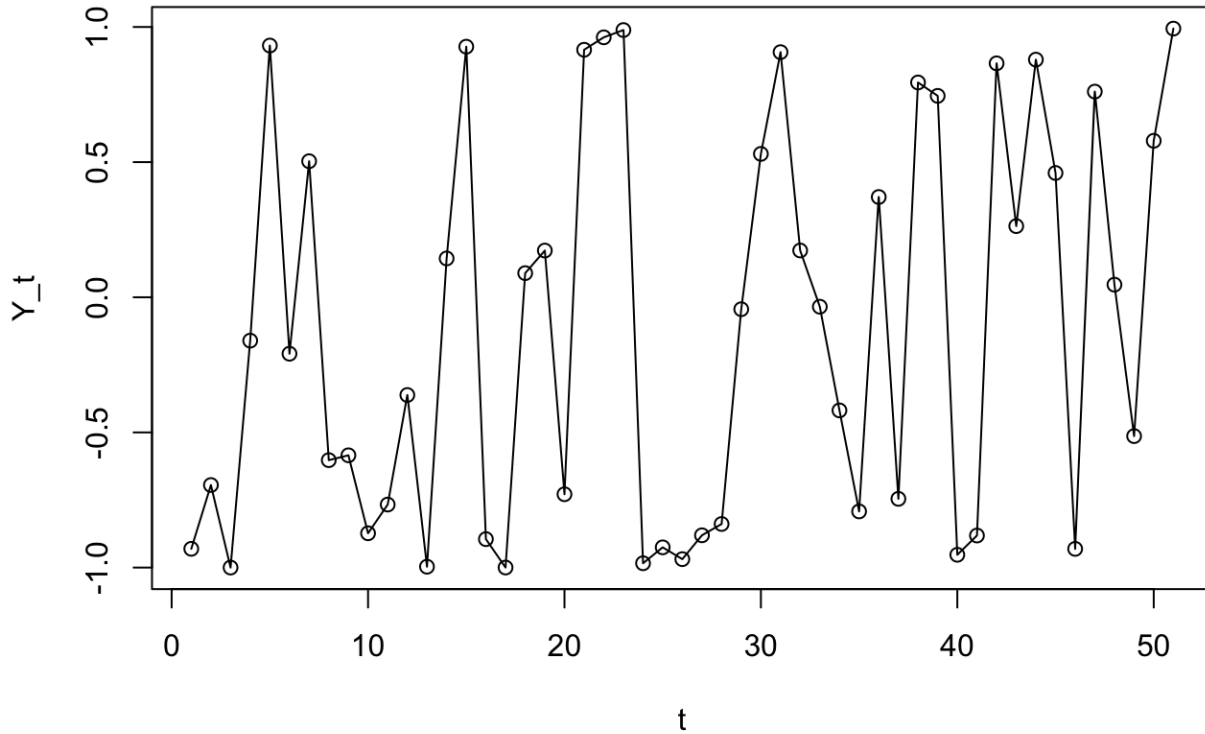
where  $\Phi \sim \text{Uniform}(0, 1)$ . Figure 17 displays the time series plot of this process:

```

phi = runif(51,0,1)
t = seq(0,50,1)
Y = cos(2*pi*((t/12)+phi))
plot(ts(Y), type = "o", ylab="Y_t", xlab="t", main = "Figure 17. Time series plot of a random cosine wave.")

```

**Figure 17. Time series plot of a random cosine wave.**



Statistical properties of this process are

$$\begin{aligned}
 E(Y_t) &= \int_0^1 \cos \left[ 2\pi \left( \frac{t}{12} + \phi \right) \right] f(\phi) d\phi \\
 &= \int_0^1 \cos \left[ 2\pi \left( \frac{t}{12} + \phi \right) \right] \cdot 1 d\phi \\
 &= \frac{1}{2\pi} \sin \left( 2\pi \frac{t}{12} + \phi \right) \Big|_0^1 \\
 &= \frac{1}{2\pi} \left[ \sin \left( 2\pi \frac{t}{12} + 2\pi \right) - \sin \left( 2\pi \frac{t}{12} \right) \right] \\
 &= 0
 \end{aligned}$$

for all  $t$ . Also,

$$\gamma_{t,s} = \frac{1}{2} \cos \left[ 2\pi \left( \frac{|t-s|}{12} \right) \right].$$

So the process is stationary with the autocorrelation function

$$\rho_k = \cos\left(2\pi \frac{k}{12}\right) \text{ for } t = 0, \pm 1, \pm 2, \dots$$

With this example, we observe that it will be difficult to assess whether or not stationarity is a reasonable assumption for a given time series on the basis of the time sequence plot of the observed data.

As another example, the random walk constructed with  $Y_t = e_1 + e_2 + \dots + e_t$  is not stationary due to non-constant variance function  $Var(Y_t) = t\sigma_e^2$  also its covariance function  $\gamma_{t,s} = t\sigma_e^2$  does not depend only on time lag for  $0 \leq t < s$ .

Instead of going on with  $Y_t$  directly, we can consider the differences of successive  $Y$ -values, denoted  $\Delta Y_t$ . Then  $Y_t - Y_{t-1} = e_t$ , so the differenced series,  $\{\Delta Y_t\}$ , is stationary.

Many real time series cannot be reasonably modeled by stationary processes since they are not in statistical- equilibrium but are evolving over time. However, we can frequently transform non- stationary series into stationary series by simple techniques such as differencing.

## Summary

In this module, we introduced a special data type called time series data. We use simple visualisations to plot time series data for descriptive purposes. Then we focused on the fundamentals of time series analysis such as mean function, autocovariance function, and autocorrelation function. We illustrated these concepts with the basic processes: the random walk, white noise, a simple moving average, and a random cosine wave. Finally, the fundamental concept of stationarity is introduced and the idea of differentiating time series to ensure stationarity is highlighted.