Presidency University
The Skill Enhancement Elective Course
STAT03SEEC01
Data Analysis Project

Niranjan Dey

January 17, 2020

Introduction:

- Poliomyelitis is a crippling disease with dramatically visible impact on the patient. A 1985 eradication program has helped guide a more recently launched global eradication effort. In addition, oral polio vaccine (OPV), given in large-scale programs, has been essential to this success. Before 1955, and the licensure of inactivated polio vaccine (IPV) , poliomyelitis was a continuing major cause of permanent disability across the world. In the United States alone, more than 20,000 cases of paralytic polio cases were annually reported during the early 1950s. The process of vaccine manufacture had to be changed, thus resulting in Oral Polio Vaccine (OPV). The success of OPV in the US, Canada, most European countries, and the USSR, made it a logical choice for use in the Americas. Other important reasons to use OPV included the substantially lower cost of OPV compared with IPV; the ability of OPV to induce intestinal immunity, thus facilitating the interruption of wild poliovirus transmission; the capacity of OPV viruses to spread and immunize close contacts; the demonstrated efficacy of OPV in controlling outbreaks; the ease of administration of OPV, a significant advantage in mass campaigns; and the potential ability of OPV viruses to displace the circulation of wild poliovirus in the environment. Over the last fifty years the disease has been brought under control by the use of oral vaccine.
- We have a data on Polio Cases from February 1960 to January 1975. Our project is based on **Time Series analysis** of the given data. It is of considerable interest to identify **trend**, **seasonality** and other features of data on **incidence of polio**.

Data Analysis:

The main problems in the analysis of the time series are:

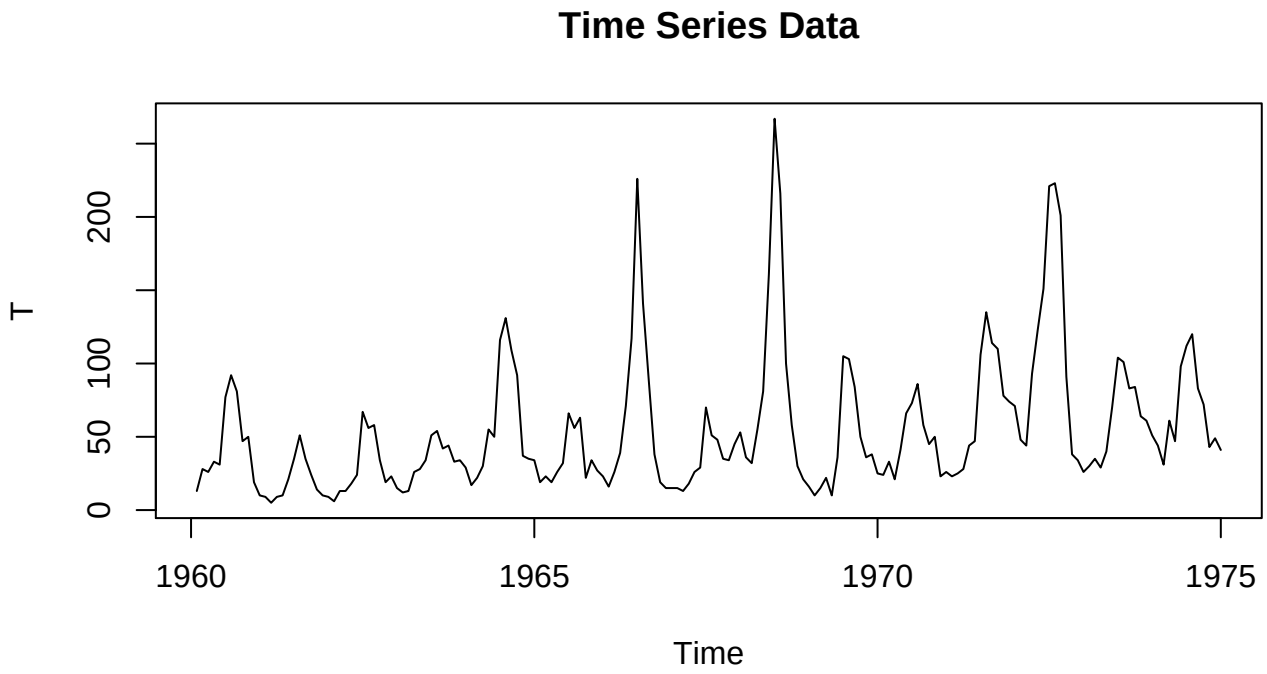
- To identify the forces or components at work, the net effect of whose interaction is exhibited by the movement of a time series.
- To isolate, study, analyse and measure them independently i.e. by holding other things constant.
- To forecast.

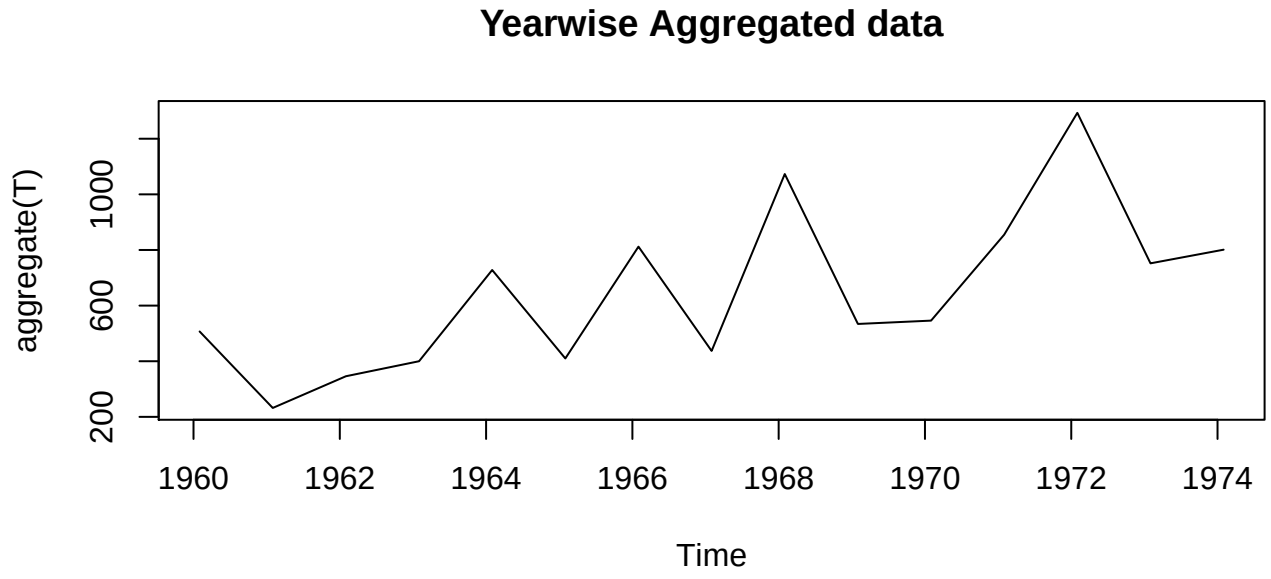
Reading and plotting Time Series Data:

We are given a data frame so first we read it using **read.table()** and making it time series using function **ts()**

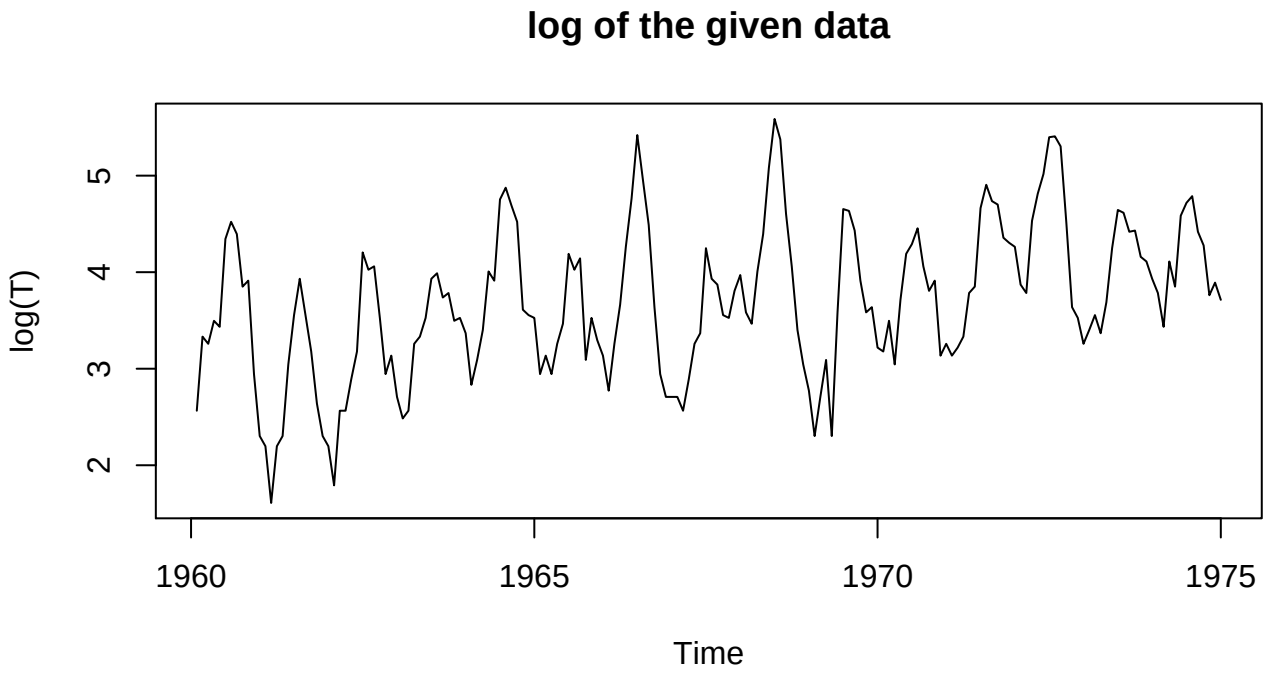
```
P=read.table("D:\\Niranjan Dey\\Poliocases.csv",header=T,sep=",")
T <- ts(P$cases , start = c(1960,2) , end = c(1975,1) , frequency = 12)
```

- Plotting time series and Yearwise aggregated data:





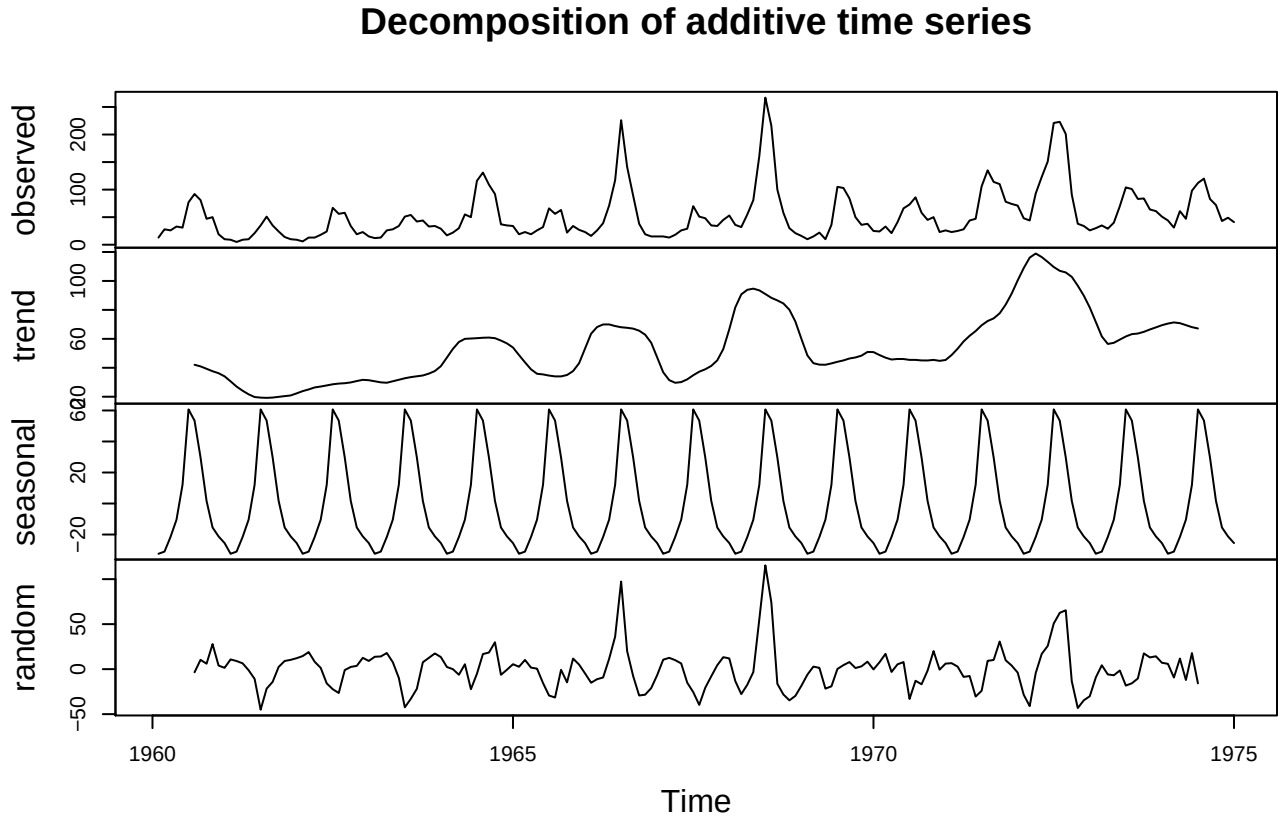
- Aggregated data shows that there is a peak of poliocases in most of the even years (like '60,'64,'66,'68,'72) and a trough in poliocases in almost all of odd years.
- In this case, it appears that an **additive model is not appropriate** for describing this time series, since the size of the seasonal fluctuations and random fluctuations seem to increase with the level of the time series. Thus, we may need to transform the time series in order to get a transformed time series that can be described using an additive model. We can transform the time series by calculating the natural log of the original data:

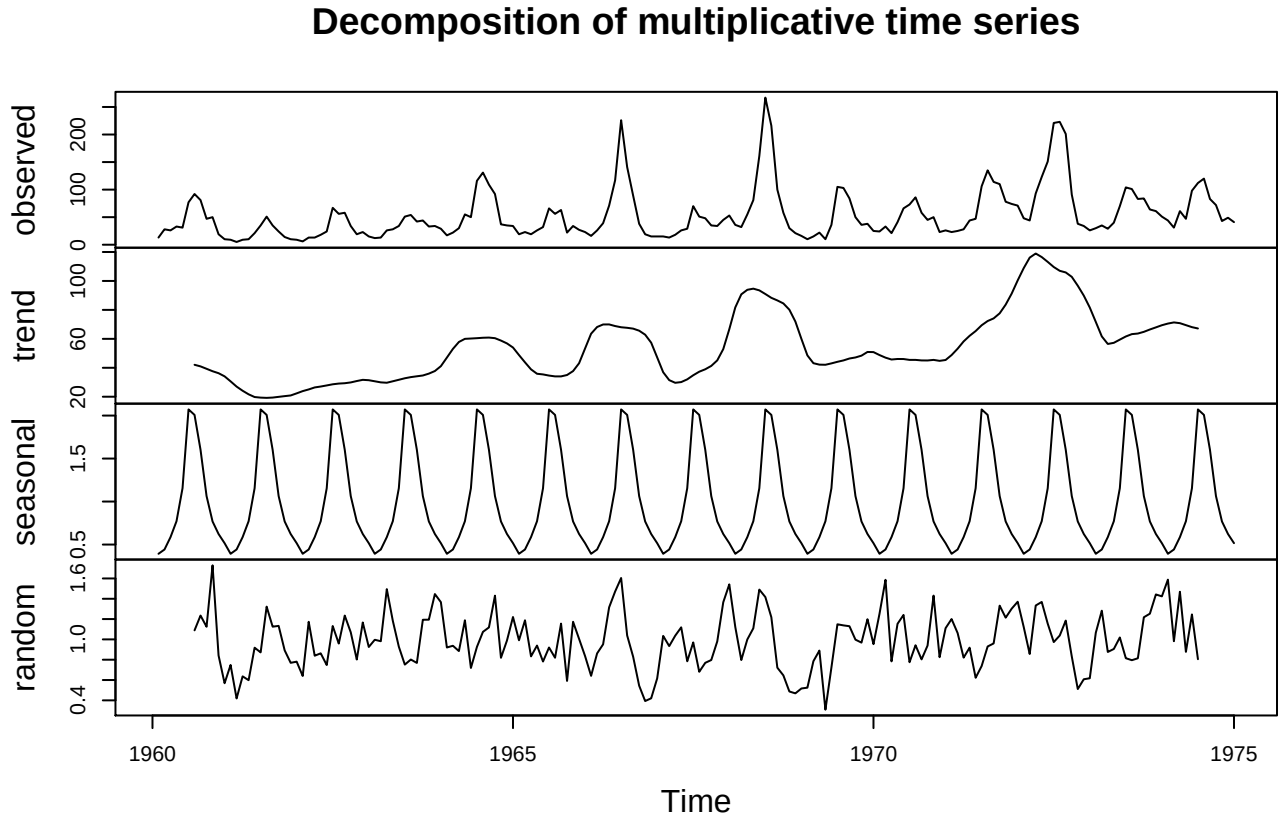


- Here we can see that the size of the seasonal fluctuations and random fluctuations in the log-transformed time series seem to be roughly constant over time, and do not depend on the level of the time series. Thus, the log-transformed time series can probably be described using an additive model.

Decomposing Time Series:

Decomposing a time series means separating it into its constituent components, which are usually a trend component and an irregular component, and if it is a seasonal time series, a seasonal component.


Decomposition of additive time series
observed
trend
seasonal
random
Time
1960
1965
1970
1975

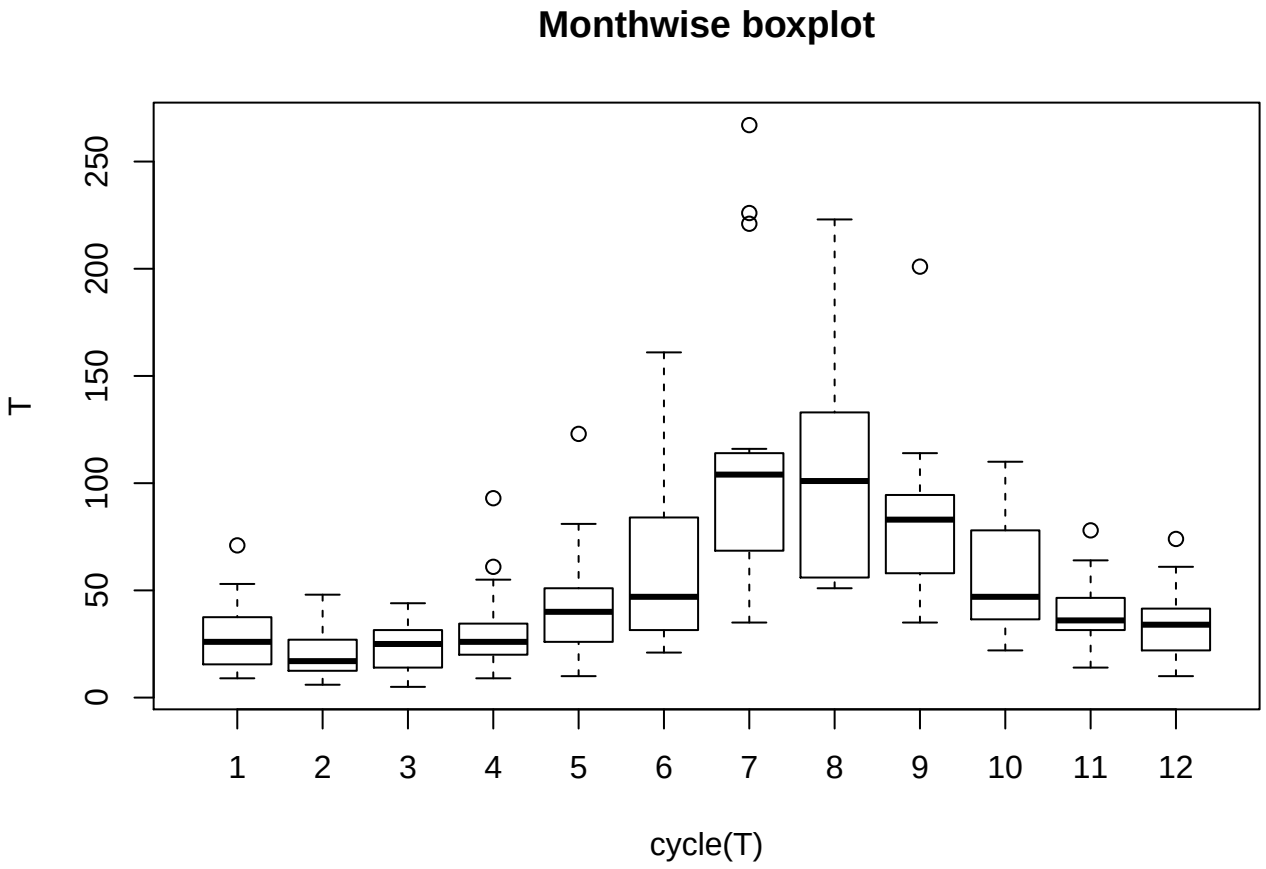

Decomposition of multiplicative time series
observed
trend
seasonal
random
Time
1960
1965
1970
1975

- The estimated values of the trend component: (12 point moving averages)

```
           Jan       Feb       Mar       Apr       May       Jun       Jul
1960                  NA        NA        NA        NA        NA        NA
1961  34.08333  30.62500  27.00000  24.12500  21.66667  19.79167  19.37500
1962  22.33333  23.87500  25.04167  26.41667  27.04167  27.79167  28.58333
1963  31.41667  30.66667  29.91667  29.66667  30.66667  31.70833  32.75000
1964  41.12500  47.04167  53.04167  57.83333  60.00000  60.20833  60.45833
1965  54.00000  48.79167  43.75000  38.91667  35.87500  35.41667  34.62500
1966  53.33333  63.54167  68.16667  69.91667  69.95833  68.83333  68.00000
1967  47.16667  36.91667  31.45833  29.62500  30.12500  32.00000  34.83333
1968  66.62500  81.70833  90.75000  93.87500  94.66667  93.50000  90.95833
1969  60.00000  48.54167  43.16667  42.16667  42.08333  43.04167  44.12500
1970  50.83333  48.79167  47.00000  45.70833  46.08333  46.04167  45.45833
1971  45.37500  48.79167  53.16667  58.20833  62.08333  65.37500  69.37500
1972 100.37500 108.83333 116.12500 118.95833 116.50000 113.16667 109.62500
1973  81.62500  71.66667  61.66667  56.45833  57.25000  59.45833  61.62500
1974  69.41667  70.54167  71.33333  70.83333  69.45833  68.08333  67.16667
1975        NA
           Aug       Sep       Oct       Nov       Dec
1960  42.08333  40.95833  39.29167  37.62500  36.25000
1961  19.20833  19.41667  19.91667  20.41667  20.87500
1962  29.08333  29.33333  29.87500  30.83333  31.66667
1963  33.54167  34.12500  34.66667  35.95833  37.75000
1964  60.75000  60.87500  60.45833  58.79167  56.83333
1965  34.04167  34.04167  35.00000  37.70833  43.12500
1966  67.62500  67.04167  65.62500  62.87500  57.33333
1967  37.29167  38.95833  41.29167  45.12500  52.91667
1968  88.33333  86.54167  84.45833  80.12500  71.95833
1969  45.08333  46.41667  47.12500  48.37500  50.91667
1970  45.45833  45.08333  45.04167  45.45833  44.79167
1971  72.29167  74.12500  77.62500  83.62500  91.25000
1972 107.00000 105.87500 102.83333  96.70833  89.87500
1973  63.25000  63.66667  64.83333  66.45833  67.91667
1974        NA        NA        NA        NA        NA
1975
```

- The estimated values of the seasonal component:

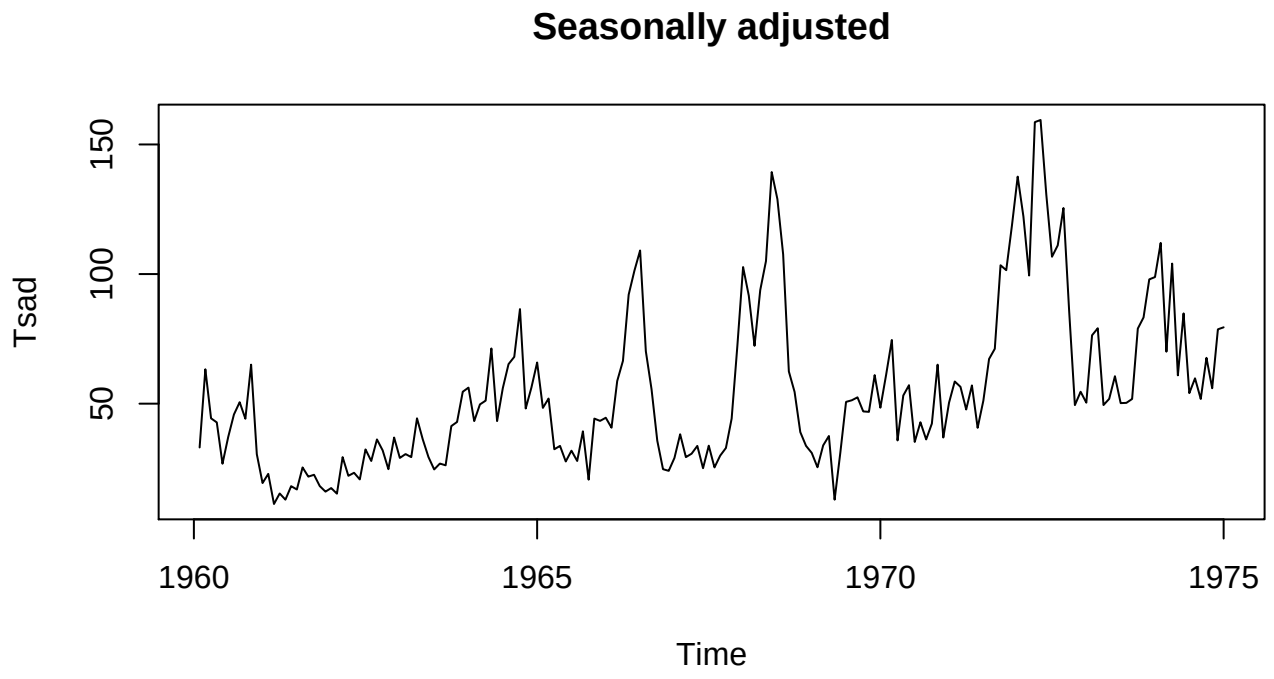
```
 [1] -32.415179 -31.004464 -21.584821 -10.424107  12.078869  60.683036
 [7]  53.462798  29.718750   1.677083 -15.468750 -21.209821 -25.513393
```



- The estimated seasonal factors are given for the months January-December, and are the same for each year. The largest seasonal factor is for July (about 2.07148), and the lowest is for February (about 0.3927455), indicating that there seems to be a peak in poliocases in July and a trough in poliocases in February each year. Also the **boxplot supports the conclution**.

Seasonally Adjusting:

We have a seasonal time series that can be described using an multiplicaitive model, you can seasonally adjust the time series by estimating the seasonal component, and dividing the estimated seasonal component from the original time series. Then plotting the seasonally adjusted time series using the "plot()" function:

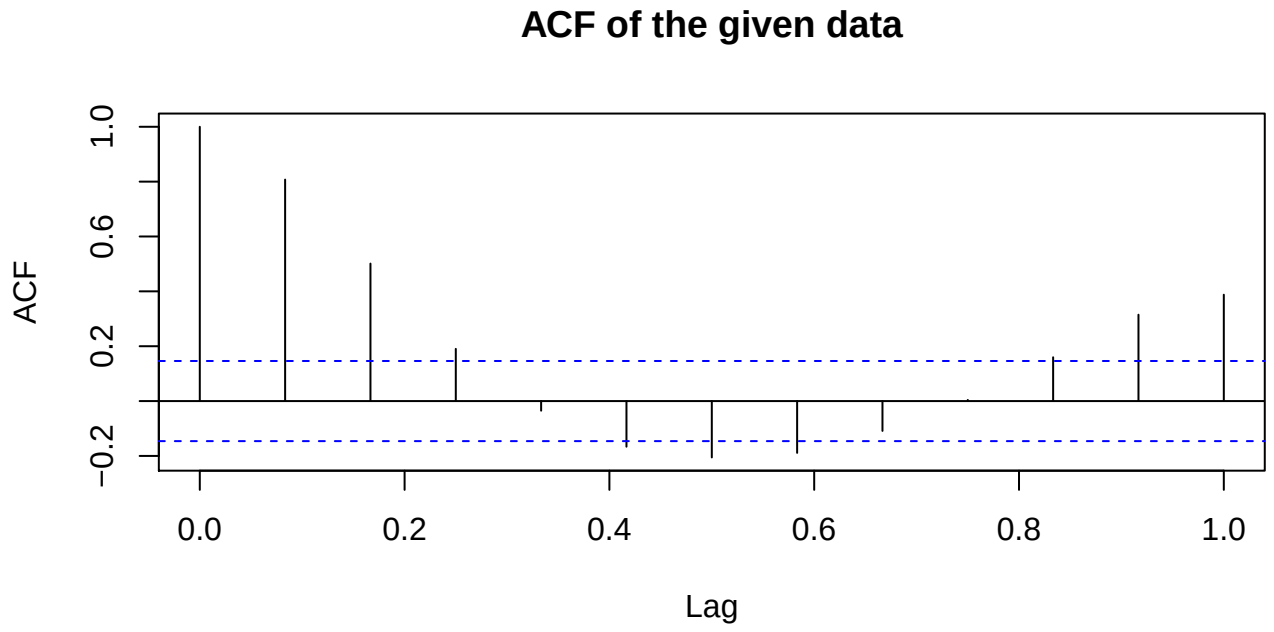


- You can see that the seasonal variation has been removed from the seasonally adjusted time series. The seasonally adjusted time series now just contains the trend component and an irregular component.
- **Justification of Seasonal Adjustment:**

Now we will find standard deviations of the original series, trend, and random element seperately:

```
sd(T[7:174])

[1] 46.77295

sd(T[7:174]-T.mul$trend[7:174])

[1] 38.6201

sd(T.mul$random[7:174])

[1] 0.2766785
```

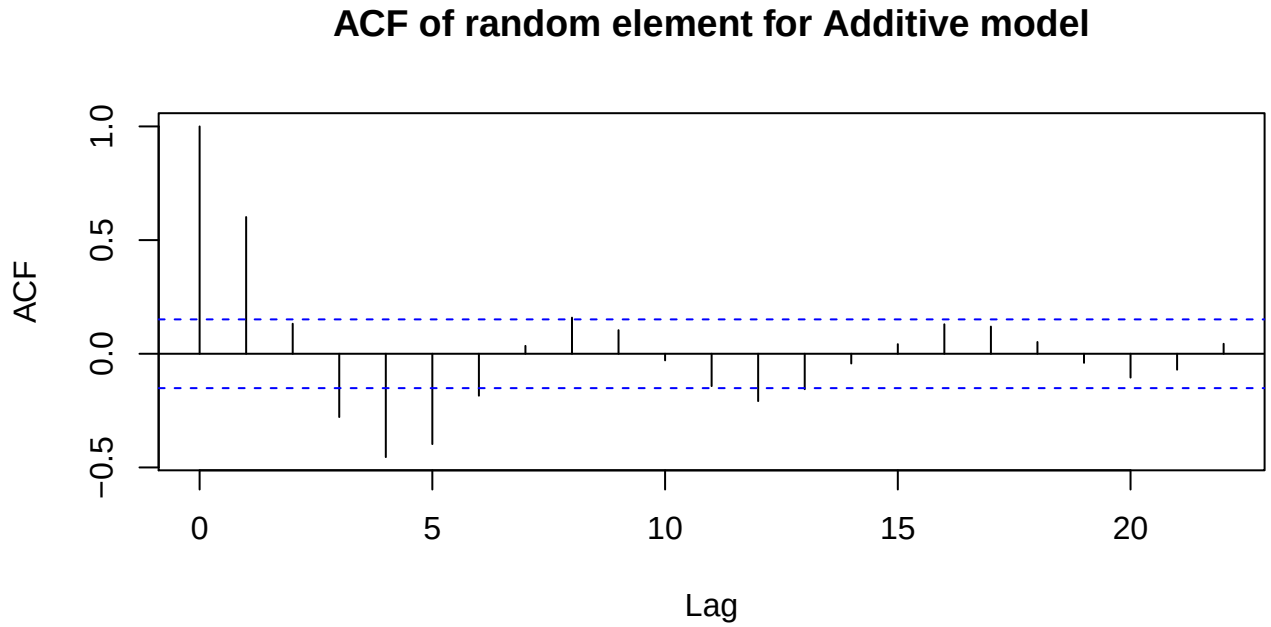
- The reduction in the standard deviation shows that the seasonal adjustment has been very effective.

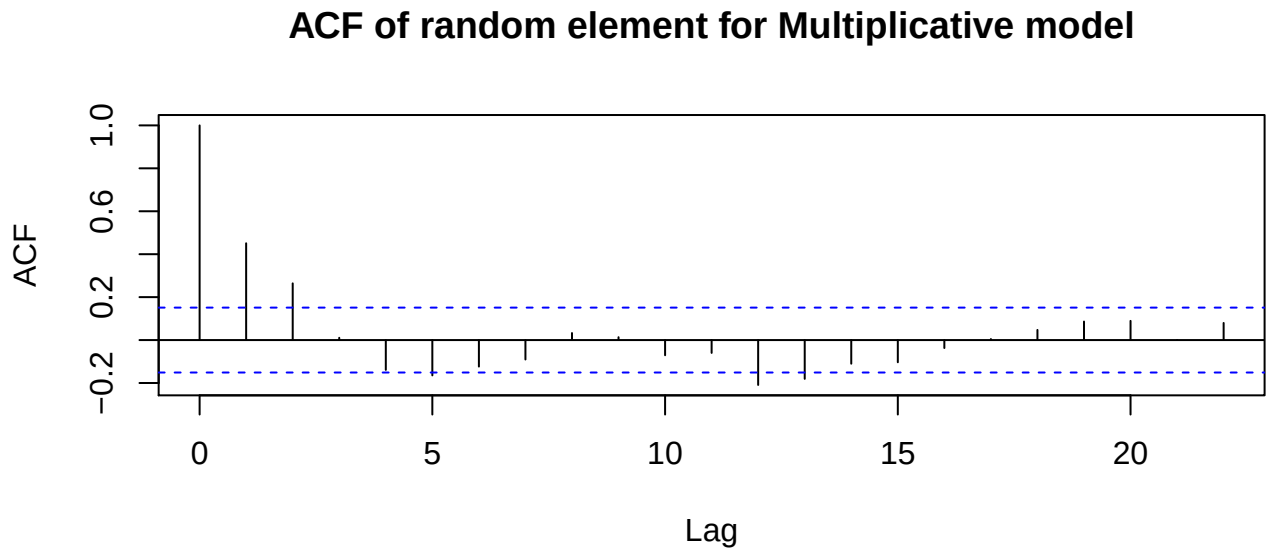


- For monthly series, a significant autocorrelation at lag 12 indicates that the seasonal adjustment is adequate.

Random element:

First to compare between Additive model & Multiplicative model we shall look at the correlogram of the ramdom element:



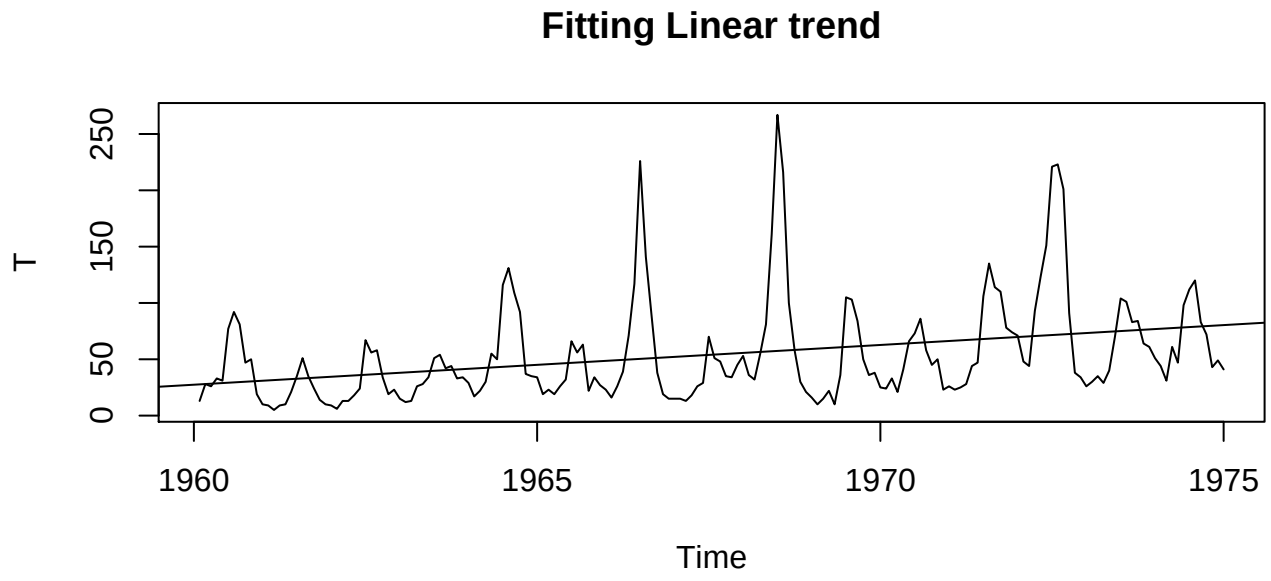


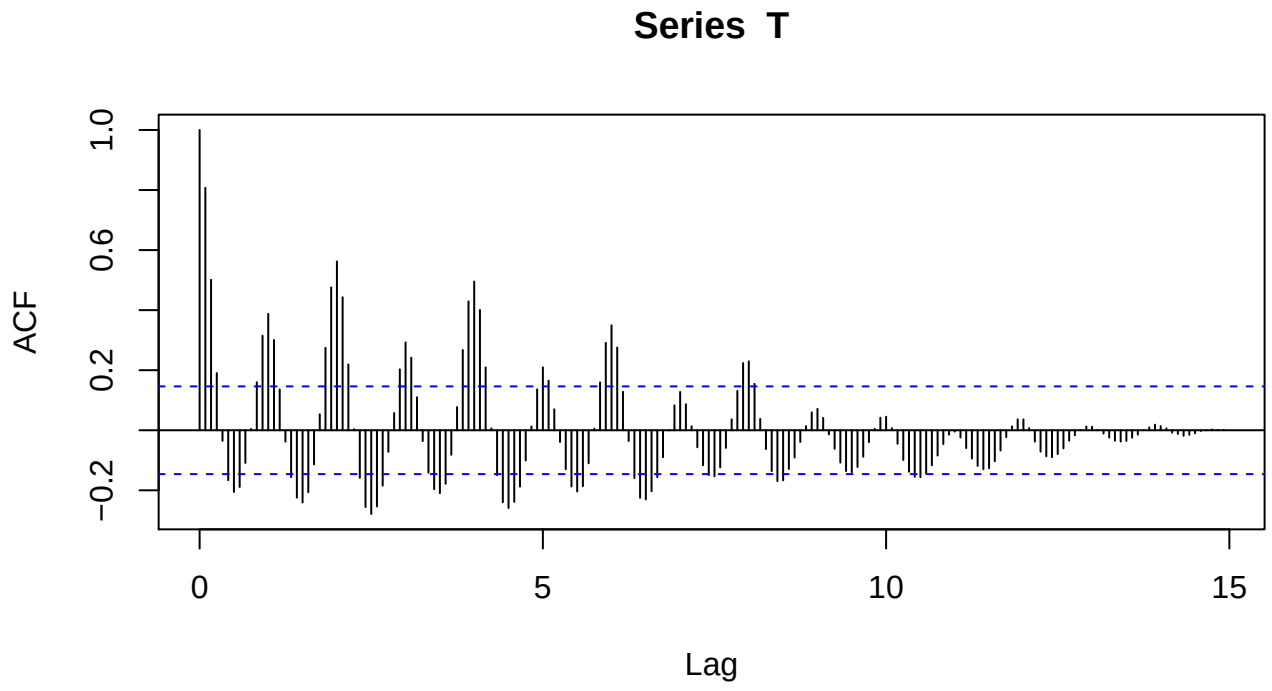
- It is clear that Autocorrelation exceeds confidence bound more number of cases in Additive Model than Multiplicative Model.

Adding a Straight line by Least Squares Method:

The linear model of first degree is:

```
Call:
lm(formula = T ~ time(T))

Coefficients:
(Intercept)      time(T)
  -6887.328        3.528
```





- The correlogram for wave heights has a well-defined shape that appears like a sampled damped cosine function. This is typical of correlograms of time series generated by an autoregressive model of order 2 **AR(2).**

```
[1] 0.3343137
```

- Correlation between observed and fitted values is very low (about **0.3343).**

To test if the time series is stationary:

Use Augmented Dickey-Fuller Test (adf test). A p-Value of less than 0.05 in adf.test() indicates that it is stationary.

```
library(tseries)

Registered S3 method overwritten by 'xts':
method    from
as.zoo.xts zoo
Registered S3 method overwritten by 'quantmod':
method            from
as.zoo.data.frame zoo

adf.test(T) # p-value < 0.05 indicates the TS is stationary
```

```
Augmented Dickey-Fuller Test

data:  T
Dickey-Fuller = -6.1301, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

- Hence the data given is stationary. **No need to make it stationary.**

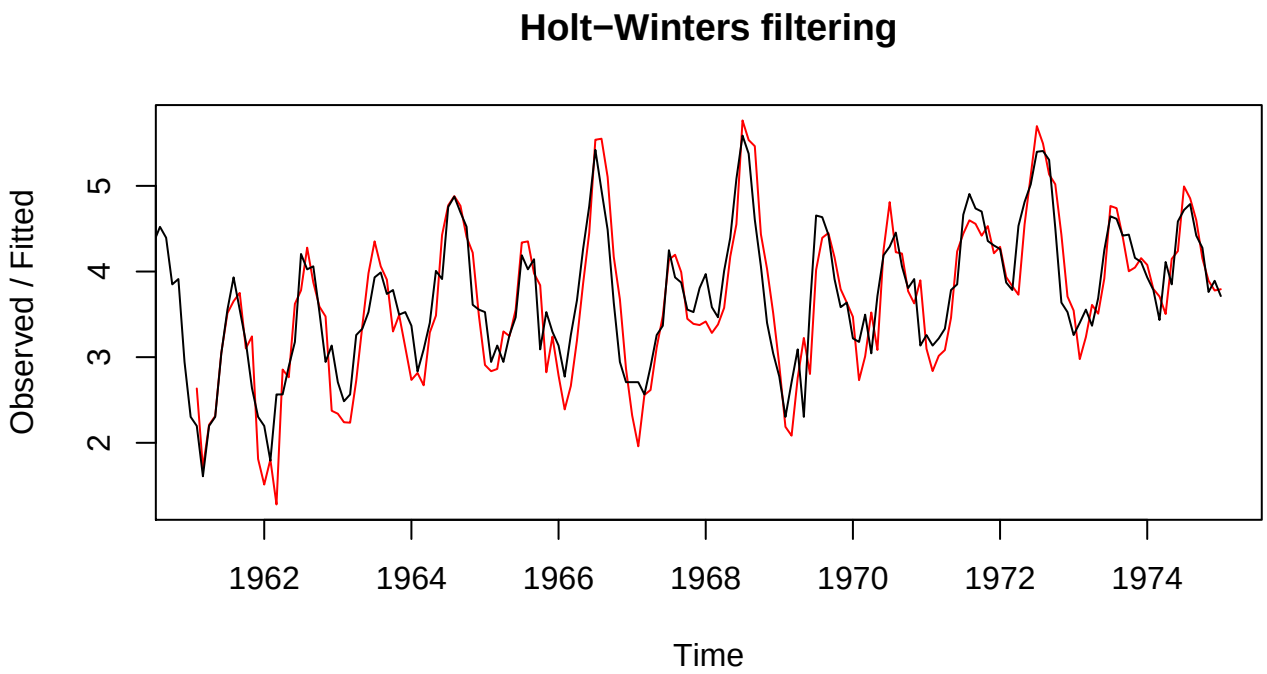
Holt's Exponential Smoothing:

- Holt-Winters exponential smoothing estimates the level, slope and seasonal component at the current time point. Smoothing is controlled by three parameters: alpha, beta, and gamma, for the estimates of the level, slope b of the trend component, and the seasonal component, respectively, at the current time point. The parameters alpha, beta and gamma all have values between 0 and 1, and values that are close to 0 mean that relatively little weight is placed on the most recent observations when making forecasts of future values.

```
library(forecast)

Registered S3 methods overwritten by 'forecast':
method            from
fitted.fracdiff    fracdiff
residuals.fracdiff fracdiff

H=HoltWinters(log(T))
H

Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = log(T))

Smoothing parameters:
 alpha: 0.7490159
 beta : 0.01092434
 gamma: 0.9561161

Coefficients:
            [,1]
a    3.819787852
b   -0.009931514
s1  -0.218085184
s2  -0.350622847
s3  -0.126815593
s4  -0.148559739
s5   0.328735973
```

```
s6   0.680788213
s7   0.805605815
s8   0.582351331
s9   0.354263209
s10 -0.051258355
s11  0.002008394
s12 -0.105340930

H$SSE   #---Measure of Error---

[1] 24.41007
```

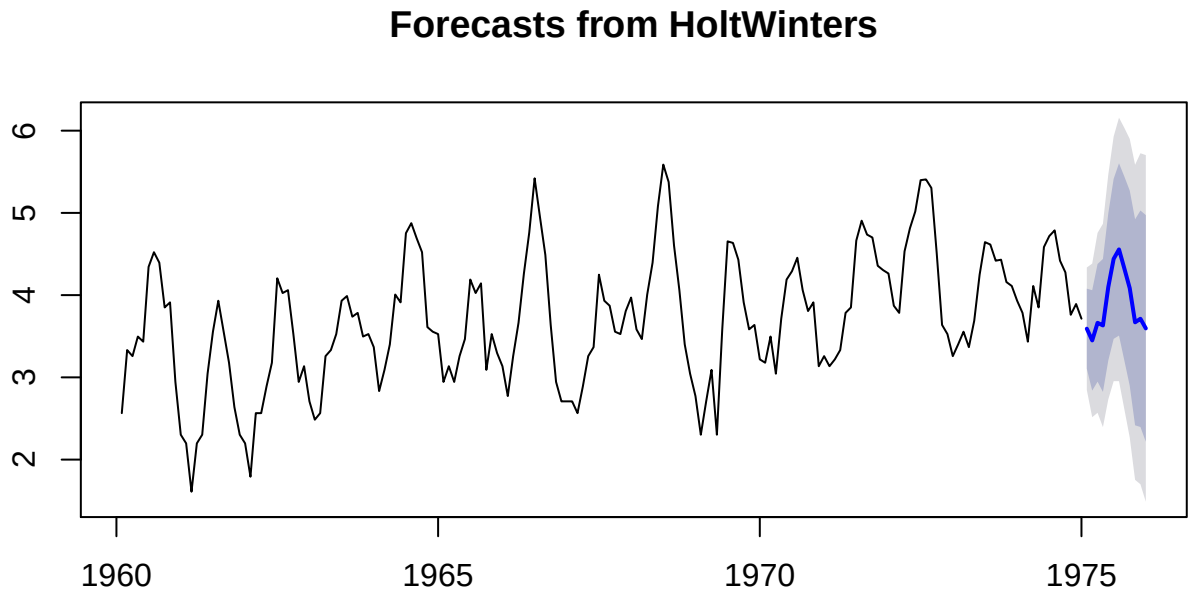
- The estimated values of alpha, beta and gamma are 0.749, 0.01, and 0.956, respectively. The value of alpha (0.749) is high enough, indicating that the estimate of the level at the current time point is based upon mainly recent observations and some observations in the more distant past. The value of beta is 0.01, indicating that the estimate of the slope b of the trend component is not updated over the time series, and instead is set equal to its initial value. This makes good intuitive sense, as the level changes quite a bit over the time series, but the slope b of the trend component remains roughly the same. In contrast, the value of gamma (0.956) is high, indicating that the **estimate of the seasonal component** at the current time point is just based upon very recent observations.
- As for simple exponential smoothing and Holt's exponential smoothing, we can plot the original time series as a black line, with the forecasted values as a red line on top of that:



- We see from the plot that the Holt-Winters exponential method is very successful in predicting the seasonal peaks, which occur roughly in July every year.

Forecasts using Exponential Smoothing:

- To make forecasts for future times not included in the original time series, we use the "forecast()" function in the "forecast" package. For example, the original data for Poliocases is from February 1960 to January 1975. If we wanted to make **forecasts for February 1975 to January 1976 (12 more months)**, and plot the forecasts:



- The forecasts are shown as a blue line, and the dark gray and light gray shaded areas show 80% and 95% prediction intervals, respectively.

Conclusion and Discussions:

Based on our Data analysis,

(i) We verify the fact that in the light of the given data, the linear trend fails to explain the data. The correlation between observed and fitted values is low (about **0.3343**) for linear trend.

(ii) In this data the seasonal component is strong enough as expressed in both Decomposition & Holt's Exponential Smoothing.

(iii) From observation the largest seasonal factor is for July (about 2.07148), and the lowest is for February (about 0.3927455).

(iv) In the light of given data, there is a peak of poliocases in most of the **even years** (like '60,'64,'66,'68,'72) and a trough in poliocases in almost all of **odd years**.

(v) It is evident that Multiplicative Model fits the better than Additive Model comparing the random element.

(vi) The random element has small standard deviation and Autocorrelation does not exceeds the confidence levels for most of the time. So it can be ignored.

R codes:

```
#---Reading and plotting Time Series Data---
P=read.table("D:\\Niranjan Dey\\Poliocases.csv",header=T,sep=",")
T <- ts(P$cases , start = c(1960,2) , end = c(1975,1) , frequency = 12)
plot(T,main="Time Series Data")
plot(aggregate(T),main="Yearwise Aggregated data")
plot(log(T),main="log of the given data")
#---Decomposing Time Series---
T.addi=decompose(T)
T.mul=decompose(T,type="mult")
plot(T.addi)
plot(T.mul)
T.addi$trend
T.addi$figure
boxplot(T~cycle(T),main="Monthwise boxplot")
#---Seasonal Adjustment---
Tsad=(T/T.mul$seasonal)
plot(Tsad,main="Seasonally adjusted")
sd(T[7:174])
sd(T[7:174]-T.mul$trend[7:174])
sd(T.mul$random[7:174])
acf(T,lag.max=12)
#---Random element---
acf(T.addi$random[7:174],main="ACF of random element for Additive model")
acf(T.mul$random[7:174],main="ACF of random element for Multiplicative model")
#---Adding a Straight line by Least Squares Method---
m1=lm(T~time(T))
m1
plot(T,main="Fitting Linear trend")
abline(m1)
acf(T,lag.max=200)
Tht=-6887.328+3.528*time(T)
cor(T,Tht)
#---To test if the time series is stationary---
library(tseries)
adf.test(T) #---p-value < 0.05 indicates the TS is stationary
#---Holts Exponential Smoothing---
library(forecast)
H=HoltWinters(log(T))
H
H$SSE
plot(H)
K=forecast(H, h=12)
plot(K)
```

References:

The following books were used as a reference to the analysis done in the report:-

- Fundamentals of Applied Statistics, by S.C. Gupta & V.K. Kapoor
- Time Series Analysis Jonathan With Applications in R, by D. Cryer ,Kung-Sik Chan
- Introductory Time Series with R, by Paul S.P. Cowpertwait · Andrew V. Metcalfe
- Time Series Analysis and Its Applications With R Examples, by Robert H. Shumway , David S. Stoffer
- [online] Available from : http://r-statistics.co/Time-Series-Analysis-With-R.html
- [online] Available from : https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.ht
- [online] Available from : https://www.statmethods.net/advstats/timeseries.html

Acknowledgment:

I would like to express my sincere gratitude to my respected instructor **Prof. Atanu Kumar Ghosh** for his guidance and constant supervision as well as for providing all the necessary information regarding the project & also for his support in completing this project.