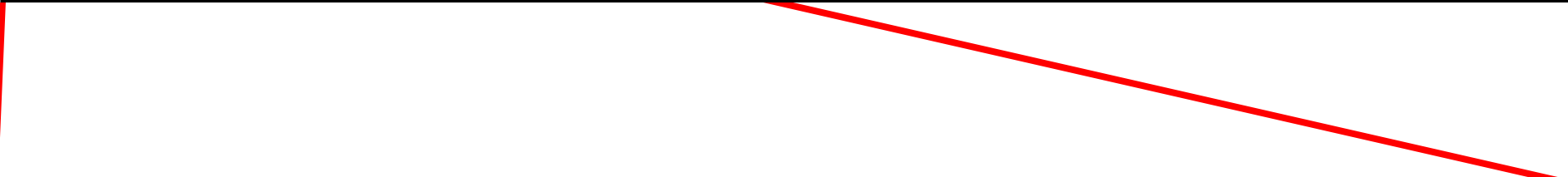


BMMS2074 Statistics for Data Science

# **TOPIC 5**

## **SIMPLE DESCRIPTIVE TECHNIQUES**

Simple Descriptive Techniques								
<ul style="list-style-type: none"><li>• Time plot</li><li>• Transformation</li><li>• Analysing series which contain a trend – curve fitting, filtering and differencing</li><li>• Analysing series which contain seasonal variation</li><li>• Autocorrelation</li><li>• Correlogram</li></ul>	1, 3	3	1.5			1	3.5	9



Simple Descriptive Techniques
<ul style="list-style-type: none"><li>• Time plot</li><li>• Transformation</li><li>• Analysing series which contain a trend – curve fitting, filtering and differencing</li><li>• Analysing series which contain seasonal variation</li><li>• Autocorrelation</li><li>• Correlogram</li></ul>

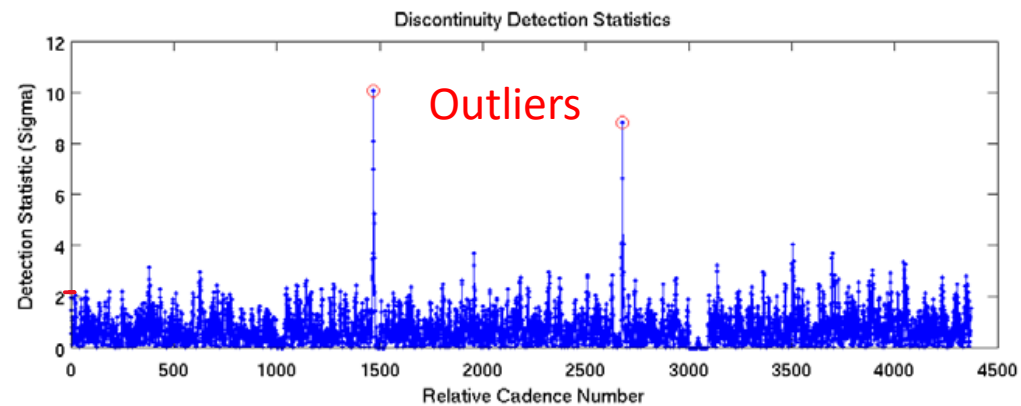
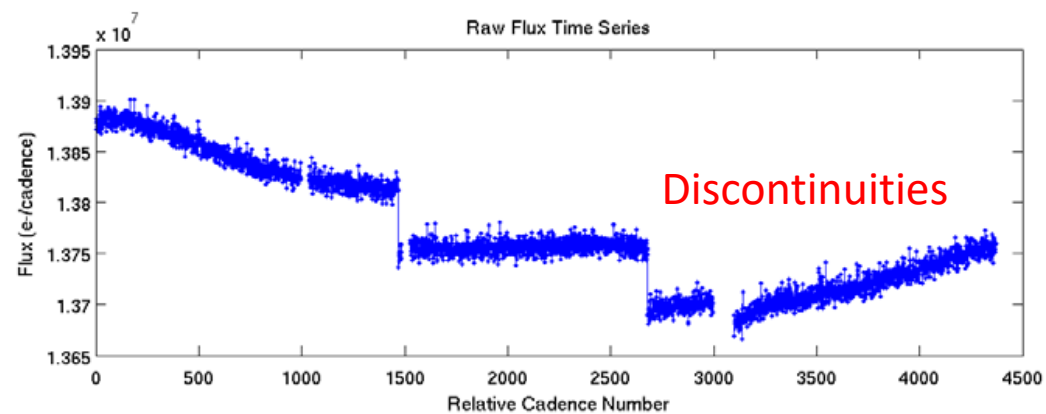
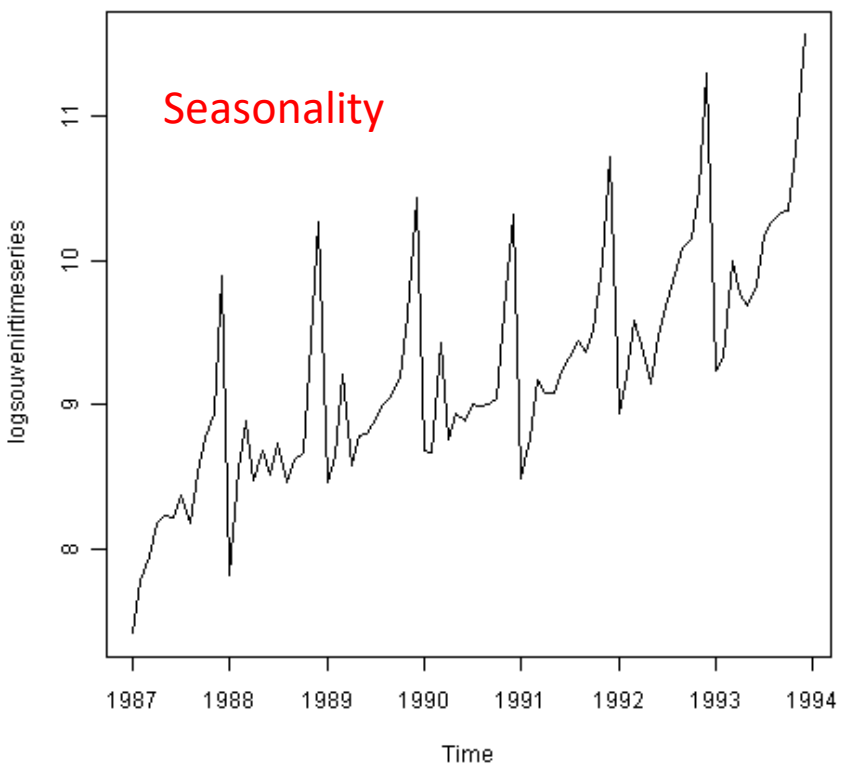
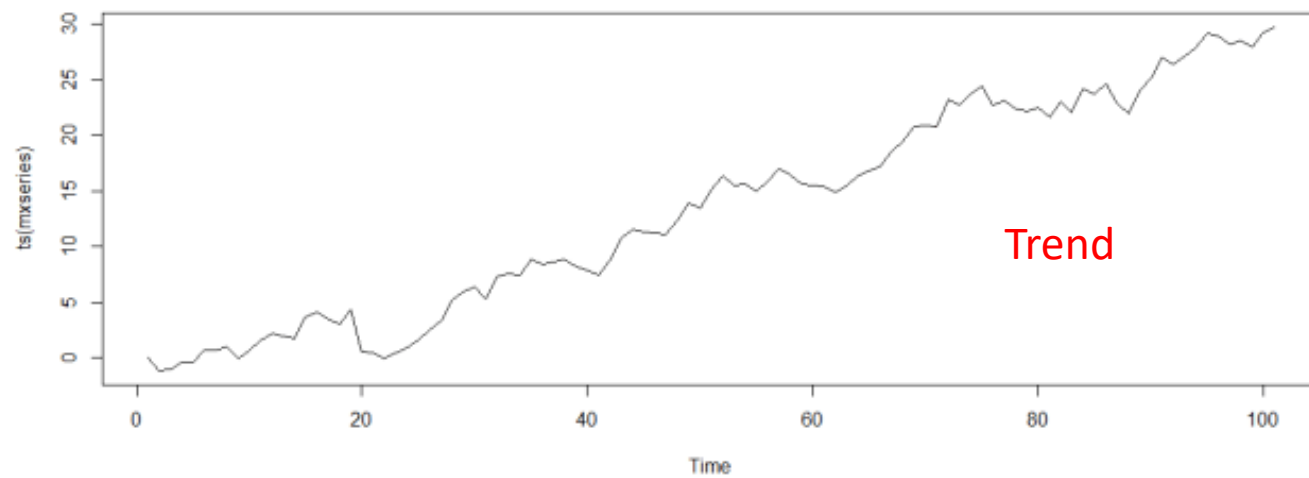
## **5.0 Introduction**

Descriptive methods should generally be tried before attempting more complicated procedures, because they can be vital in “cleaning” the data, and then getting a “feel” for them, before trying to generate ideas as regards a suitable model. So, we must focus on ways of understanding typical time-series effects, such as trend, seasonality and correlations between successive observations.

### **5.1 Time plot**

The first step in analysing a set of data is to plot the observations against time. This will often show up the most important properties of a series. Features such as trend, seasonality, discontinuities, will usually be visible if present in the series. If the series is approximately stationary, it will be useful to compute the mean and standard deviation of the observations.

A time plot is a two-dimensional plot of time series data where the variable being measured is plotted on the vertical axis while time is always plotted on an even scale along the horizontal axis.



### Example 5.1

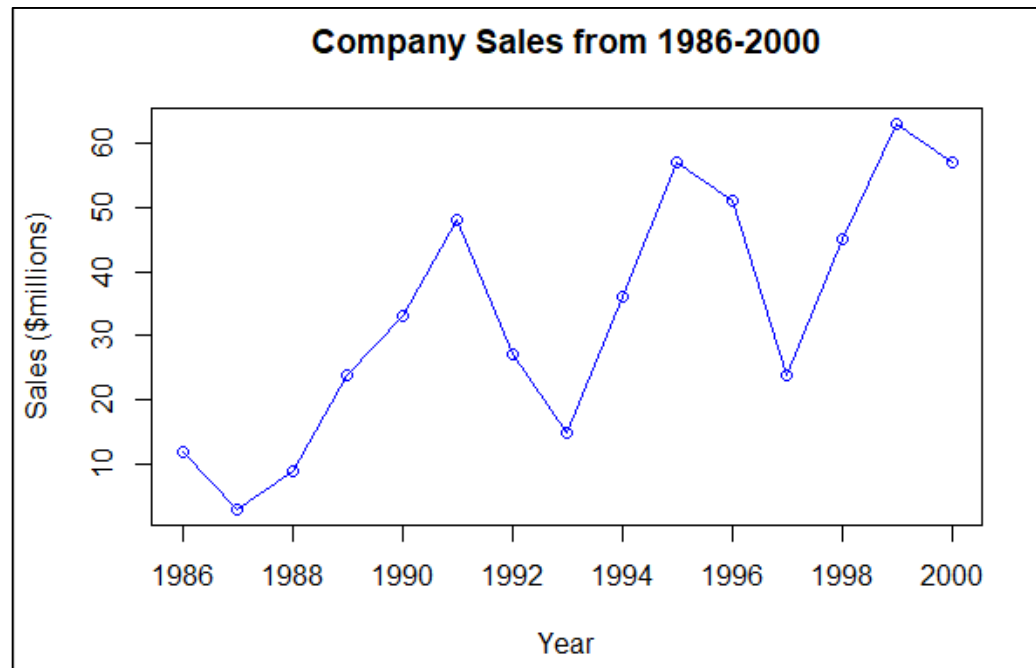
The table shows the sales of a company in millions of dollars.

Year	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000
Sales (RM'000)	12	3	9	24	33	48	27	15	36	57	51	24	45	63	57

Plot a time series graph and comment on the pattern of this series.

R-codes:

```
Y <- c(12,3,9,24,33,48,27,15,36,57,51,24,45,63,57)
Y <- ts(Y, frequency = 1, start = 1986)
plot(Y, type="o", xlab="Year", ylab="Sales ($millions)", main="Company Sales
from 1986-2000")
```



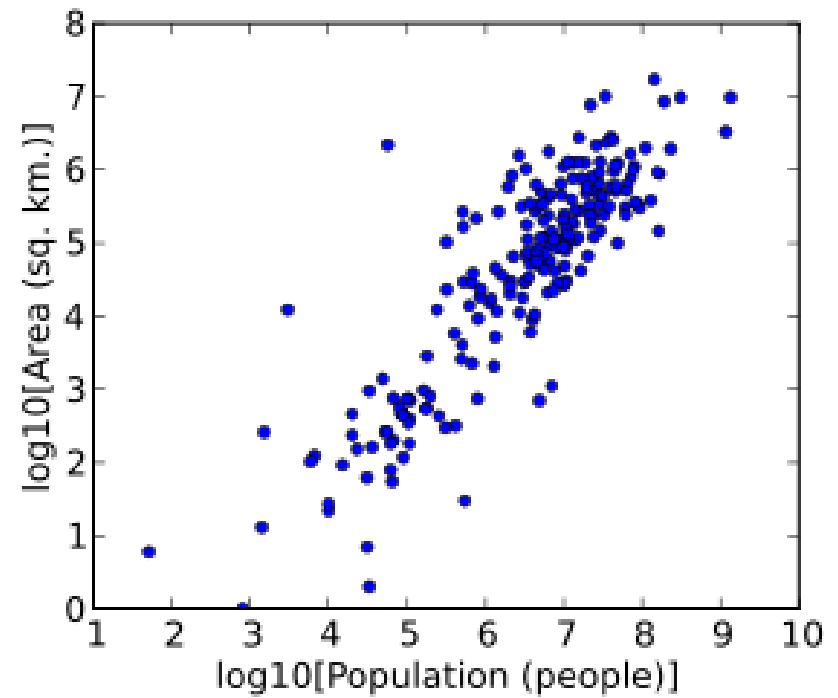
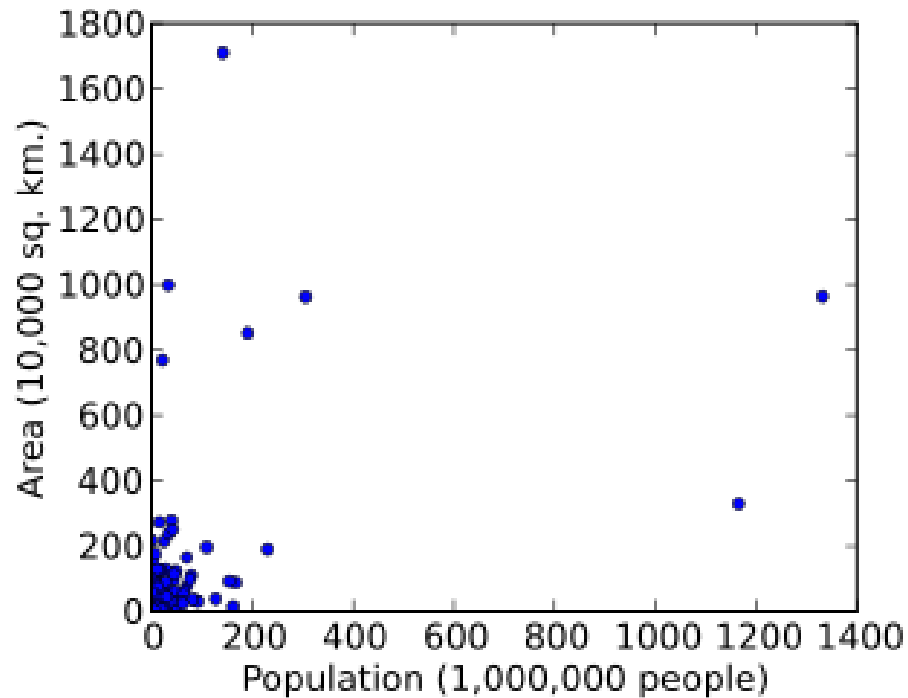
The features of this graph are its cyclical nature and an apparent upward long-term trend.

## 5.2 Transformations

Plotting the data may indicate if it is desirable to transform the values of the observed variable.

### Example

### Simple Linear Regression



The two main reasons for making a transformation are

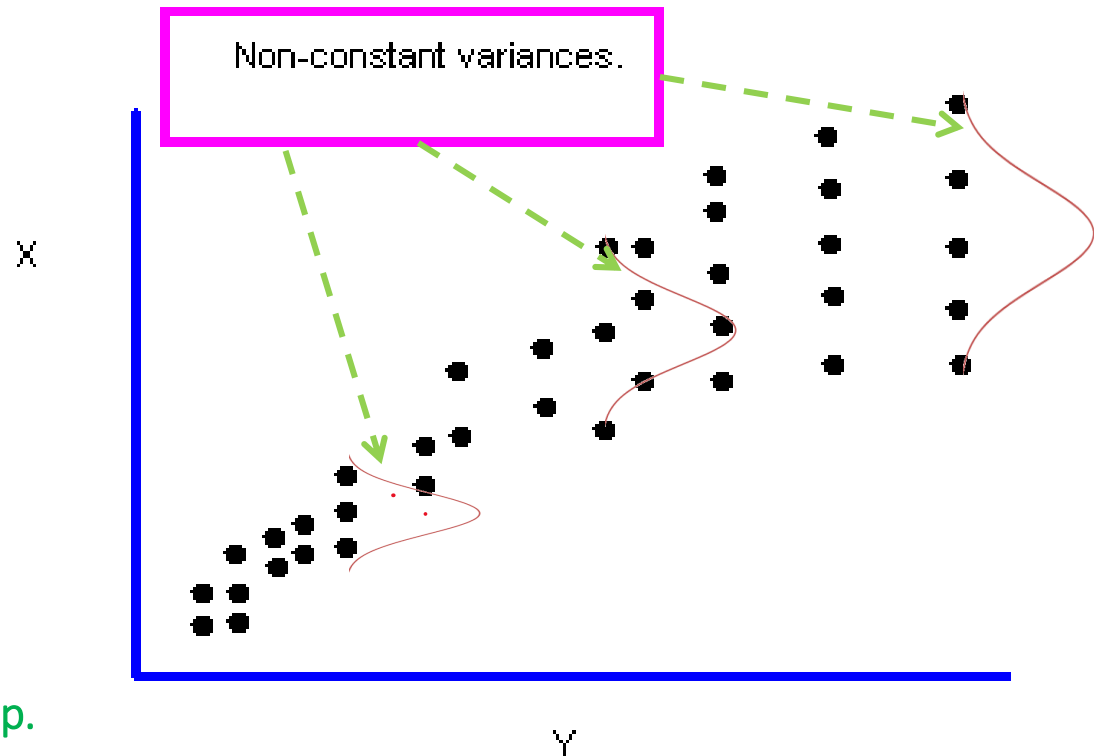
1. *to stabilize the variance*. If there is trend in the series and the variance appears to increase with the mean, then it may be advisable to transform the data. In particular, if the standard deviation is directly proportional to the mean, a logarithmic transformation is appropriate.

$$E(y) = \beta_0 + \beta_1 X \quad \uparrow$$

$$\Rightarrow \text{Var}(y) = \sigma^2 \quad \uparrow$$

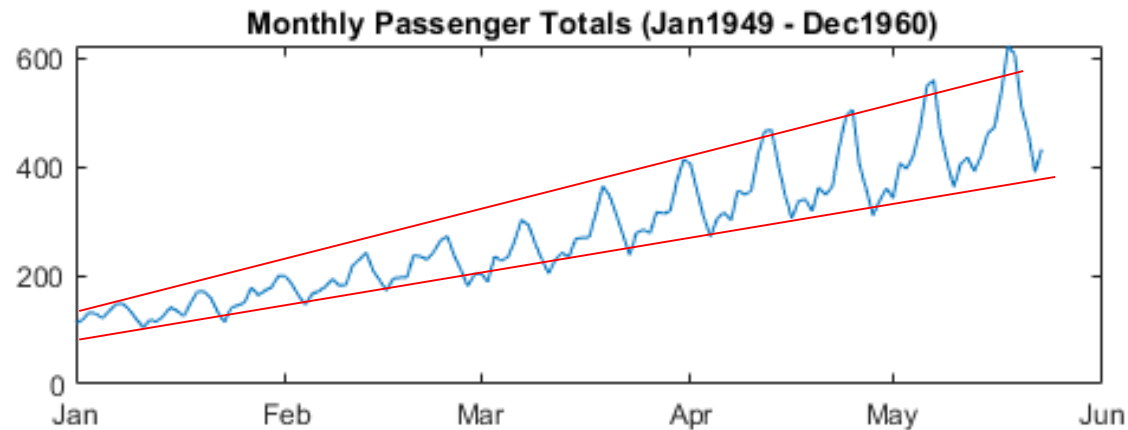
Note:

If variance change through time without a trend, transformation may not help.

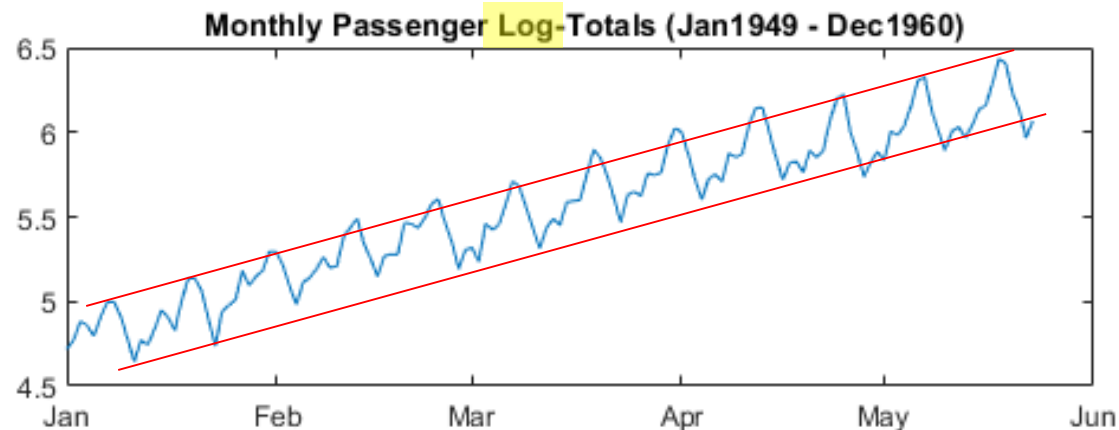


2. *to make the seasonal effect additive.* If there is a trend in the series and the size of the seasonal effect appears to increase with the mean, then it may be advisable to transform the data so as to make the seasonal effect constant. In particular if the size of the seasonal effect is directly proportional to the mean, then the seasonal effect is said to be multiplicative and a logarithmic transformation is appropriate to make the effect additive. However, this transformation will only stabilize the variance if the error term is also thought to be multiplicative, a point which is sometimes overlooked.

Multiplicative



Additive



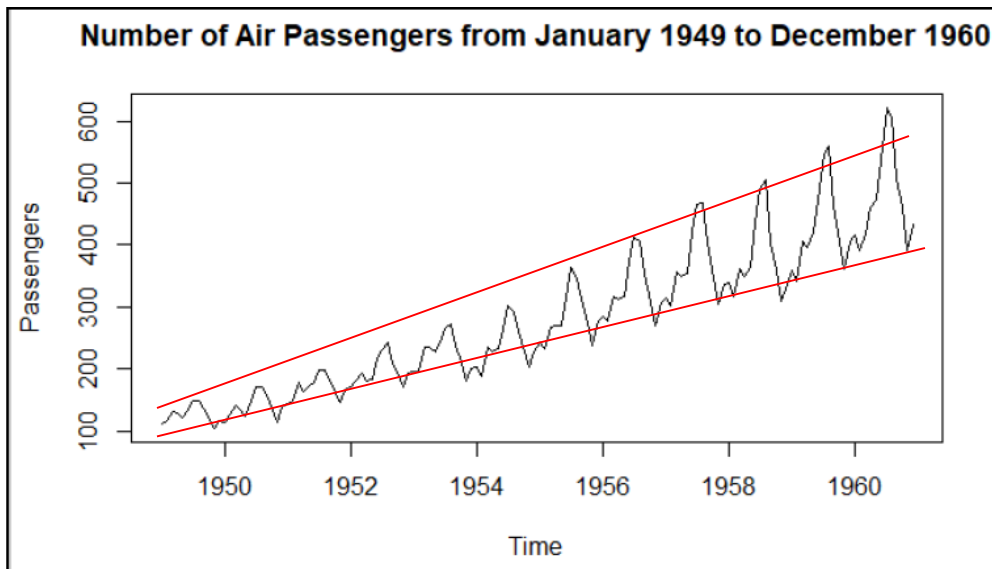


## Example 5.2

Consider the following number of air passengers from Jan 1949 to Apr 1960, stabilize the variance of the time series by using the log transform.

R-codes:

```
Y <- c(112,118,132,129,121,135,148,148,136,119,104,118,  
      115,126,141,135,125,149,170,170,158,133,114,140,  
      145,150,178,163,172,178,199,199,184,162,146,166,  
      171,180,193,181,183,218,230,242,209,191,172,194,  
      196,196,236,235,229,243,264,272,237,211,180,201,  
      204,188,235,227,234,264,302,293,259,229,203,229,  
      242,233,267,269,270,315,364,347,312,274,237,278,  
      284,277,317,313,318,374,413,405,355,306,271,306,  
      315,301,356,348,355,422,465,467,404,347,305,336,  
      340,318,362,348,363,435,491,505,404,359,310,337,  
      360,342,406,396,420,472,548,559,463,407,362,405,  
      417,391,419,461,472,536,622,606,508,461,390,432)  
Y <- ts(Y, frequency = 12, start = c(1949,1))  
plot(Y, ylab="Passengers", main="Number of Air Passengers from January 1949 to December 1960")
```



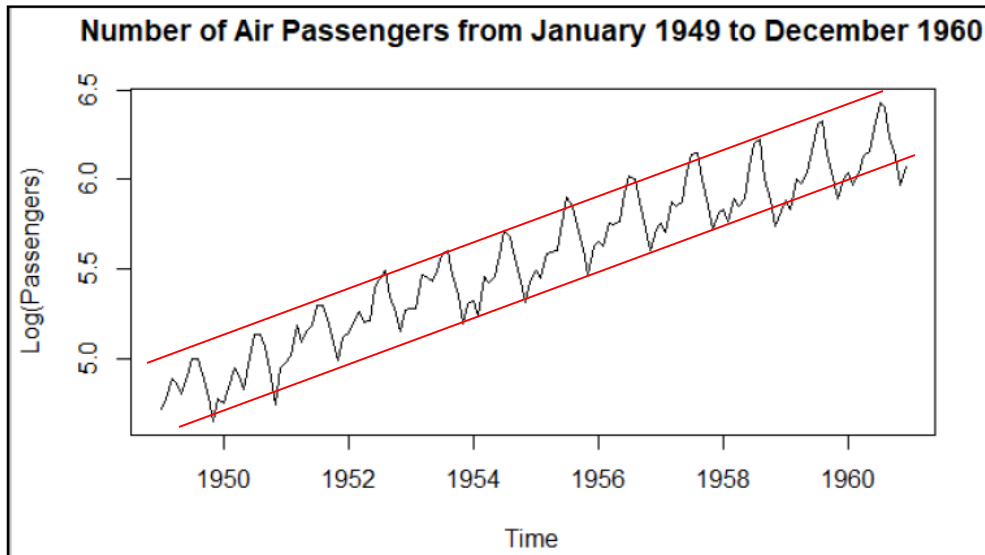
Multiplicative

## R-codes:

```
Y <- log(Y); Y
Y <- ts(Y, frequency = 12, start = c(1949,1))
plot(Y, ylab="Log(Passengers)", main="Number of Air Passengers from January 1949 to December 1960")
```

Output:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1949	4.718499	4.770685	4.882802	4.859812	4.795791	4.905275	4.997212	4.997212	4.912655	4.779123	4.644391	4.770685
1950	4.744932	4.836282	4.948760	4.905275	4.828314	5.003946	5.135798	5.135798	5.062595	4.890349	4.736198	4.941642
1951	4.976734	5.010635	5.181784	5.093750	5.147494	5.181784	5.293305	5.293305	5.214936	5.087596	4.983607	5.111988
1952	5.141664	5.192957	5.262690	5.198497	5.209486	5.384495	5.438079	5.488938	5.342334	5.252273	5.147494	5.267858
1953	5.278115	5.278115	5.463832	5.459586	5.433722	5.493061	5.575949	5.605802	5.468060	5.351858	5.192957	5.303305
1954	5.318120	5.236442	5.459586	5.424950	5.455321	5.575949	5.710427	5.680173	5.556828	5.433722	5.313206	5.433722
1955	5.488938	5.451038	5.587249	5.594711	5.598422	5.752573	5.897154	5.849325	5.743003	5.613128	5.468060	5.627621
1956	5.648974	5.624018	5.758902	5.746203	5.762051	5.924256	6.023448	6.003887	5.872118	5.723585	5.602119	5.723585
1957	5.752573	5.707110	5.874931	5.852202	5.872118	6.045005	6.142037	6.146329	6.001415	5.849325	5.720312	5.817111
1958	5.828946	5.762051	5.891644	5.852202	5.894403	6.075346	6.196444	6.224558	6.001415	5.883322	5.736572	5.820083
1959	5.886104	5.834811	6.006353	5.981414	6.040255	6.156979	6.306275	6.326149	6.137727	6.008813	5.891644	6.003887
1960	6.033086	5.968708	6.037871	6.133398	6.156979	6.284134	6.432940	6.406880	6.230481	6.133398	5.966147	6.068426



Additive

Note:

Transformation of data must be use with caution.

- There are problems in practice with transformations in that a transformation, which makes the seasonal effect additive, may fail to stabilize the variance. Thus, it may be impossible to achieve all the above requirement at the same time.
- It is more difficult to interpret and forecasts produced by the transformed model may have to be “transformed back” in order to be of use. This can introduce biasing effects.
- Transforming the data is encourage if doing so makes physical sense. For example, the percentage data is transformed using a log transform.

### **5.3     Standard Time Series Models**

The main idea of constructing time series models is to study how various factors contribute to the ultimate formation of individual values of the time series. There are different types of time series models. However, the most common types are: (a) **the additive model** and (b) **the multiplicative model**.

## Time Series Additive Model

The time series additive model assumes that the value of time series  $Y$  at a particular time point is the algebraic sum of the trend ( $T$ ), cyclic variation ( $C$ ), seasonal variation ( $S$ ) and random factor of irregular variation ( $I$ ).

$$Y = T + C + S + I$$

The additive models assume that the various components of a time series are independent of one another.

In a simple time series analysis, the overall variation of a time series  $Y$  is treated as the algebraic sum of the trend ( $T$ ), seasonal variation ( $S$ ) and residual variation ( $R$ ).

$$Y = T + S + R$$

The residual variation ( $R$ ) is defined as any variation other than trend or seasonal variation. Therefore, the cyclic movement and the irregular variation are treated as residual variation.

Time Series Multiplicative Model

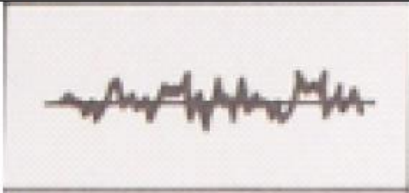
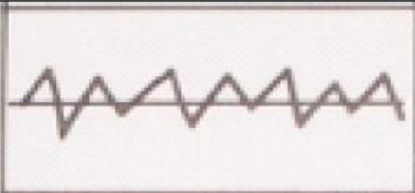



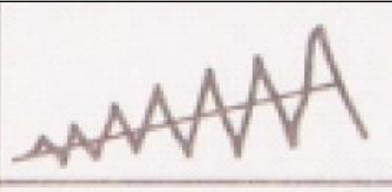



The time series multiplicative model assumes that the value of time series  $Y$  is the product of the trend ( $T$ ), the cyclic variation ( $C$ ), the seasonal variation ( $S$ ) and the irregular variation ( $I$ ).

$$Y = T \times C \times S \times I$$

The time series multiplicative model implies that the various components of a time series are **not** independent of one another. The multiplicative model assumes that seasonal variation and residual factors are *proportional to* the trend. In a simple time series treatment, the cyclic movement and the irregular variation are treated as residual variation.

$$Y = T \times S \times R$$

**Figure 5.1:** Pegels' Classification

	No Seasonal Effect 1	Additive Seasonal 2	Multiplicative Seasonal 3
No Trend Effect A			
Additive Trend B			
Multiplicative Trend C			

## 5.4 Analysing series which contain a trend

The analysis of a time series which exhibits ‘long term’ change in mean depends on whether one wants to (a) measure the trend and/or (b) remove the trend in order to analyse local fluctuations. With seasonal data, it is a good idea to start by calculating successive yearly averages as these will provide a simple description of the underlying trend. An approach of this type is sometimes perfectly adequate, particularly if the trend is fairly small, but sometimes a more sophisticated approach is desired and then the following techniques can be considered.

### 5.4.1 Curve fitting (Regression)

A traditional method of dealing with non-seasonal data that contain a trend, particularly yearly data, is to fit a simple function of time such as

1. Polynomial curve (linear, quadratic, cubic etc.)

$$y_t = \alpha + \beta t \quad \rightarrow \quad y_t = 0.4 + 2t$$

2. Gompertz curve

$$\log y_t = a + br^t \quad \rightarrow \quad \log y_t = 3 + 2 \cdot 0.5^t$$

where  $a, b, r$  are parameters with  $0 < r < 1$ .

3. Logistic curve

$$y_t = \frac{a}{1 + be^{-ct}} \quad \rightarrow \quad y_t = \frac{0.7}{1 + 0.3e^{-2t}}$$



1. Polynomial curve (linear, quadratic, cubic etc.)

$$y_t = \alpha + \beta t \quad \rightarrow \quad y_t = 0.4 + 2t$$

Simple linear regression

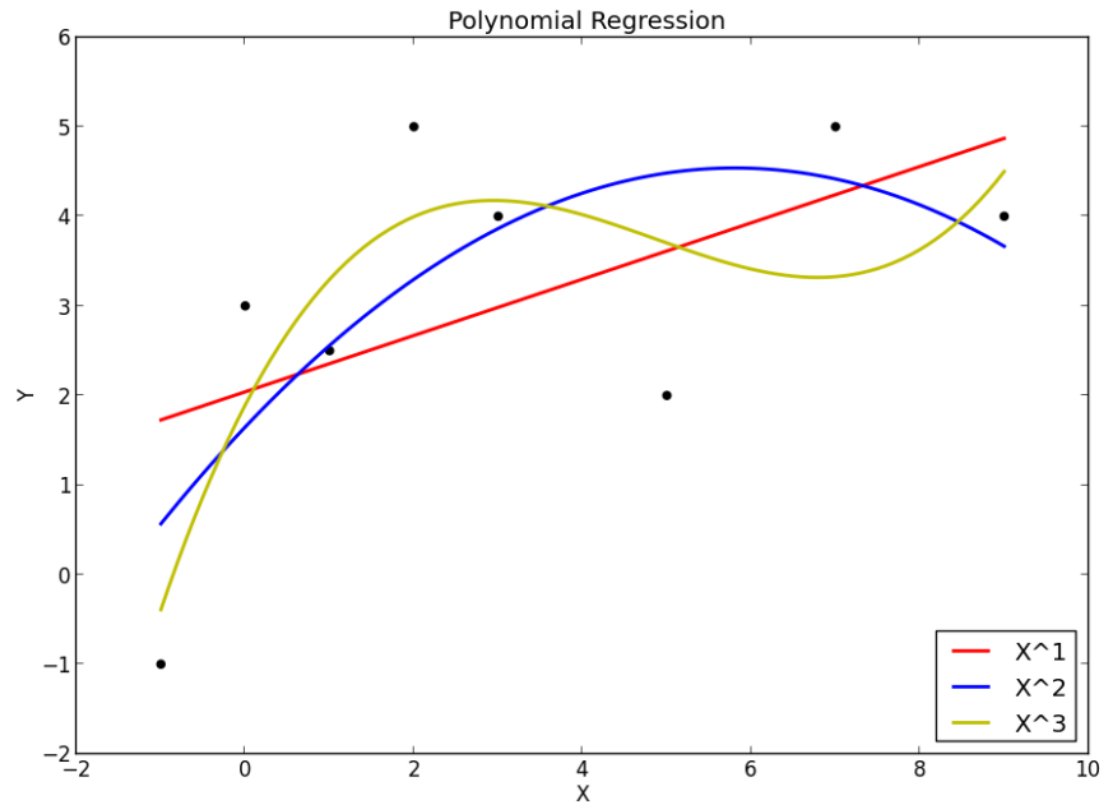
$$y = \beta_0 + \beta_1 X$$

Multiple linear regression

$$y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

Polynomial linear regression

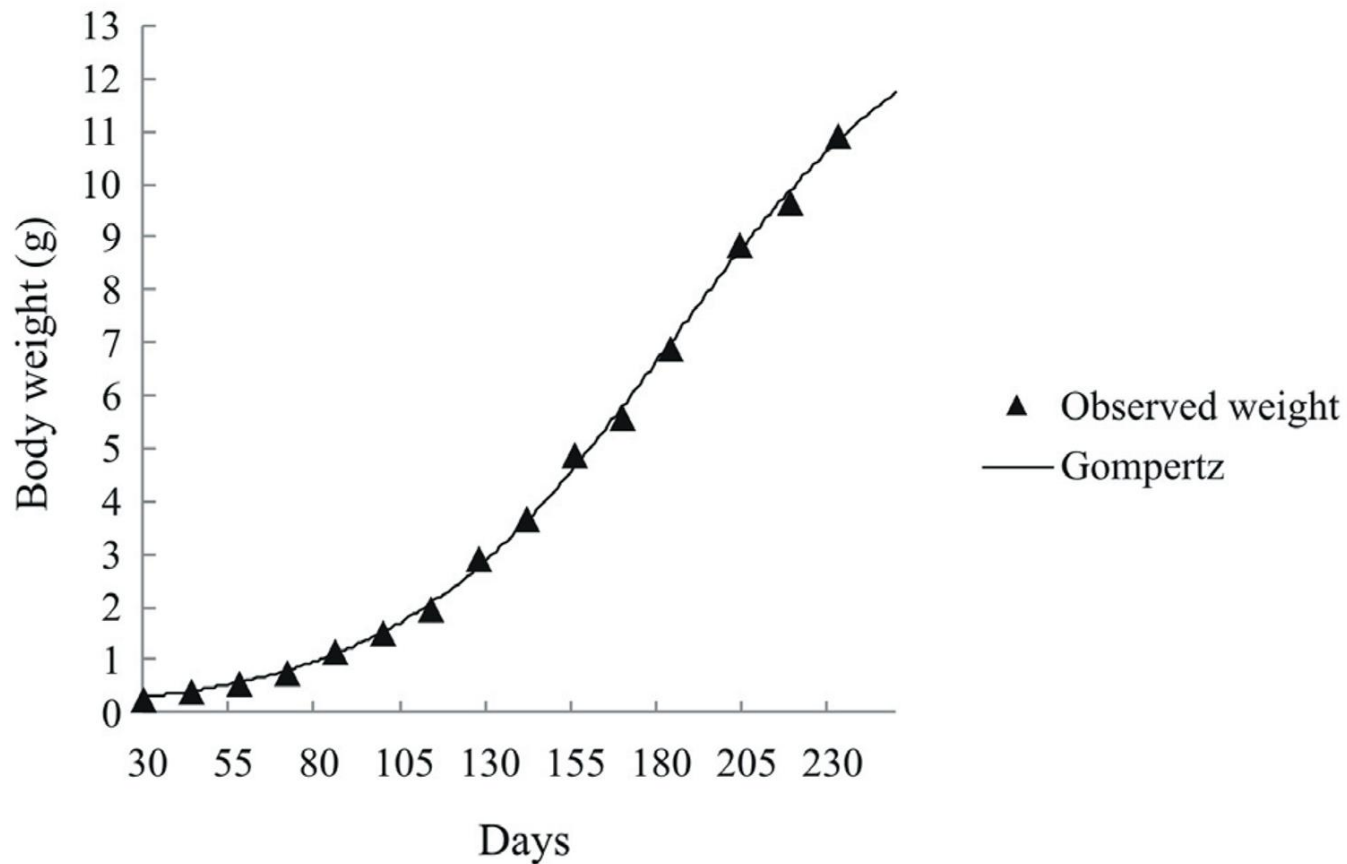
$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_k X^k$$



## 2. Gompertz curve

$$\log y_t = a + br^t \quad \rightarrow \quad \log y_t = 3 + 2 \cdot 0.5^t$$

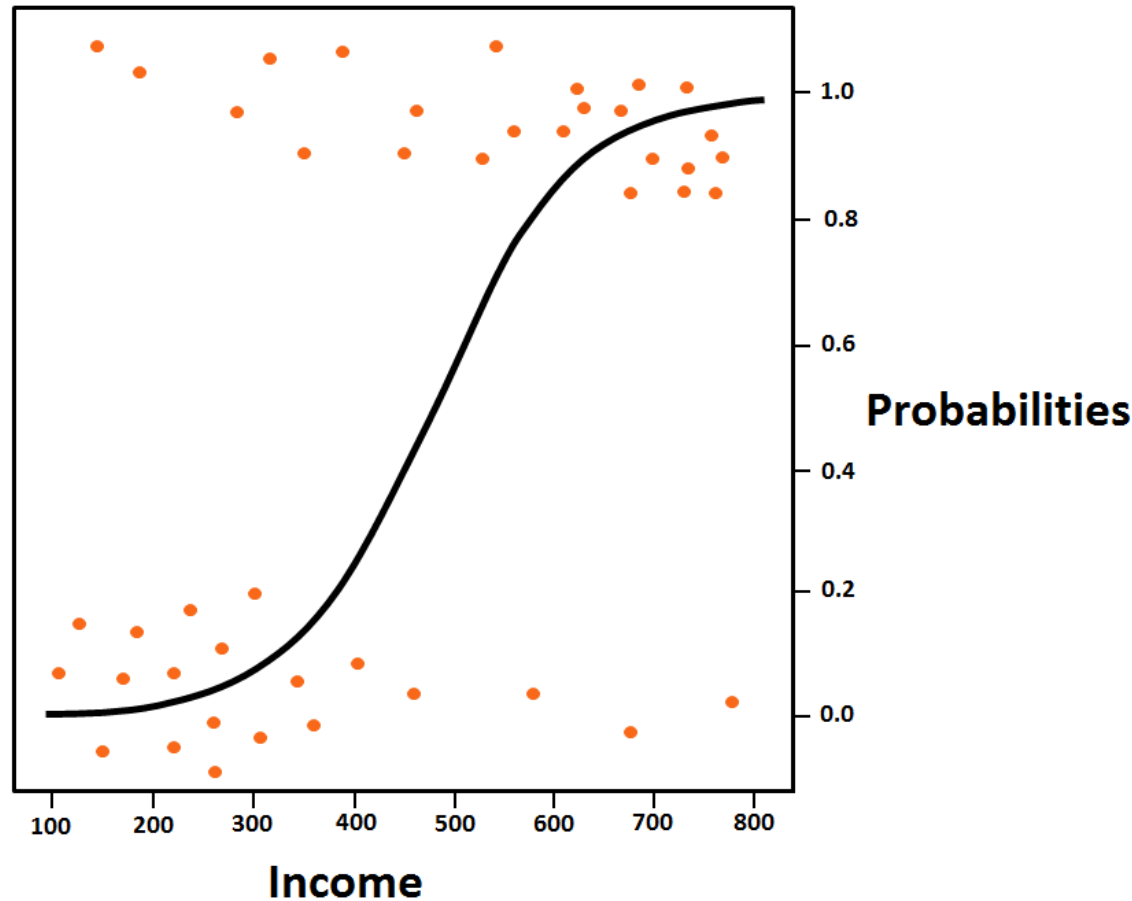
where  $a, b, r$  are parameters with  $0 < r < 1$ .





### 3. Logistic curve

$$y_t = \frac{a}{1 + be^{-ct}} \quad \rightarrow \quad y_t = \frac{0.7}{1 + 0.3e^{-2t}}$$



For all curves of this type, the fitted function provides a measure of the trend, and the residuals provide an estimate of local fluctuations, where the residuals are the differences between the observations and the corresponding values of the fitted curve.

### Example 5.3

Consider the following number of energy consumption (in quardrillion BTUs), fit the energy consumption dataset with a trend line.

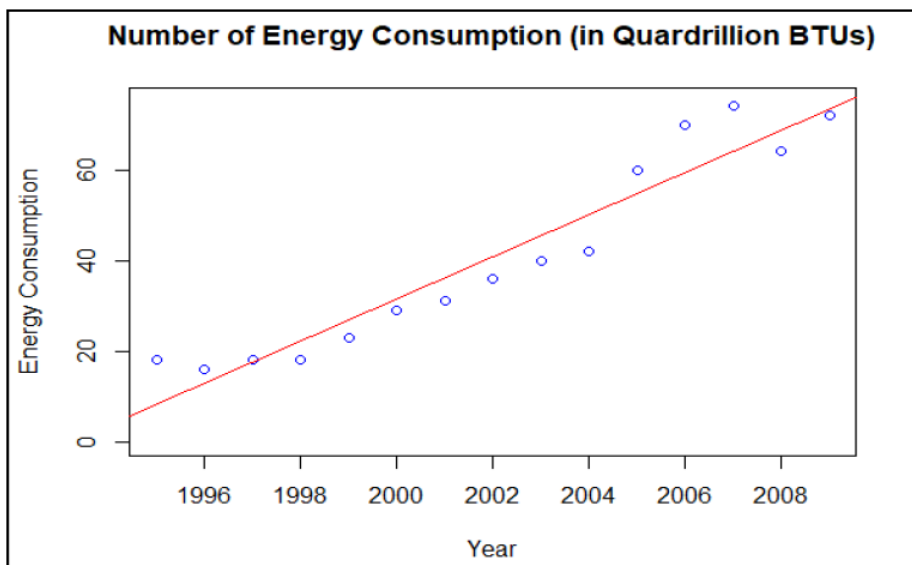
Year	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Period, $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Energy, $Y$	18	16	18	18	23	29	31	36	40	42	60	70	74	64	72

$$\sum t = 120, \sum Y = 611, \sum t^2 = 1240, \sum Y^2 = 31475, \sum Yt = 6188$$

$$\hat{b} = \frac{n\sum Yt - (\sum t)(\sum Y)}{n\sum t^2 - (\sum t)^2} = \frac{15(6188) - 120(611)}{15(1240) - 120^2} = 4.6429$$

$$\hat{a} = \bar{Y} - \hat{b}\bar{t} = \frac{611}{15} - 4.6429\left(\frac{120}{15}\right) = 3.59$$

$$\hat{y}_t = \hat{a} + \hat{b}t = 3.59 + 4.6429t$$



```
y <-  
c(18,16,18,18,23,29,31,36,40,42,60,70,74,64,72)  
t <- 1995:(1995+length(Y)-1)  
plot(t,y, ylim=c(0,75), col="blue",  
      xlab="Year",ylab="Energy Consumption",  
      main="Number of Energy Consumption (in  
            Quardrillion BTUs)")  
trend <- lm(y~t); trend  
abline(trend, col="red")
```

Output:

Call:

```
lm(formula = y ~ t)
```

Coefficients:

```
(Intercept)          t  
      3.590         4.643
```

### 5.4.2 Filtering (Moving Average)

A second procedure for dealing with a trend is to use a *linear filter*,  $a_r$  which converts one time series,  $\{y_t\}$ , into another,  $\{x_t\}$ , by the linear operation

$$x_t = \sum_{r=-q}^{+s} a_r y_{t+r}$$

where  $\{a_r\}$  is a set of weights with  $\sum a_r = 1$ , and this operation is often referred to as a moving average. Moving averages are often symmetric with  $s = q$  and  $a_j = a_{-j}$ .

The simplest example of a symmetric smoothing filter is the simple moving average (SMA), for which  $a_r = \frac{1}{2q+1}$  for  $r = -q, \dots, +q$ , and the smoothed value of  $x_t$  is given by

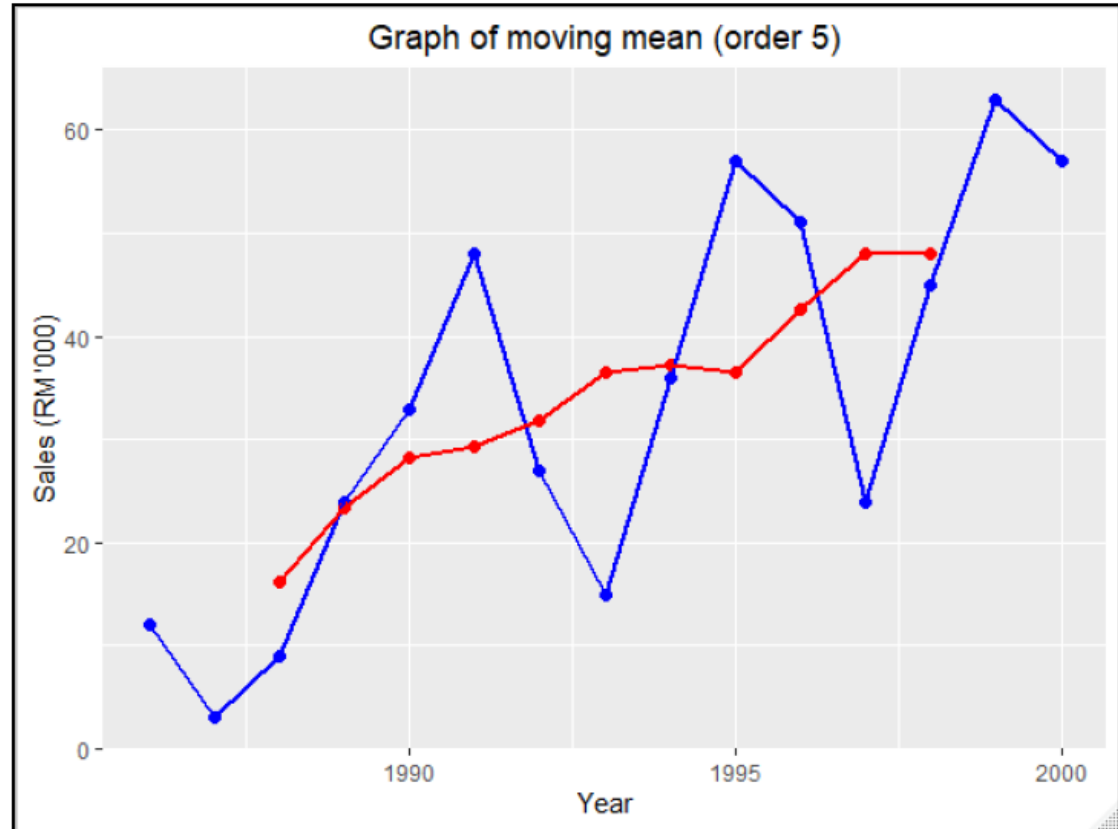
$$x_t = \text{Sm}(y_t) = \frac{1}{2q+1} \sum_{r=-q}^{+q} y_{t+r}$$

Moving average is use for smoothing a series of arithmetic means over time. The result is dependent upon choice of  $l = 2q + 1$  (length of period for computing means).

### Example 5.4

By referring to the Exp 5.1, compute a simple moving average of order 5 and then plot it on the time series graph.

Year	Sales (RM'000)	$SMA(5)$
1986	12	
1987	3	
1988	9	
1989	24	
1990	33	
1991	48	
1992	27	
1993	15	
1994	36	
1995	57	
1996	51	
1997	24	
1998	45	
1999	63	
2000	57	



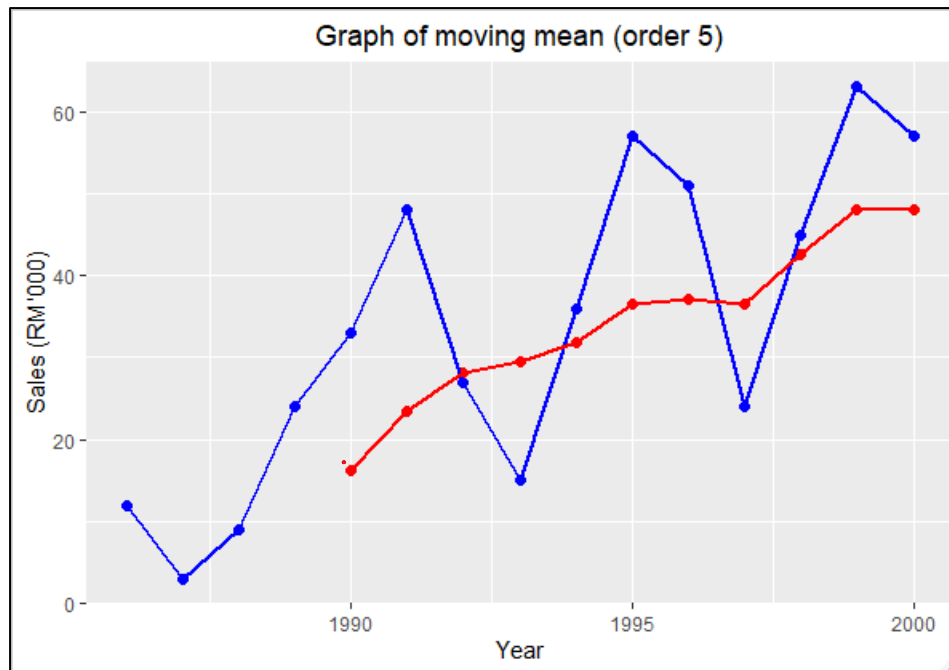
From the smoothed graph the upward trend of increasing sales can be clearly seen.

Note: The simple moving average is useful for removing seasonal variation, but it is not recommended by itself for measuring trend.

```

library(ggplot2)
Y <- c(12,3,9,24,33,48,27,15,36,57,51,24,45,63,57)
t <- 1986:(1986+length(Y)-1)
sma <- function(x,n) {filter(x,rep(1/n,n), sides = 2L)}
X <- sma(Y,5); X
df <- data.frame(t,Y,X)
ggplot(df, aes(x=t)) +
  geom_point(aes(y=Y), colour="blue", size = 2) +
  geom_point(aes(y=X), colour="red", size = 2) +
  geom_line(aes(y=Y), colour="blue", size = 1) +
  geom_line(aes(y=X), colour="red", size = 1) +
  ggtitle("Graph of moving mean (order 5)") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Year") + ylab("Sales (RM'000)")

```



### 5.4.3 Differencing

Trends, or other non-stationary patterns in the level of a series, result in positive autocorrelations that dominate the autocorrelation diagram. Therefore, it is important to **remove the non-stationarity**, so other correlation structure can be seen before proceeding with time series model building.

If a plot of  $n$  time series values indicates that these values are non-stationary, we can sometimes transform the non-stationary time series values into stationary time series values by taking the *first difference* of the non-stationary time series values.

The first difference of the time series values  $Y_1, Y_2, \dots, Y_n$  are

$$Y'_t = Y_t - Y_{t-1} \quad \text{for } t = 2, 3, \dots, n.$$

The differenced series will have only  $n - 1$  values.

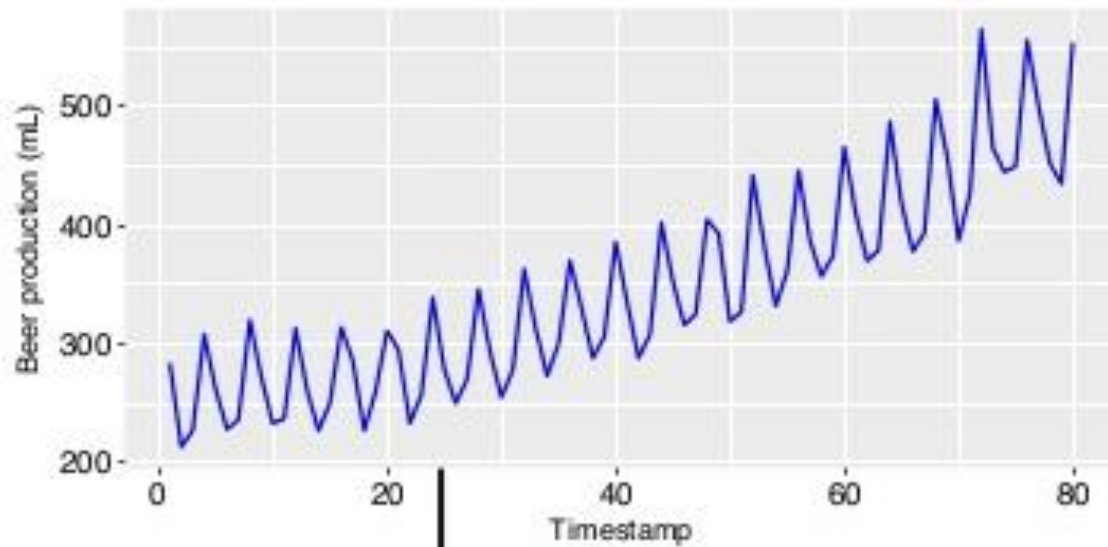
However, if first differences do not transform the time series values into stationary time series, then we can take the second differences (the first differences of the first difference) of the original time series values.

The second difference of the time series values  $\{Y_1, \dots, Y_n\}$  are

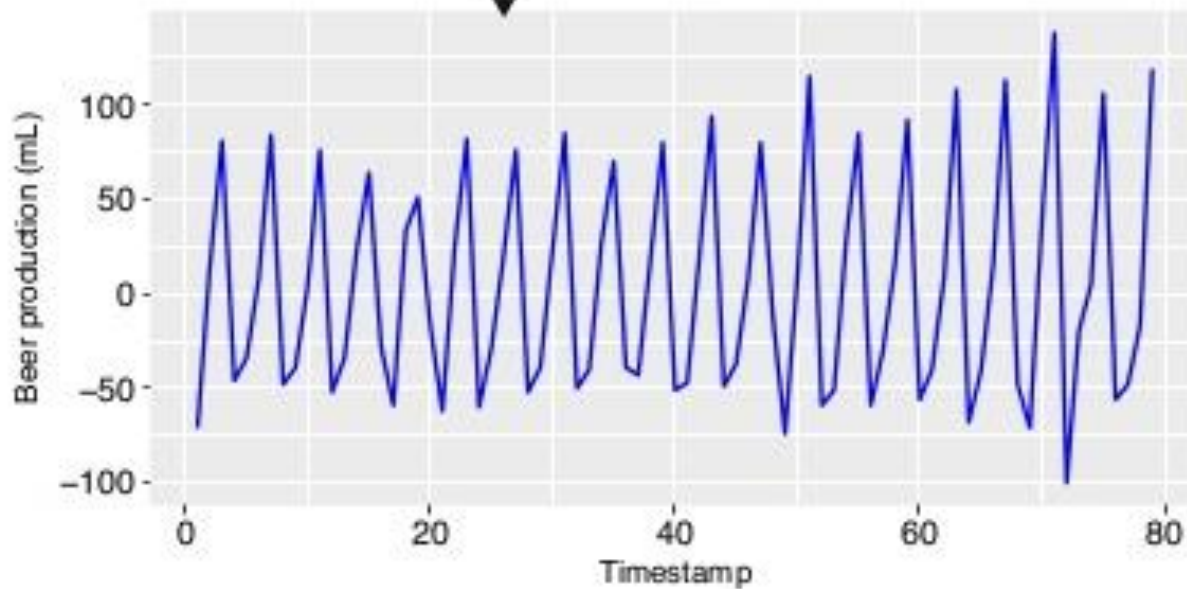
$$\begin{aligned} Y''_t &= Y'_t - Y'_{t-1} \\ &= (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) \\ &= Y_t - 2Y_{t-1} + Y_{t-2} \quad \text{for } t = 3, 4, \dots, n. \end{aligned}$$

This differenced series will have only  $n - 2$  values.

Normally, if the original time series values are non-stationary and non-seasonal, then by using first difference or second differences will produce stationary time series values (white noise series).



$$Y_t = X_t - X_{t-1}$$



Differencing order (d)  
means the number of  
times differencing is done



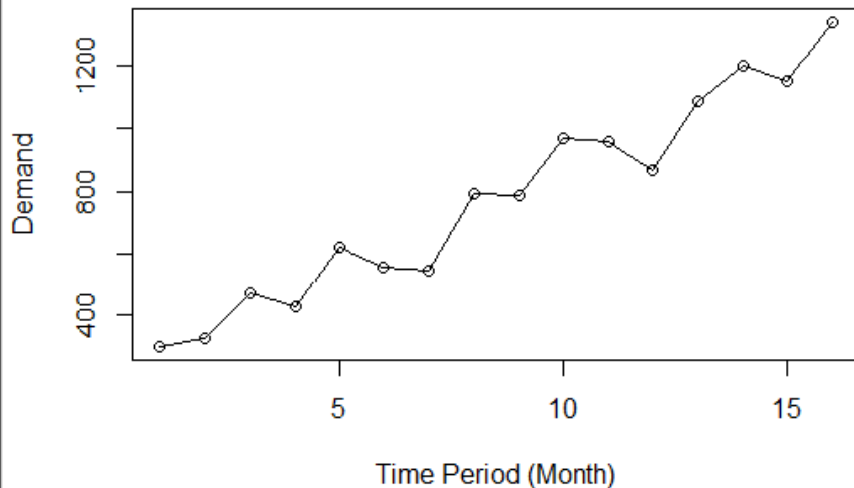
### Example 5.5

Consider the following demand from Jan '04 to Apr '05, calculate the first difference and second order difference of the dataset, then plot the first difference.

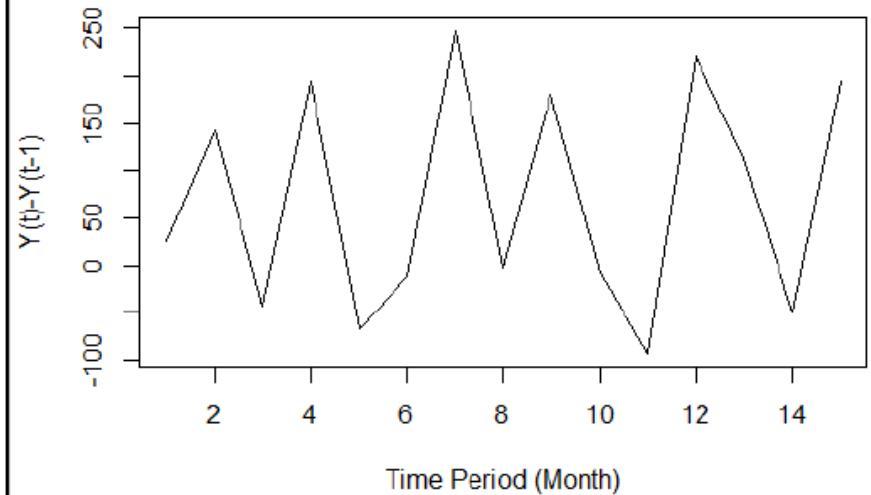
Year	Month	Period	Demand, $Y_t$	First Difference, $Y'_t$	Second Order Difference, $Y''_t$
2004	Jan	1	300		
	Feb	2	325		
	Mar	3	468		
	Apr	4	426		
	May	5	621		
	Jun	6	554		
	Jul	7	543	-11	56
	Aug	8	790	247	258
	Sept	9	787	-3	-250
	Oct	10	968	181	184
	Nov	11	960	-8	-189
	Dec	12	867	-93	-85
2005	Jan	13	1087	220	313
	Feb	14	1200	113	-107
	Mar	15	1151	-49	-162
	Apr	16	1345	194	243



**Demand from Jan '04 to Apr '05**



**First Difference**



## R-codes:

```
Y <- c(300,325,468,426,621,554,543,790,787,968,960,867,1087,1200,1151,1345)
Y <- ts(Y)
plot(Y, type="o", xlab="Time Period (Month)", ylab="Demand", main="Demand from Jan '04
to Apr '05")

Y <- c(300,325,468,426,621,554,543,790,787,968,960,867,1087,1200,1151,1345)
Y <- ts(diff(Y))
plot(Y, xlab="Time Period (Month)", ylab="Y(t)-Y(t-1)", main="First Difference")
```

The first differences transform the demand dataset into a stationary time series.

## 5.5 Analysing series which contain seasonal variation

The analysis of time series which exhibit seasonal fluctuations depends on whether one wants to (a) measure the seasonal and/or (b) eliminate them. In order to examine the statistical properties of the time series dataset, we would like to extract the residuals  $\hat{\varepsilon}_t$ . The following methods allow for estimation of the trend and the seasonal components:

1. Small trend method
2. Smoothing average for monthly, quarterly data...
3. Seasonal differencing

### 5.5.1 Small trend method

For the time series that has a small trend, it is usually adequate to simply calculate the average for each month (or quarter, or 4-week period etc.) and compare it with the corresponding yearly average figure, either as a difference (additive) or as a ratio (multiplicative).

We assume the trend within each period is constant, and due to the assumptions of additive model, the period average is an unbiased estimator of the trend, that is

$$\hat{T}_j = \frac{1}{d} \sum_{k=1}^d Y_{jk} .$$

The seasonal component estimator, which satisfies the model assumptions is

$$\hat{S}_k = \frac{1}{b} \sum_{j=1}^b (Y_{jk} - \hat{T}_j) .$$

Removing the estimates of trend and seasonality from the time series, we obtain the residuals

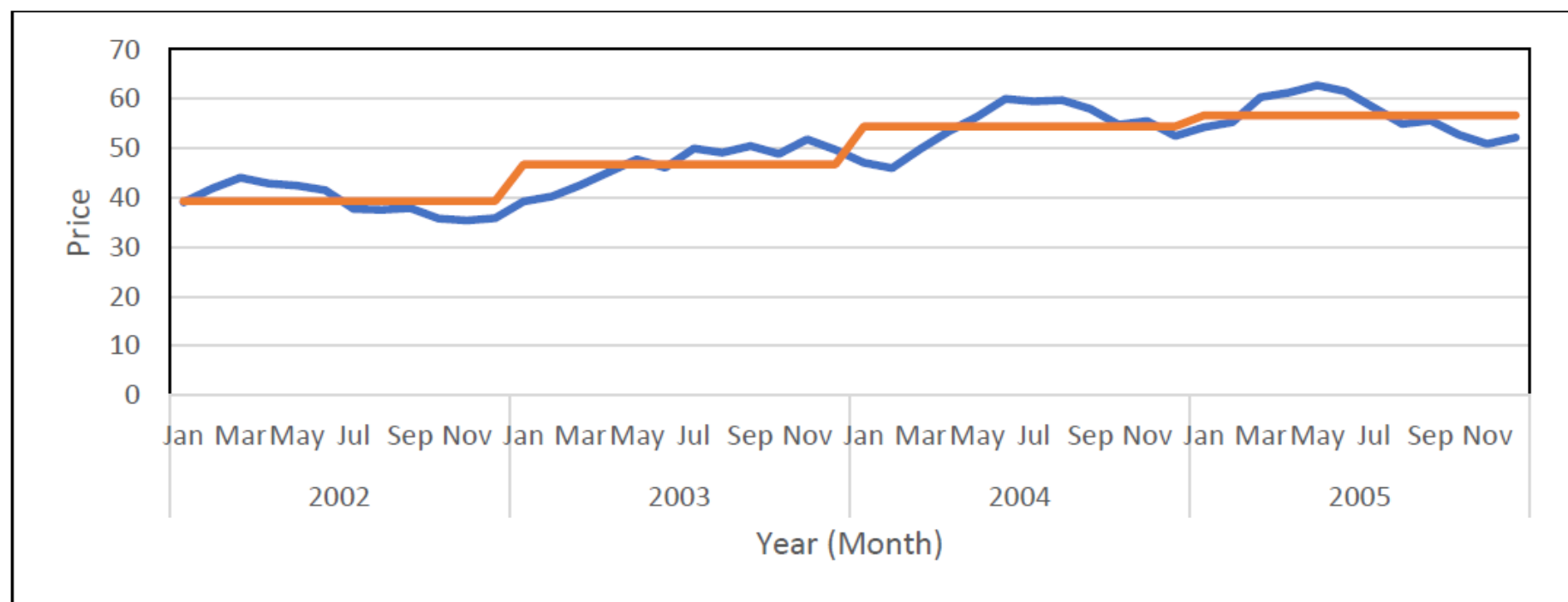
$$\hat{\varepsilon}_{jk} = Y_{jk} - \hat{T}_j - \hat{S}_k \quad \text{for } j = 1, \dots, b, \quad k = 1, \dots, d.$$

### Example 5.6

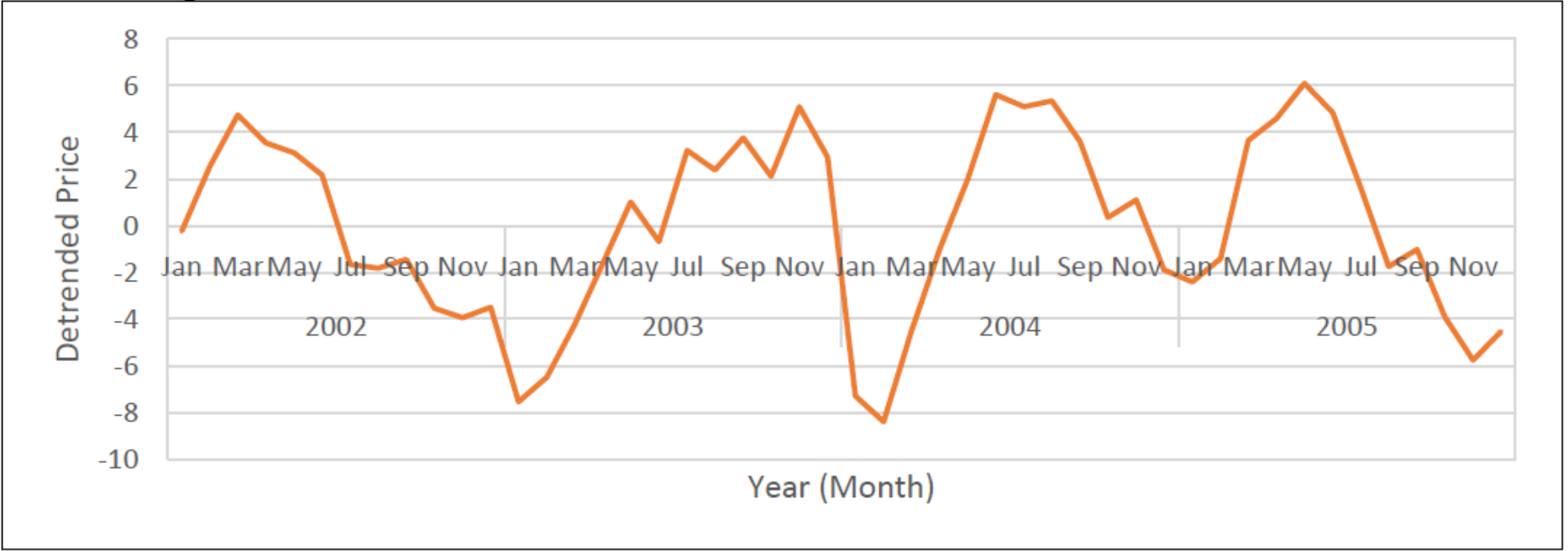
Consider the following Sioux falls cut cow prices monthly data from 2002 to 2005.

Price and the estimated trend

$Y_{jk}$	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec	$\hat{T}_j$
2002	39.1	41.88	44.06	42.88	42.45	41.5	37.67	37.5	37.88	35.8	35.38	35.83	
2003	39.2	40.25	42.5	45.13	47.75	46.06	49.96	49.13	50.5	48.85	51.83	49.67	
2004	47.1	46	49.88	53.4	56.38	60	59.5	59.75	58	54.75	55.5	52.5	
2005	54.25	55.25	60.3	61.25	62.75	61.5	58.25	54.9	55.63	52.75	50.9	52.13	

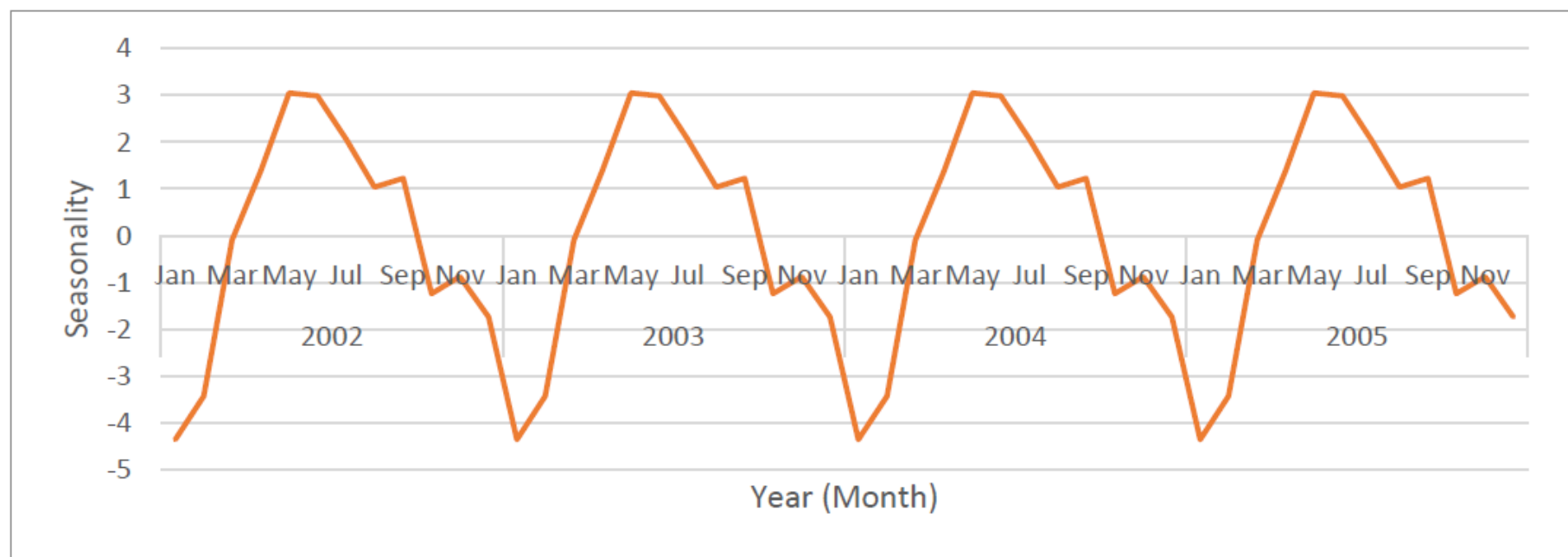


Detrended price

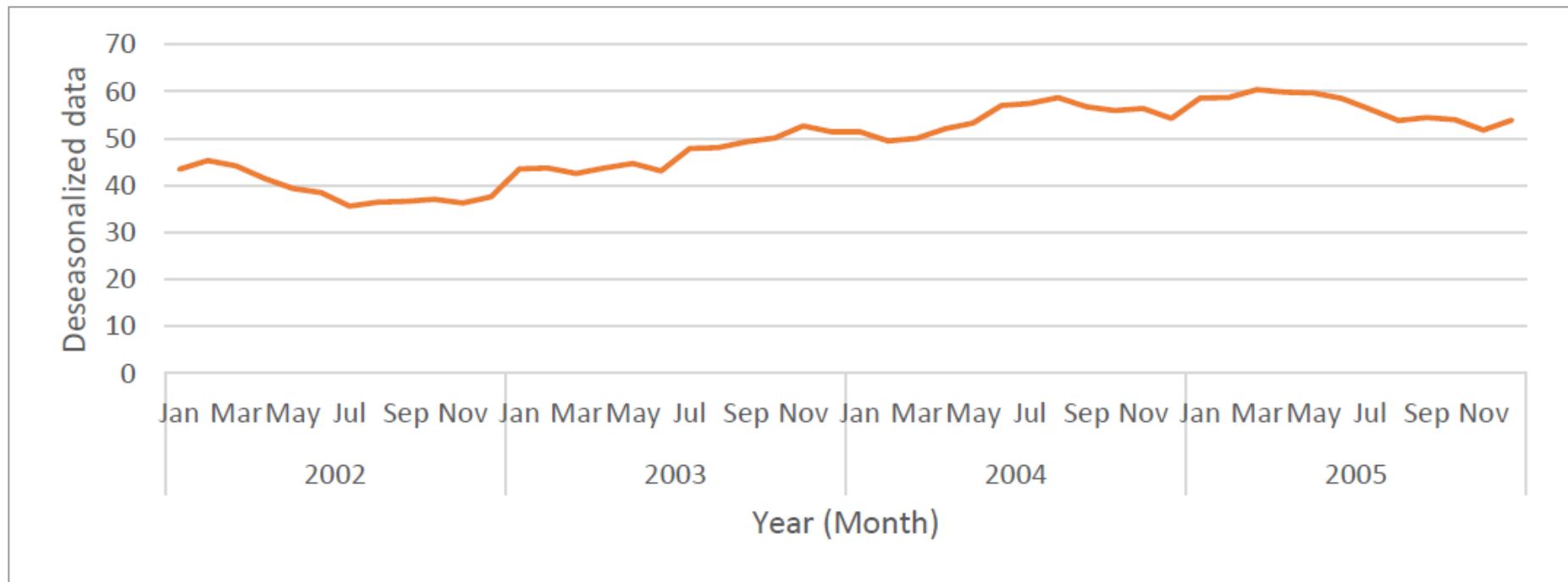


Estimated seasonal component

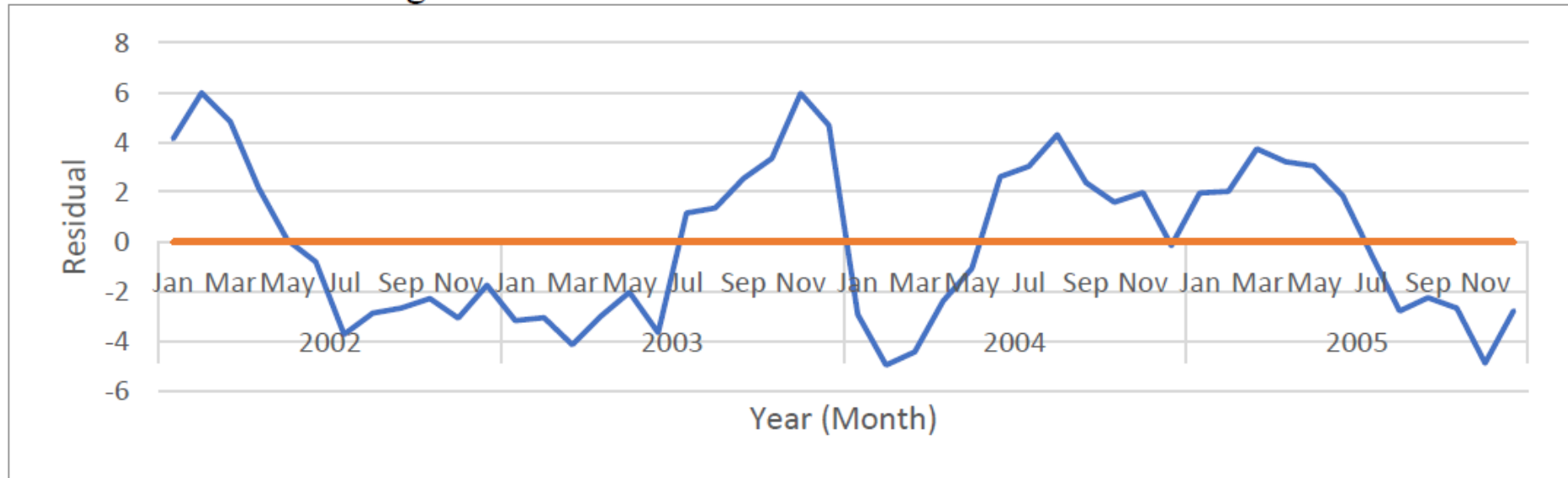
$Y_{jk} - \hat{T}_j$	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2002	-0.2275	2.5525	4.7325	3.5525	3.1225	2.1725	-1.6575	-1.8275	-1.4475	-3.5275	-3.9475	-3.4975
2003	-7.5358	-6.4858	-4.2358	-1.6058	1.0142	-0.6758	3.2242	2.3942	3.7642	2.1142	5.0942	2.9342
2004	-7.2967	-8.3967	-4.5167	-0.9967	1.9833	5.6033	5.1033	5.3533	3.6033	0.3533	1.1033	-1.8967
2005	-2.405	-1.405	3.645	4.595	6.095	4.845	1.595	-1.755	-1.025	-3.905	-5.755	-4.525
$\hat{S}_k$												



## Deseasonalization



## Residuals after removing trend and seasonal effects



Year	Month	Price, $Y_{jk}$	Trend, $\hat{T}_j$	Detrended, $Y_{jk} - \hat{T}_j$	Seasonal Component, $\hat{S}_k$	<u>Deseasonalized, <math>Y_{jk} - \hat{S}_k</math></u>	Residual, $\hat{\varepsilon}_{jk}$
2002	Jan	39.1					
	Feb	41.88					
	Mar	44.06					
	Apr	42.88					
	May	42.45					
	Jun	41.5	39.3275	2.1725	2.98625	38.51375	-0.81375
	Jul	37.67	39.3275	-1.6575	2.06625	35.60375	-3.72375
	Aug	37.5	39.3275	-1.8275	1.04125	36.45875	-2.86875
	Sep	37.88	39.3275	-1.4475	1.22375	36.65625	-2.67125
	Oct	35.8	39.3275	-3.5275	-1.24125	37.04125	-2.28625
	Nov	35.38	39.3275	-3.9475	-0.87625	36.25625	-3.07125
	Dec	35.83	39.3275	-3.4975	-1.74625	37.57625	-1.75125
2003	Jan	39.2	46.7358	-7.53583	-4.36625	43.56625	-3.16958
	Feb	40.25	46.7358	-6.48583	-3.43375	43.68375	-3.05208
	Mar	42.5	46.7358	-4.23583	-0.09375	42.59375	-4.14208
	Apr	45.13	46.7358	-1.60583	1.38625	43.74375	-2.99208
	May	47.75	46.7358	1.014167	3.05375	44.69625	-2.03958
	Jun	46.06	46.7358	-0.67583	2.98625	43.07375	-3.66208
	Jul	49.96	46.7358	3.224167	2.06625	47.89375	1.157917
	Aug	49.13	46.7358	2.394167	1.04125	48.08875	1.352917
	Sep	50.5	46.7358	3.764167	1.22375	49.27625	2.540417
	Oct	48.85	46.7358	2.114167	-1.24125	50.09125	3.355417
	Nov	51.83	46.7358	5.094167	-0.87625	52.70625	5.970417
	Dec	49.67	46.7358	2.934167	-1.74625	51.41625	4.680417

2004	Jan	47.1	54.3967	-7.29667	-4.36625	51.46625	-2.93042
	Feb	46	54.3967	-8.39667	-3.43375	49.43375	-4.96292
	Mar	49.88	54.3967	-4.51667	-0.09375	49.97375	-4.42292
	Apr	53.4	54.3967	-0.99667	1.38625	52.01375	-2.38292
	May	56.38	54.3967	1.983333	3.05375	53.32625	-1.07042
	Jun	60	54.3967	5.603333	2.98625	57.01375	2.617083
	Jul	59.5	54.3967	5.103333	2.06625	57.43375	3.037083
	Aug	59.75	54.3967	5.353333	1.04125	58.70875	4.312083
	Sep	58	54.3967	3.603333	1.22375	56.77625	2.379583
	Oct	54.75	54.3967	0.353333	-1.24125	55.99125	1.594583
	Nov	55.5	54.3967	1.103333	-0.87625	56.37625	1.979583
	Dec	52.5	54.3967	-1.89667	-1.74625	54.24625	-0.15042
2005	Jan	54.25	56.655	-2.405	-4.36625	58.61625	1.96125
	Feb	55.25	56.655	-1.405	-3.43375	58.68375	2.02875
	Mar	60.3	56.655	3.645	-0.09375	60.39375	3.73875
	Apr	61.25	56.655	4.595	1.38625	59.86375	3.20875
	May	62.75	56.655	6.095	3.05375	59.69625	3.04125
	Jun	61.5	56.655	4.845	2.98625	58.51375	1.85875
	Jul	58.25	56.655	1.595	2.06625	56.18375	-0.47125
	Aug	54.9	56.655	-1.755	1.04125	53.85875	-2.79625
	Sep	55.63	56.655	-1.025	1.22375	54.40625	-2.24875
	Oct	52.75	56.655	-3.905	-1.24125	53.99125	-2.66375
	Nov	50.9	56.655	-5.755	-0.87625	51.77625	-4.87875
	Dec	52.13	56.655	-4.525	-1.74625	53.87625	-2.77875



### 5.5.2 Classical decomposition

Classical decomposition consists of the following steps:

**Step 1** Estimate trend using a moving average filter of the period length  $d$ . If the period  $d$  is odd, then use  $\hat{T}_t = \text{Sm}(Y_t)$  with  $q$  specified by the equation  $d = 2q + 1$ . If the period  $d = 2q$  is even, then slightly modify  $\text{Sm}(Y_t)$  and use

$$\hat{T}_t = \frac{1}{d} \left( \frac{1}{2} y_{t-q} + y_{t-q+1} + \cdots + \frac{1}{2} y_{t+q} \right), \quad t = q + 1, \dots, n - q$$

or

$$\hat{T}_t = \begin{cases} \frac{1}{d} \left( \frac{1}{2} y_{t-q} + y_{t-q+1} + \cdots + \frac{1}{2} y_{t+q} \right) & \text{for } d = 2q, \quad q + 1 < t < n - q \\ \frac{1}{d} (y_{t-q} + y_{t-q+1} + \cdots + y_{t+q}) & \text{for } d = 2q + 1, \quad q + 1 < t < n - q \end{cases}$$

**Step 2** Estimate seasonal effects  $S_k$  for  $k = 1, \dots, d$ :

compute the averages of the detrended values and adjust them (normalized the seasonal indices so that they are sum to zero in the additive case, or average to one in the multiplicative case) so as the seasonal effects meet the model assumptions, that is estimate the seasonal component  $S_k$  as

$$\hat{S}_k = \begin{cases} \overline{(Y_l - \hat{T}_l)_k} - \frac{1}{d} \sum_{i=1}^d \overline{(Y_l - \hat{T}_l)_i}, & k = 1, \dots, d, \\ \hat{S}_{k-d}, & k > d. \end{cases} \quad (\text{additive})$$

$$\hat{S}_k = \begin{cases} \overline{\left( \frac{Y_l}{\hat{T}_l} \right)_k} \cdot d \sum_{i=1}^d \overline{\left( \frac{Y_l}{\hat{T}_l} \right)_i}, & k = 1, \dots, d, \\ \hat{S}_{k-d}, & k > d. \end{cases} \quad (\text{multiplicative})$$

**Step 3** Remove the seasonality to obtain

$$D_t = Y_t - \hat{S}_t, \quad t = 1, \dots, n. \quad (\text{additive})$$

$$D_t = Y_t / \hat{S}_t, \quad t = 1, \dots, n. \quad (\text{multiplicative})$$

**Step 4** Re-estimate the trend  $\hat{T}_t$  from the deseasonalized variables  $\{D_t\}$ .

**Step 5** Calculate the residuals

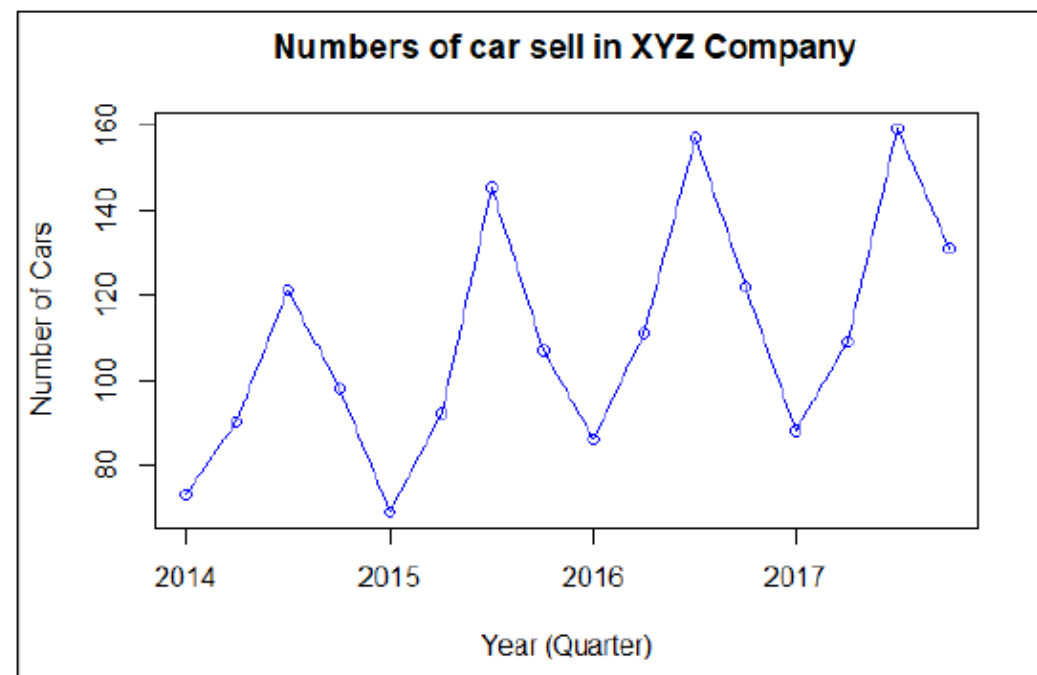
$$\hat{\varepsilon}_t = Y_t - \hat{T}_t - \hat{S}_t. \quad (\text{additive})$$

$$\hat{\varepsilon}_t = \frac{Y_t}{\hat{T}_t \hat{S}_t}. \quad (\text{multiplicative})$$

### Example 5.7: Additive Model

Consider the following numbers of car sell by quarter for a period of 4 years in XYZ company.

Year	Quarter			
	1	2	3	4
2014	73	90	121	98
2015	69	92	145	107
2016	86	111	157	122
2017	88	109	159	131



The time plot above suggests that the numbers of car sell display a linear demand and a constant (additive) seasonal variation. Thus, we can apply the additive model while actual results fluctuate up and down according to the quarter of the year and so a moving average of four will be used.

Year	Quarter	Numbers of car (Y)	4-quarter moving total	4-quarter moving average	4-quarter Centred moving average (Trend, T)	Detrended (Y – T)	Seasonal variation (S)	Deseasonalized (Y – S)	Residual (Y – T – S)
2014	1	73							
	2	90							
	3	121							
	4	98							
2015	1	69							
	2	92							
	3	145							
	4	107							
2016	1	86			113.75	-27.75	-28.5521	114.5521	0.8021
	2	111	461	115.25	117.125	-6.125	-8.51042	119.5104	2.3854
	3	157	476	119	119.25	37.75	35.23958	121.7604	2.5104
	4	122	478	119.5	119.25	2.75	1.822917	120.1771	0.9271
2017	1	88	476	119	119.25	-31.25	-28.5521	116.5521	-2.6979
	2	109	478	119.5	120.625	-11.625	-8.51042	117.5104	-3.1146
	3	159	487	121.75			35.23958	123.7604	
	4	131					1.822917	129.1771	

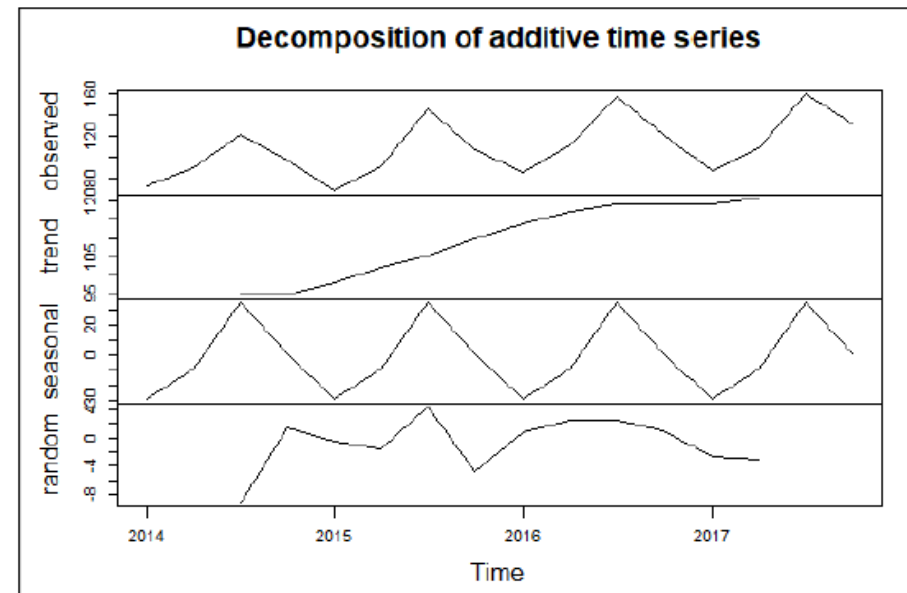
Year	Quarter				
	1	2	3	4	
2014					
2015					
2016					
2017					
Total					
Average SV					
Adjustment					
Adjusted SV					

## R-codes:

```
Y <- c(73,90,121,98,69,92,145,107,86,111,157,122,88,109,159,131)
Y <- ts(Y, frequency = 4, start = 2014)
plot(Y, ylim=c(0,180), type="o", col="Blue", main="Numbers of car sell in
XYZ Company", xlab="Year (Quarter)", ylab="Number of Cars")
components <- decompose(Y)
plot(components)
cbind(components$x,components$trend,components$seasonal,components$random)
```

Output:

	components\$x	components\$trend	components\$seasonal	components\$random
2014 Q1	73	NA	-28.552083	NA
2014 Q2	90	NA	-8.510417	NA
2014 Q3	121	95.000	35.239583	-9.2395833
2014 Q4	98	94.750	1.822917	1.4270833
2015 Q1	69	98.000	-28.552083	-0.4479167
2015 Q2	92	102.125	-8.510417	-1.6145833
2015 Q3	145	105.375	35.239583	4.3854167
2015 Q4	107	109.875	1.822917	-4.6979167
2016 Q1	86	113.750	-28.552083	0.8020833
2016 Q2	111	117.125	-8.510417	2.3854167
2016 Q3	157	119.250	35.239583	2.5104167
2016 Q4	122	119.250	1.822917	0.9270833
2017 Q1	88	119.250	-28.552083	-2.6979167
2017 Q2	109	120.625	-8.510417	-3.1145833
2017 Q3	159	NA	35.239583	NA
2017 Q4	131	NA	1.822917	NA



**Example 5.8: Multiplicative Model**

Year	Day	Numbers of car (Y)	5-day moving total	5-day moving average (Trend, T)	Detrended, $Y \div T$	Seasonal variation (S)	Deseasonalized ( $Y \div S$ )	Residual ( $Y \div T \div S$ )
Week 1	Mon	80						
	Tue	104						
	Wed	94						
	Thurs	120						
	Fri	62						
Week 2	Mon	82	471	94.2	0.8705	0.8638	94.9294	1.0077
	Tue	110	476	95.2	1.1555	1.1636	94.5342	0.9930
	Wed	97	478	95.6	1.0146	1.0138	95.6796	1.0008
	Thurs	125	480	96	1.3021	1.2991	96.2205	1.0023
	Fri	64	486	97.2	0.6584	0.6597	97.0138	0.9981
Week 3	Mon	84	489	97.8	0.8589	0.8638	97.2447	0.9943
	Tue	116	494	98.8	1.1741	1.1636	99.6906	1.0090
	Wed	100	496	99.2	1.0081	1.0138	98.6388	0.9943
	Thurs	130				1.2991	100.0693	
	Fri	66				0.6597	100.0455	

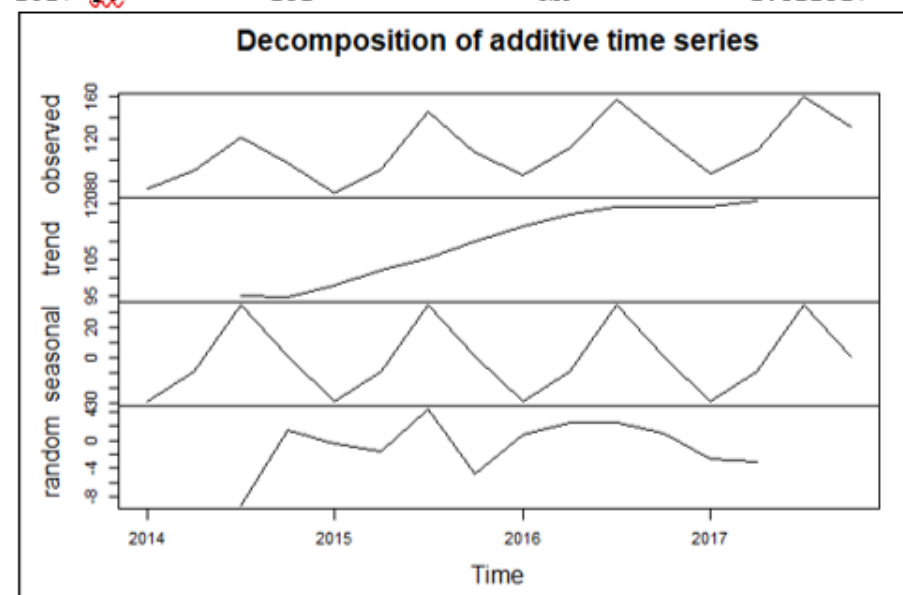
Week	Day					
	Mon	Tue	Wed	Thurs	Fri	
1						
2						
3						
Total						
Average SV						
Adjustment						
Adjusted SV						

## R-codes:

```
Y <- c(73,90,121,98,69,92,145,107,86,111,157,122,88,109,159,131)
Y <- ts(Y, frequency = 4, start = 2014)
plot(Y, ylim=c(0,180), type="o", col="Blue", main="Numbers of car sell in
XYZ Company", xlab="Year (Quarter)", ylab="Number of Cars")
components <- decompose(Y)
plot(components)
cbind(components$x,components$trend,components$seasonal,components$random)
```

Output:

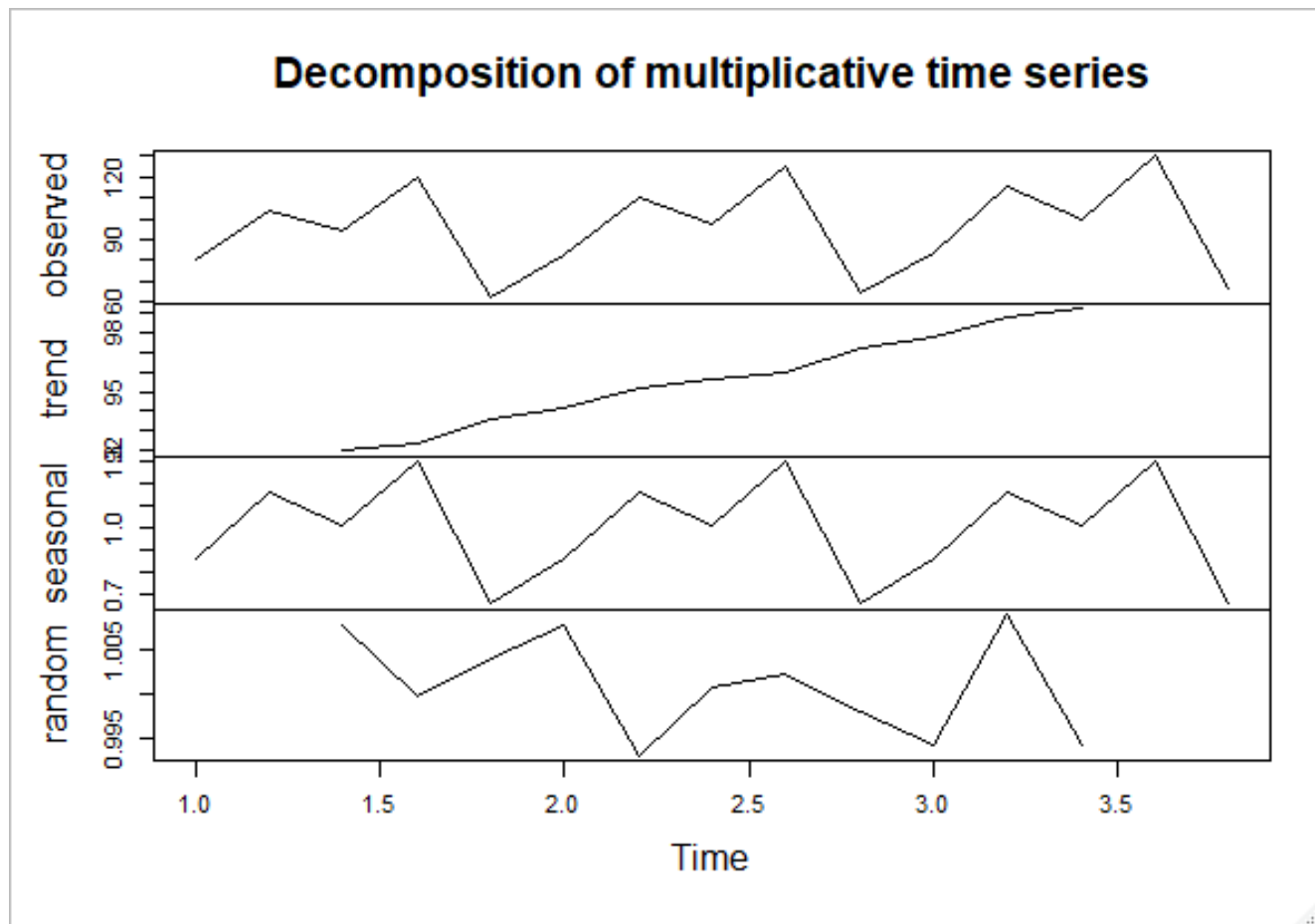
	<u>components\$x</u>	<u>components\$trend</u>	<u>components\$seasonal</u>	<u>components\$random</u>
2014 Q1	73	NA	-28.552083	NA
2014 Q2	90	NA	-8.510417	NA
2014 Q3	121	95.000	35.239583	-9.2395833
2014 Q4	98	94.750	1.822917	1.4270833
2015 Q1	69	98.000	-28.552083	-0.4479167
2015 Q2	92	102.125	-8.510417	-1.6145833
2015 Q3	145	105.375	35.239583	4.3854167
2015 Q4	107	109.875	1.822917	-4.6979167
2016 Q1	86	113.750	-28.552083	0.8020833
2016 Q2	111	117.125	-8.510417	2.3854167
2016 Q3	157	119.250	35.239583	2.5104167
2016 Q4	122	119.250	1.822917	0.9270833
2017 Q1	88	119.250	-28.552083	-2.6979167
2017 Q2	109	120.625	-8.510417	-3.1145833
2017 Q3	159	NA	35.239583	NA
2017 Q4	131	NA	1.822917	NA



```

Y <- c(80,104,94,120,62,82,110,97,125,64,84,116,100,130,66)
Y <- ts(Y, frequency = 5, start = 1)
plot(Y, type="o", col="Blue", main="Numbers of car sell in XYZ
Company", xlab="Week", ylab="Number of Cars")
components <- decompose(Y, type="multiplicative")
plot(components)
cbind(components$x, components$trend, components$seasonal, components$
random)

```



Note: The length of the moving average **must be equal to the seasonal frequency** (for monthly series, we would take  $l = 12$ ). This process of average the two moving averages is called **centering**. Thus, the trend at time  $t$  can be estimated by the centered moving average.

### 5.5.3 Seasonal differencing

For seasonal data which is non-stationary, it may be appropriate to take seasonal difference. A seasonal difference is the difference between an observation and the corresponding observation from the previous year. The differences (from the previous year) may be about the same for each month of the year giving us a stationary series. So, for monthly data having an annual 12-month pattern, we let  $Y'_t = Y_t - Y_{t-12}$ .

Let  $d$  be the span of the periodic seasonal behaviour, a seasonal differencing at lag  $d$  is given by

$$Y'_t = Y_t - Y_{t-d}.$$



### Example 5.9

Reconsider Exp 5.5, calculate the seasonal differencing at lag 12 for the sample demand dataset.

Year	Month	Period	Demand	lag(1)	lag(2)	lag(12)	Seasonal differencing, $Y'_t$
2004	Jan	1	300				
	Feb	2	325				
	Mar	3	468				
	Apr	4	426				
	May	5	621				
	Jun	6	554	621	426		
	Jul	7	543	554	621		
	Aug	8	790	543	554		
	Sept	9	787	790	543		
	Oct	10	968	787	790		
	Nov	11	960	968	787		
	Dec	12	867	960	968		
2005	Jan	13	1087	867	960		
	Feb	14	1200	1087	867		
	Mar	15	1151	1200	1087		
	Apr	16	1345	1151	1200		

## 5.6 The Autocorrelation Function (ACF)

Autocorrelation is the key statistic in time series analysis. It is a correlation coefficient (or the correlation of the time series with itself, lagged by 1, 2 or more periods). While it is usually impossible to obtain a complete description of a stochastic process, the autocorrelation function will be useful because it provides a partial description of the process for modelling purposes.

We define the autocorrelation with lag  $k$  as

$$\rho_k = \frac{E[(Y_t - \mu_y)(Y_{t-k} - \mu_y)]}{\sqrt{E[(Y_t - \mu_y)^2]E[(Y_{t-k} - \mu_y)^2]}} = \frac{Cov(Y_t, Y_{t-k})}{\sigma_{Y_t}\sigma_{Y_{t-k}}}$$

Consider the time series values  $y_1, y_2, \dots, y_n$ . We usually use simpler formula which gives almost the same answers. The sample autocorrelation at lag  $k$ , denoted by  $r_k$ , is

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

Then  $r_1$  indicates how successive values of  $Y$  relate to each other,  $r_2$  indicates how  $Y$  values two periods apart relate to each other, and so on. Together, the autocorrelations at lags 1, 2, ..., make up the autocorrelation function or *ACF*.

In-practice the autocorrelation coefficients are usually calculated by computing the series of autocovariance coefficients,  $\gamma_k$ , which are defined by

$$\gamma_k = \frac{1}{n} \sum_{t=k+1}^n (y_t - \bar{y})(y_{t-k} - \bar{y})$$

Therefore,  $r_k = \frac{\gamma_k}{\gamma_0}$  for  $k = 1, 2, \dots, m$ , where  $m \leq N$ .

Example 5.10

Consider the following two years of monthly sales data (24 observations), calculate the lag 3 autocorrelation coefficient for the sample dataset.

Period	Sales, $y_t$	$y_t - \bar{y}$	$(y_t - \bar{y})^2$	3-Unit Lag	$y_{t-3} - \bar{y}$	$(y_t - \bar{y})(y_{t-3} - \bar{y})$
1	9.08					
2	12.63					
3	15.00					
4	20.73	2.5367	6.4347			
5	2.20	-15.9933	255.7867			
6	18.00	-0.1933	0.0374			
7	7.16	-11.0333	121.7344	20.73	2.5367	-27.9879
8	18.28	0.0867	0.0075	2.2	-15.9933	-1.3861
9	21.00	2.8067	7.8774	18	-0.1933	-0.5426
10	19.68	1.4867	2.2102	7.16	-11.0333	-16.4029
11	15.54	-2.6533	7.0402	18.28	0.0867	-0.2300
12	24.00	5.8067	33.7174	21	2.8067	16.2974
13	16.10	-2.0933	4.3820	19.68	1.4867	-3.1121
14	11.93	-6.2633	39.2293	15.54	-2.6533	16.6187
15	27.00	8.8067	77.5574	24	5.8067	51.1374
16	12.51	-5.6833	32.3003	16.1	-2.0933	11.8971
17	20.04	1.8467	3.4102	11.93	-6.2633	-11.5663
18	30.00	11.8067	139.3974	27	8.8067	103.9774
19	12.41	-5.7833	33.4469	12.51	-5.6833	32.8686
20	14.33	-3.8633	14.9253	20.04	1.8467	-7.1343
21	33.00	14.8067	219.2374	30	11.8067	174.8174
22	22.11	3.9167	15.3403	12.41	-5.7833	-22.6514
23	17.91	-0.2833	0.0803	14.33	-3.8633	1.0946
24	36.00	17.8067	317.0774	33	14.8067	263.6574

This shows that there is 0.45 correlation with every three-time units (months in this case). This could be a sign of seasonality in buyer behavior or some business event happening regularly every three months that influence sales. For example, every three months (March, June, September and December) your company releases a new product which all your customers want to buy right away.

## 5.7 Correlogram

A correlogram, also known as sample autocorrelation function, is a plot of sample autocorrelation coefficients  $r_k$  against lag  $k$  for  $k = 0, 1, \dots, m$ , where  $m$  is usually much less than  $n$ .

### Example 5.11

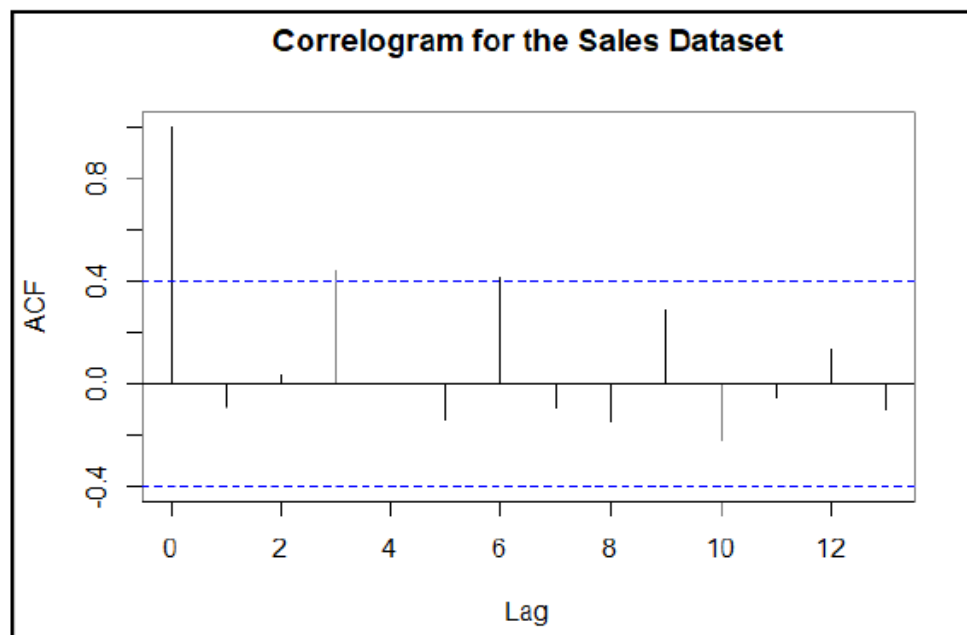
Refer to Exp 5.10

```
Y <- c(9.08,12.63,15,20.73,2.2,18,7.16,18.28,21,19.68,15.54,24,  
       16.1,11.93,27,12.51,20.04,30,12.41,14.33,33,22.11,17.91,36)  
ACF <- acf(Y, main="Correlogram for the Sales Dataset")  
ACF$acf
```

R-codes:

```
Y <- c(300,325,468,426,621,554,543,790,787,968,960,867,1087,1200,1151,1345)  
ACF <- acf(Y, main="Correlogram for the Demand Dataset")  
ACF$acf
```

```
      [,1]  
[1,] 1.000000000  
[2,] -0.092839705  
[3,] 0.038724850  
[4,] 0.445111171  
[5,] -0.003455823  
[6,] -0.138769889  
[7,] 0.407319088  
[8,] -0.099116631  
[9,] -0.149228006  
[10,] 0.281604546  
[11,] -0.223908537  
[12,] -0.053114593  
[13,] 0.133685400  
[14,] -0.102758600
```



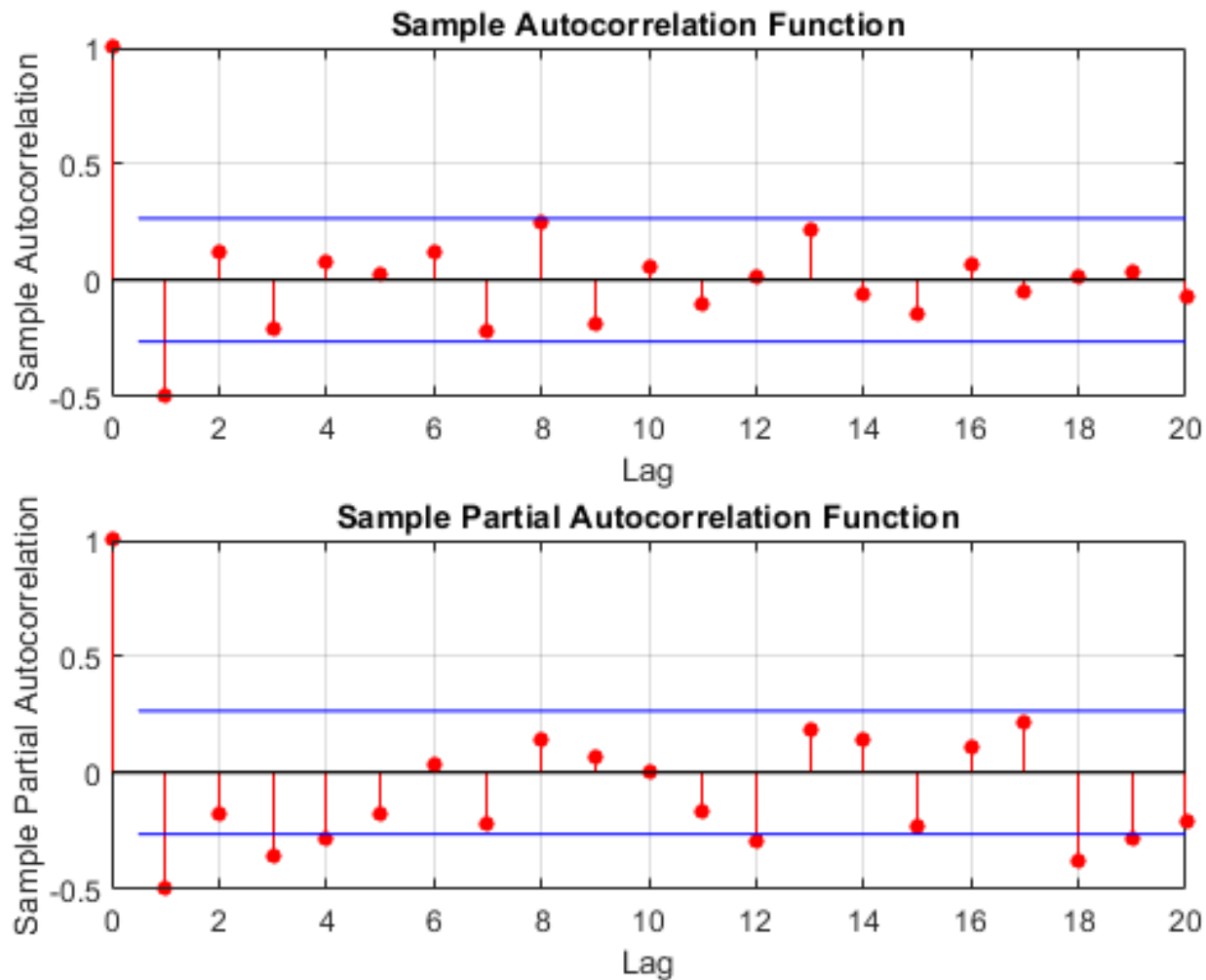
Note: A lag of zero is perfectly correlated since it's putting the exact same data against itself.

Interpreting the meaning of a set of autocorrelation coefficients is not always easy but it can let us check the information about the

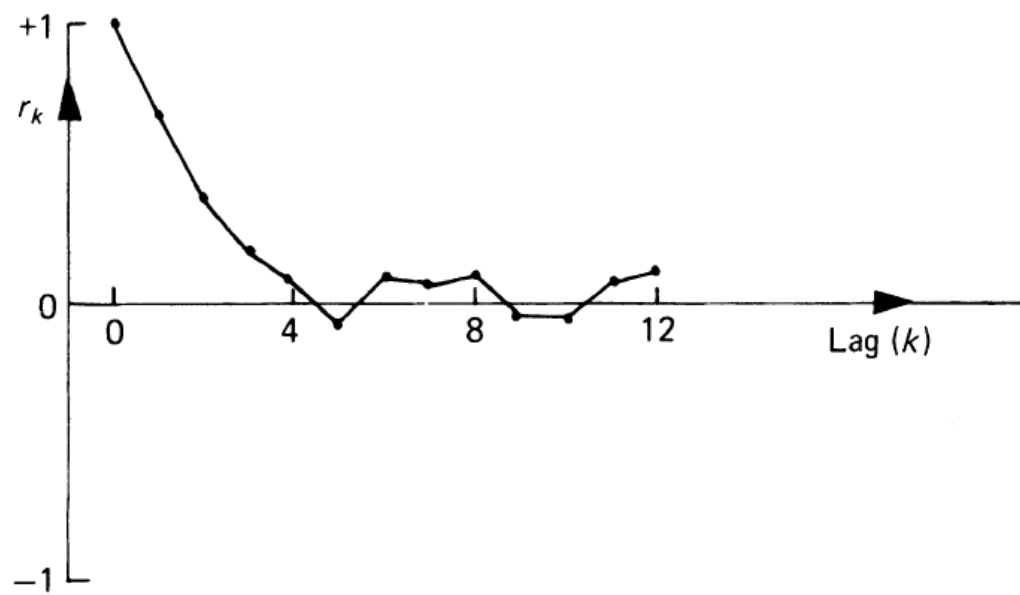
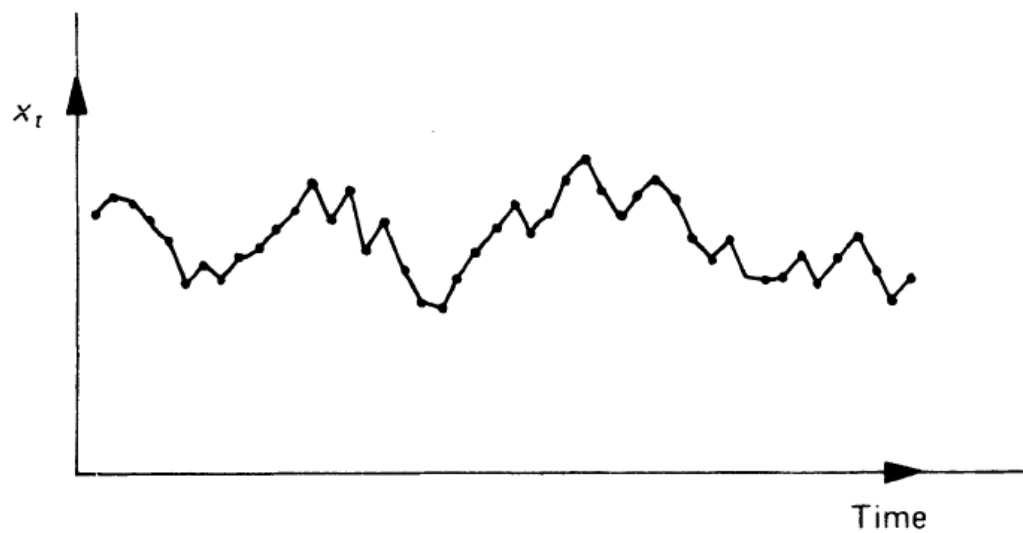
- Random series
- Short-term correlation
- Alternating series
- Non-stationary series
- Seasonal series
- Outliers

Interpreting the meaning of a set of autocorrelation coefficients is not always easy but it can let us check the information about the

- Random series

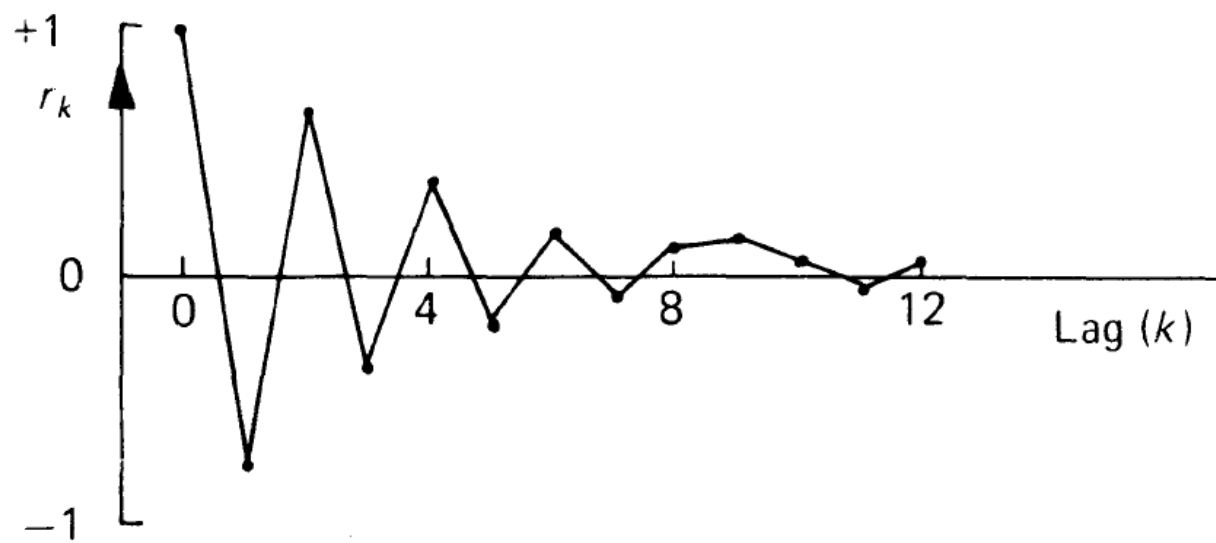
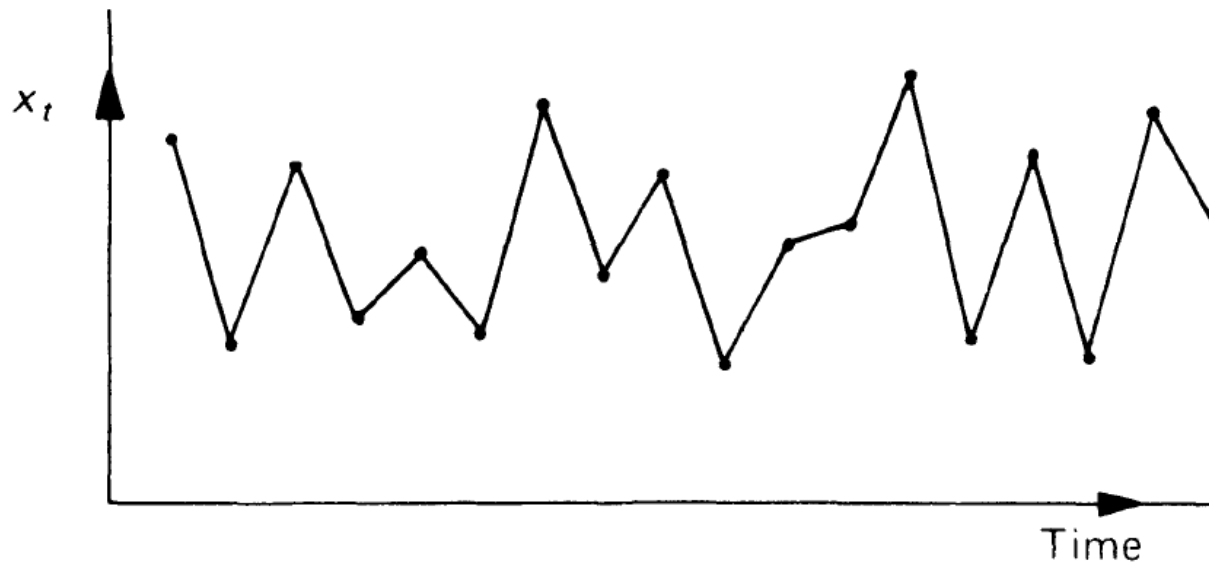


- Short-term correlation

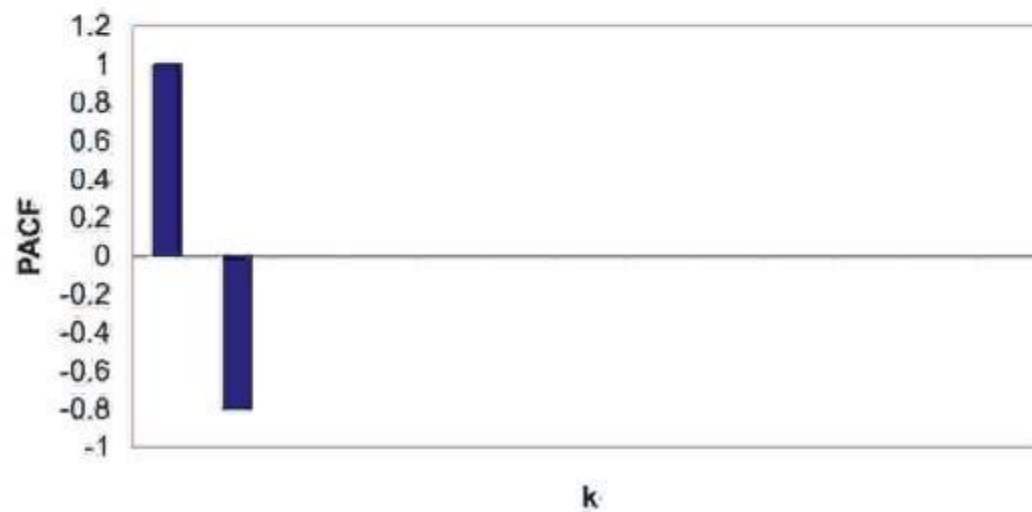
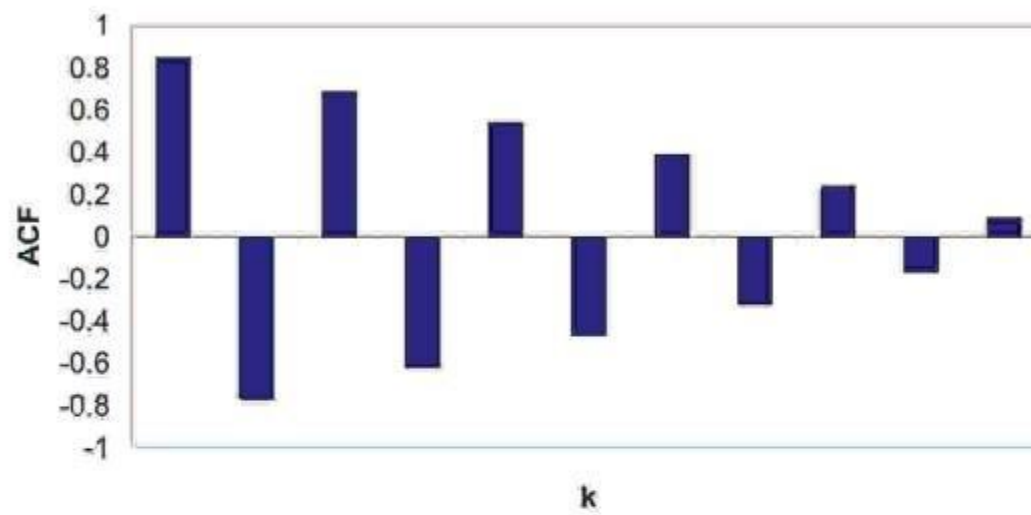




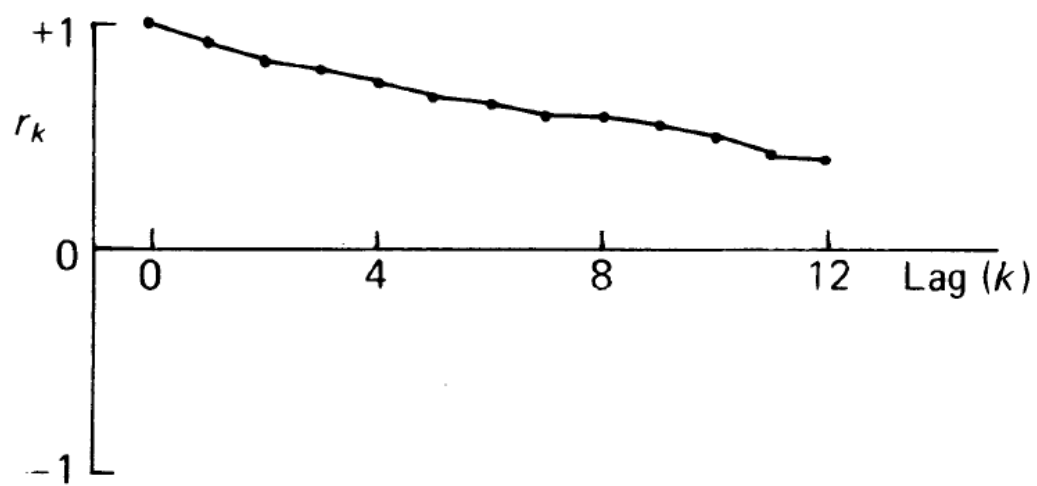
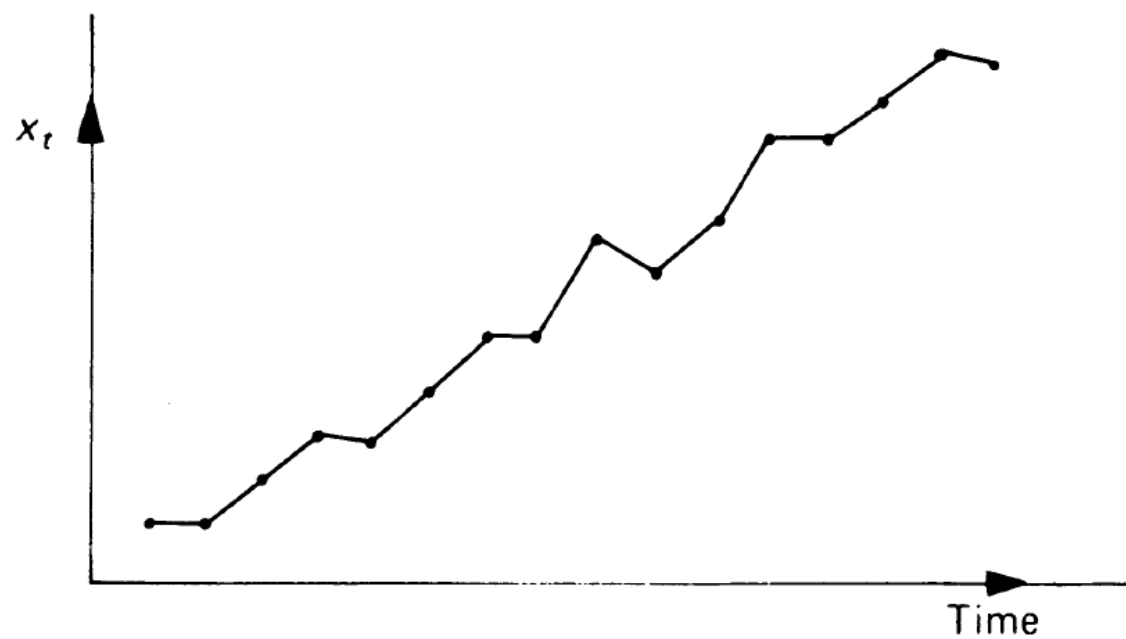
- Alternating series



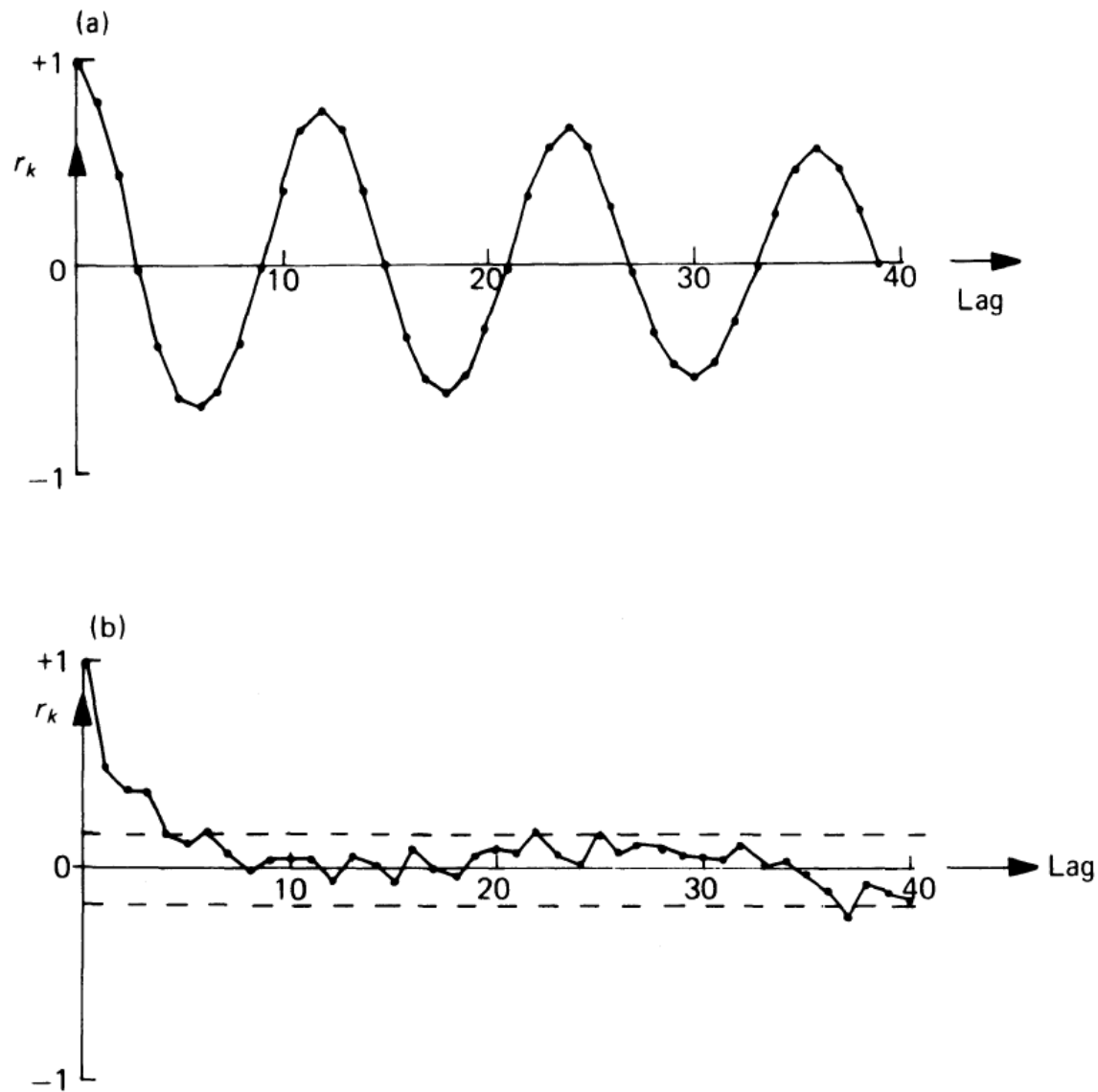
## AR(2) Model



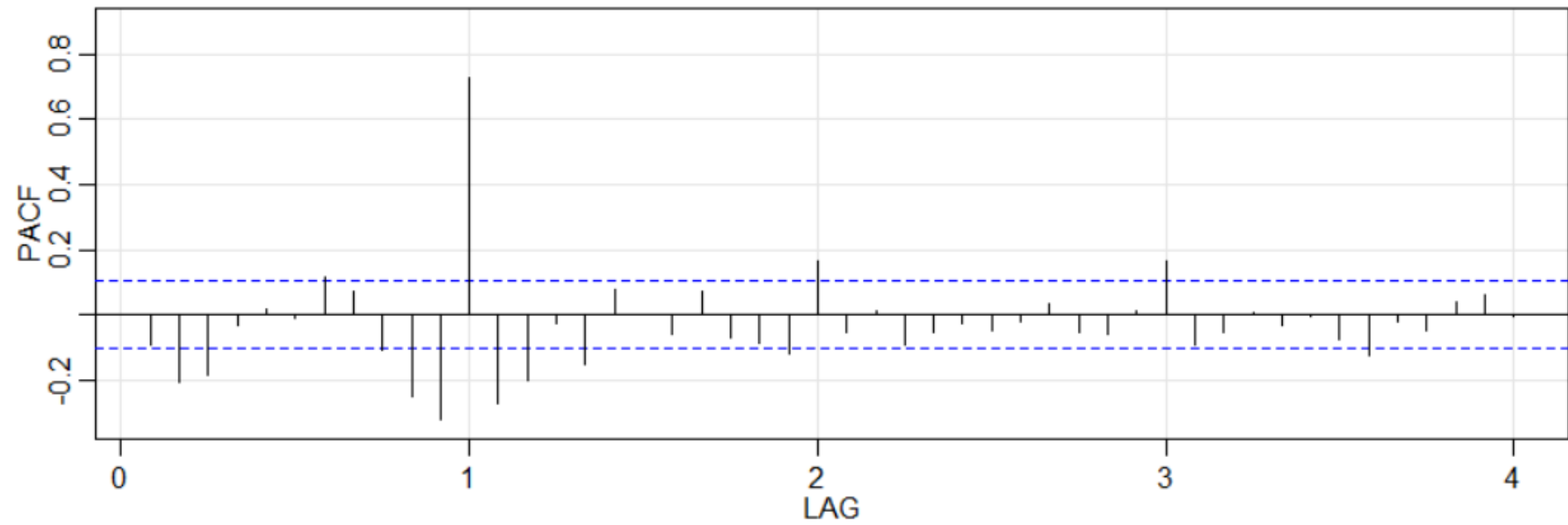
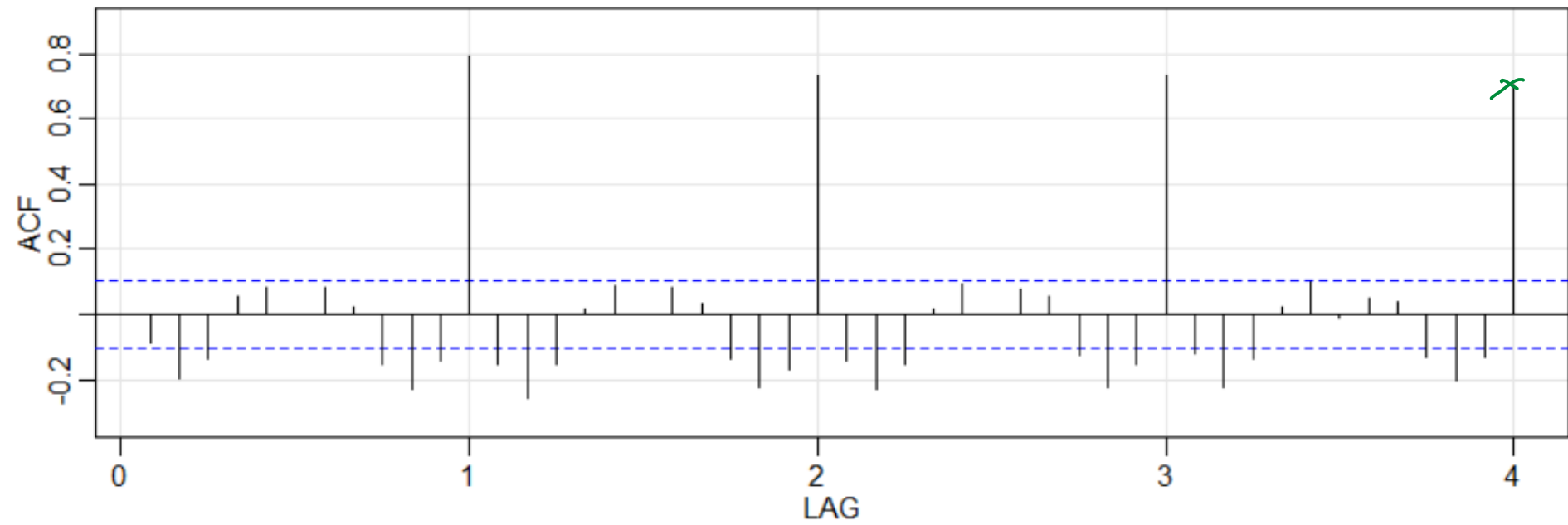
- Non-stationary series

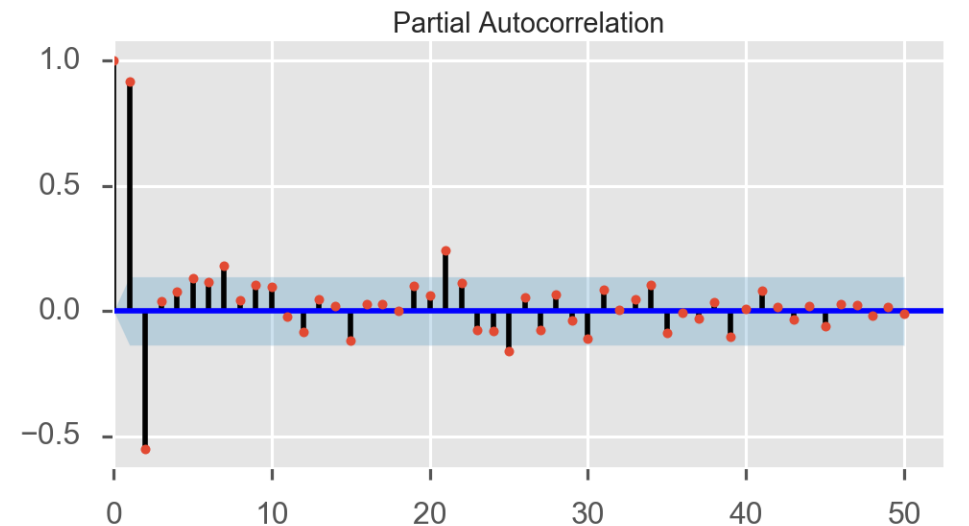
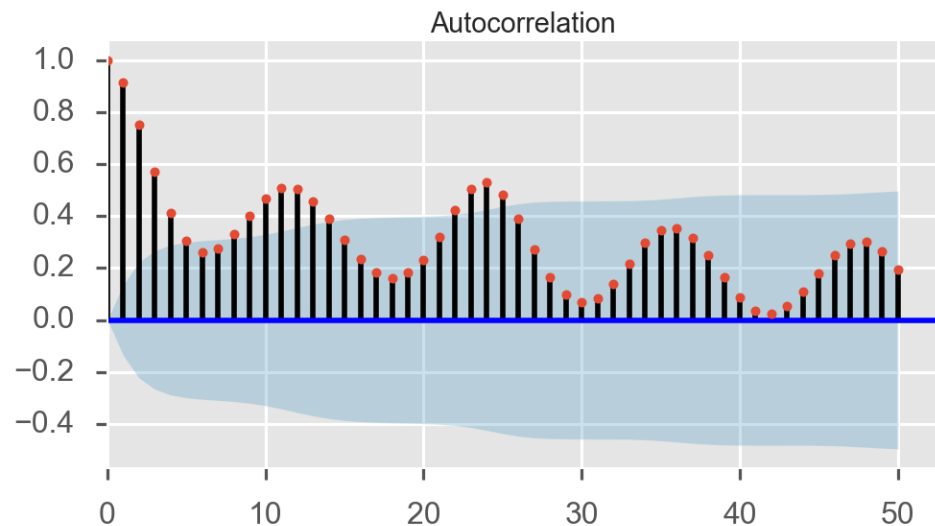
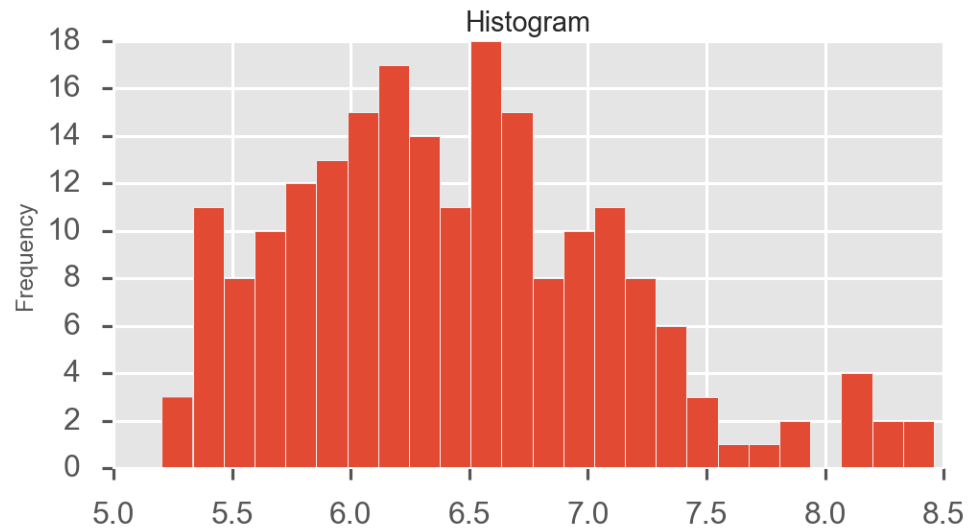
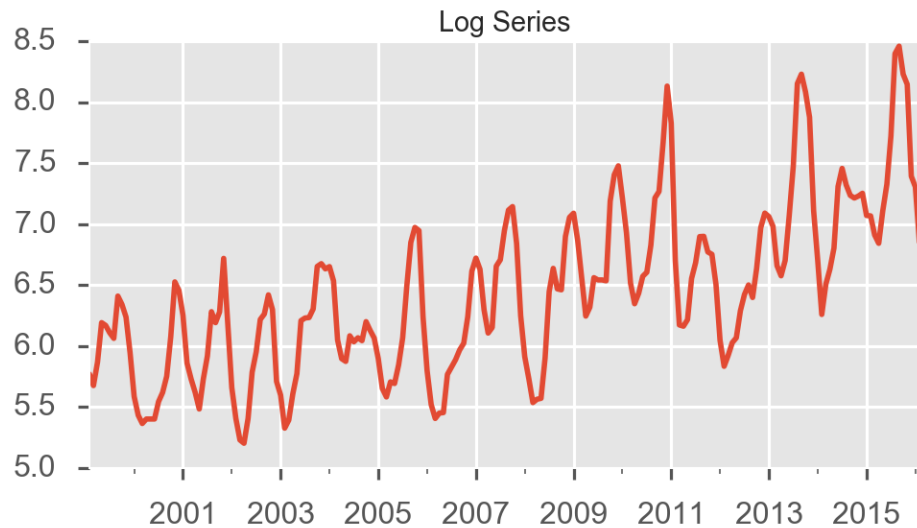


- Seasonal series

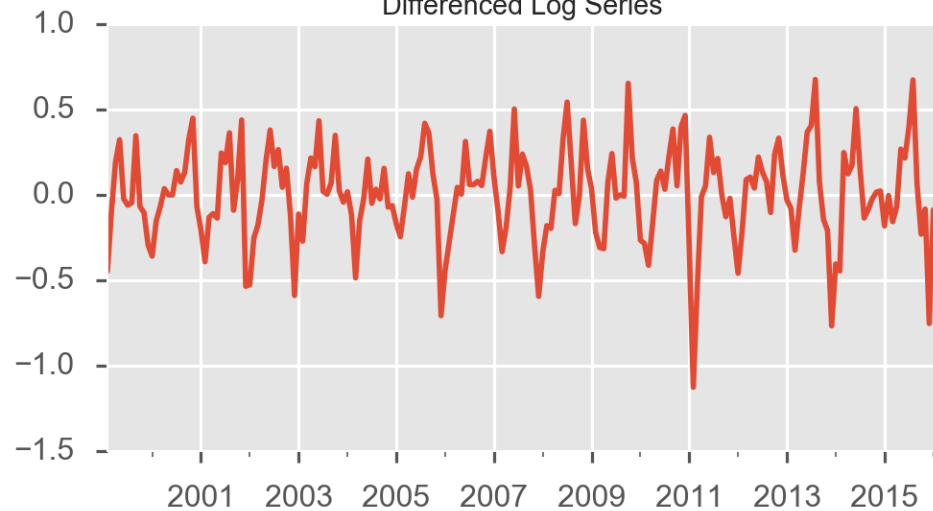


Series: diff(prodn)

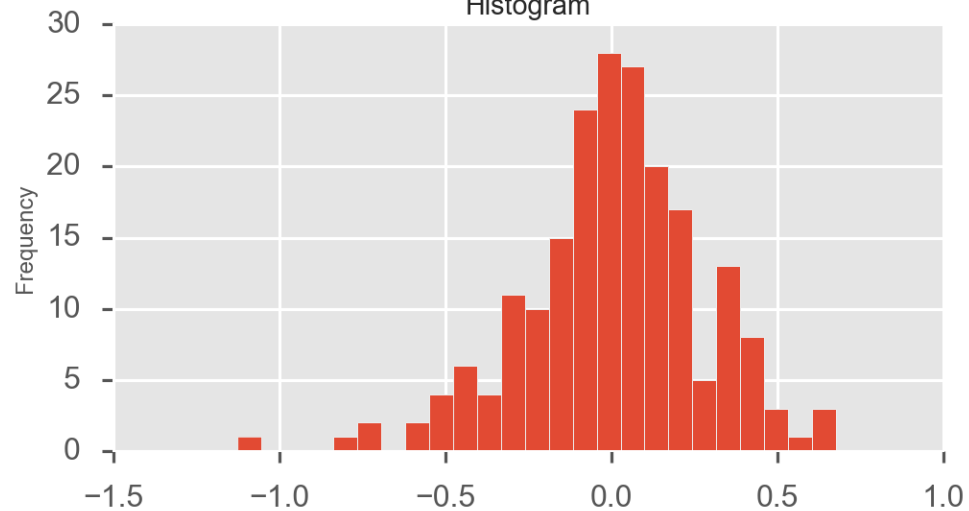




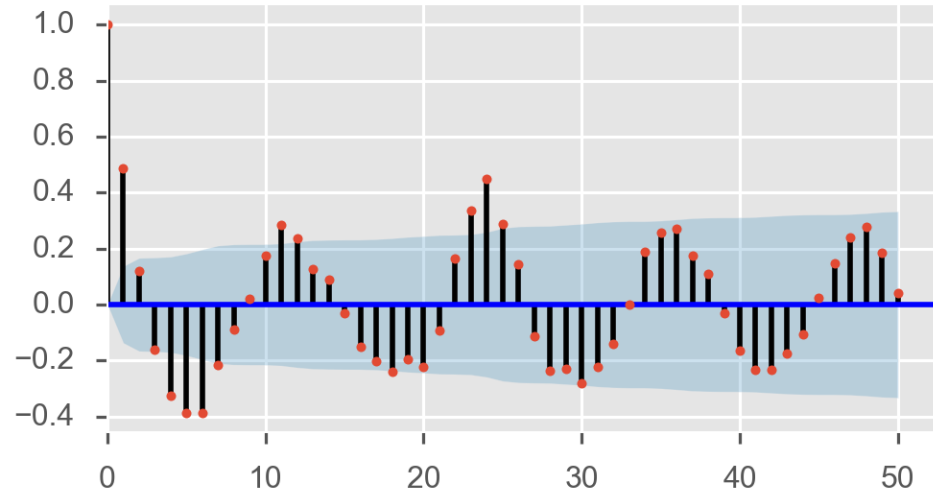
Differenced Log Series



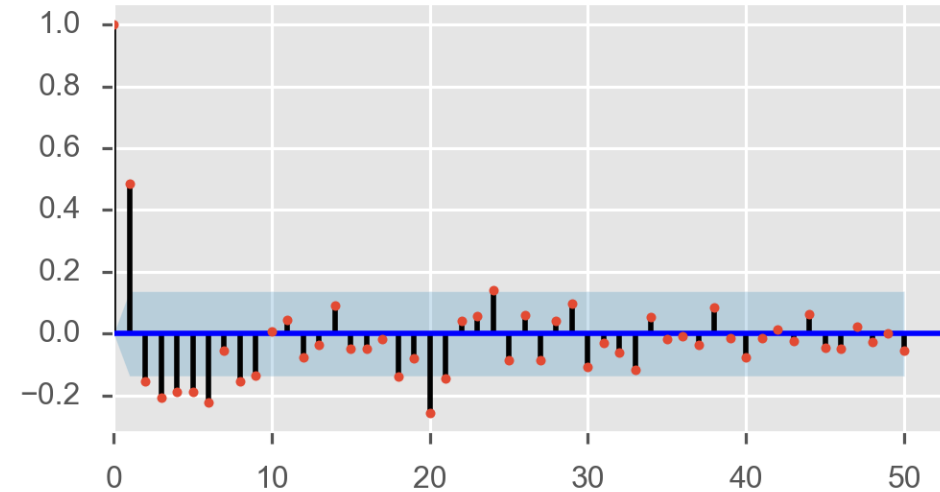
Histogram



Autocorrelation

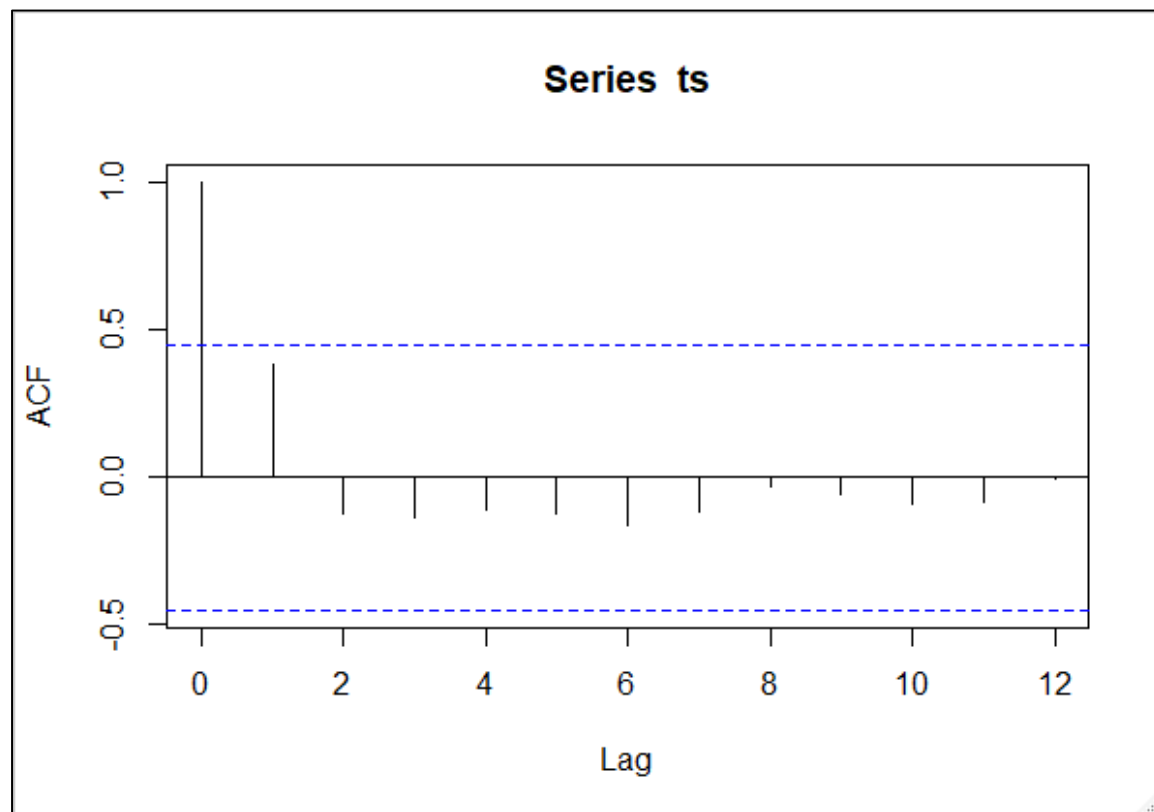


Partial Autocorrelation



- Outliers

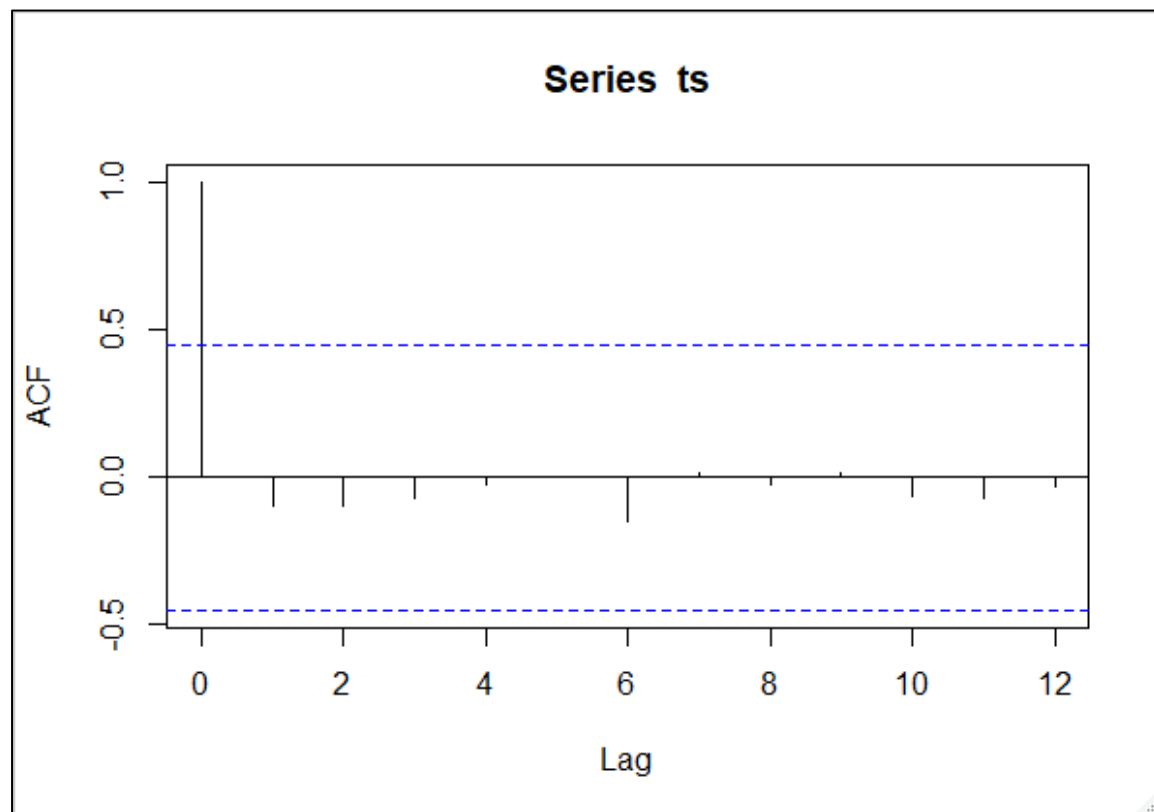
```
ts <- c(1,2,3,1,2,2,30,40,1,3,2,4,1,3,2,3,1,1,2)  
acf(ts)
```





- Outliers

```
ts <- c(1,2,3,1,2,2,30,1,1,3,2,4,1,3,2,3,1,1,2)  
acf(ts)
```



- Outliers

```
ts <- c(1,2,3,1,2,2,30,1,3,2,4,1,3,2,3,1,30,2)  
acf(ts)
```

