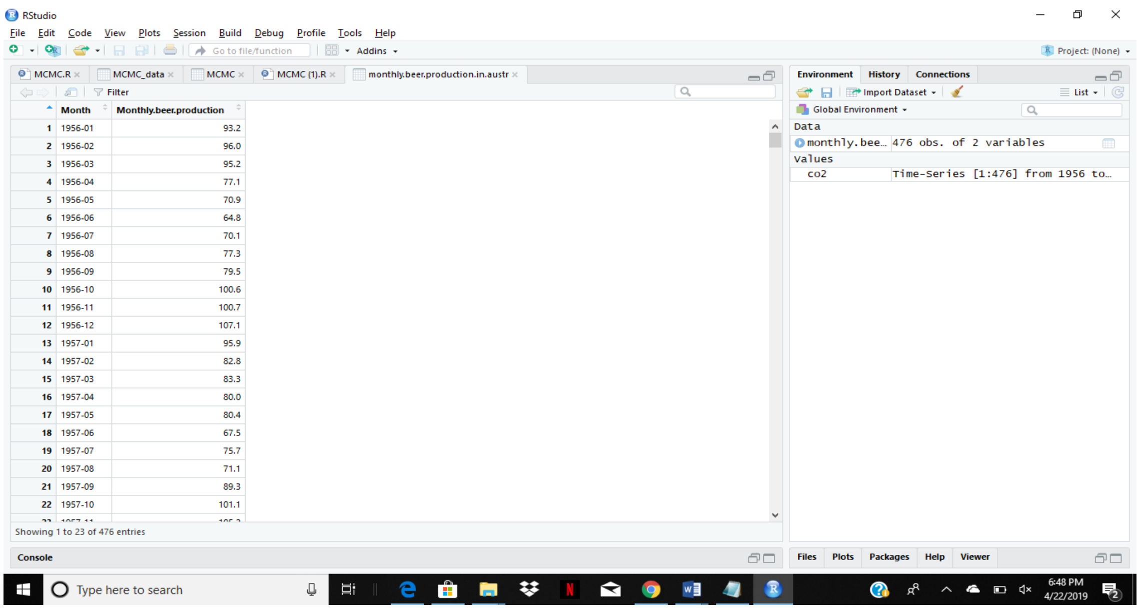
- Techniques used for time series analysis:
  - 1. ARIMA models
  - 2. Box-Jenkins models

- Data file used for this analysis is monthly.beer.production.in.austr. There are 2 fields only: Year and month, beer production quantity. This is a univariate data.
- Viewing it in R:



For ARIMA Model:

- Assumptions made are
  - constant mean, covariance is a function of lag.
  - The future extrapolation represents past trend.
- Data should be stationary – by stationary it means that the properties of the series doesn't depend on the time when it is captured. A white noise series and series with cyclic behavior can also be considered as stationary series.

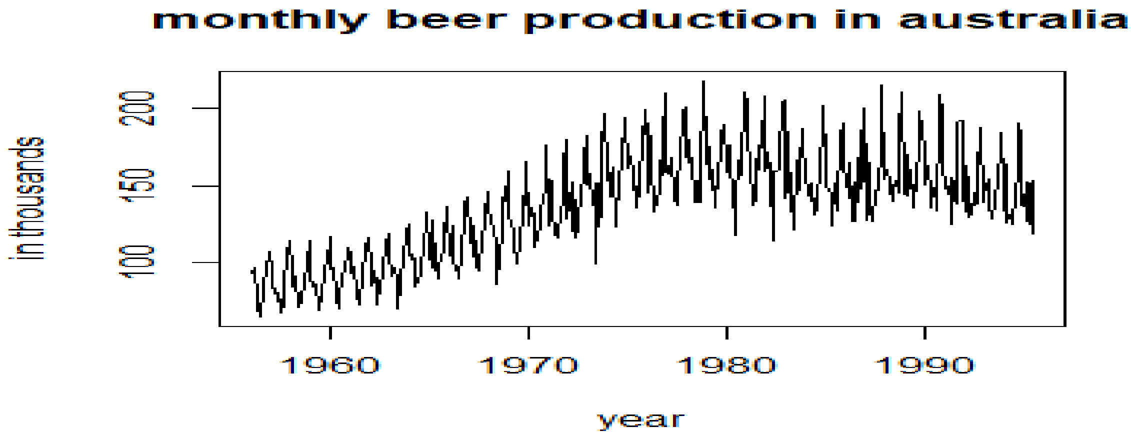
- 2. Data should be univariate – ARIMA works on a single variable. Auto-regression is all about regression with the past values.

- Converting this data into time series using ts function:

```
mydata= ts(monthly.beer.production.in.austr$Monthly.beer.production,
frequency = 12,start=c(1956,1))
```

```
plot(mydata,xlab='year',ylab="in thousands",main='monthly beer production  in australia')
```

- The frequency above states the number of times in the year. Since it is a monthly data, the frequency is set to 12.
- The following graph shows how actual data looks like:

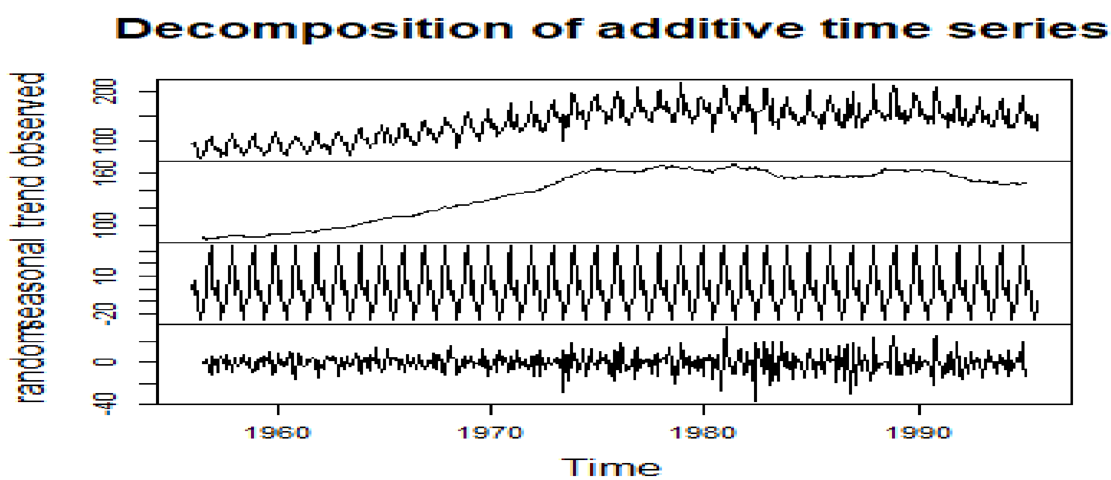


- We can infer from the graph itself that the data points follow an upward trends and then maintains the tends downward.
  - the three components of a time series data:
  - Trend: A long-term increase or decrease in the data is referred to as a trend. It is not necessarily linear. It is the underlying pattern in the data over time.
  - Seasonal: When a series is influenced by seasonal factors i.e. quarter of the year, month or days of a week seasonality exists in the series. It is always of a fixed and known period. E.g. – A sudden rise in sales during Christmas, etc.
  - Cyclic: When data exhibit rises and falls that are not of the fixed period we call it a cyclic pattern. For e.g. – duration of these fluctuations is usually of at least 2 years.

  Components if time series:

  components.ts = decompose(mydata)

  plot(components.ts)



Here we get 4 components:

- Observed – the actual data plot
- Trend – the overall upward or downward movement of the data points
- Seasonal – any monthly/yearly pattern of the data points
- Random – unexplainable part of the data

Observing these 4 graphs closely, we can find out if the data satisfies all the assumptions of ARIMA modeling, mainly, stationarity and seasonality.

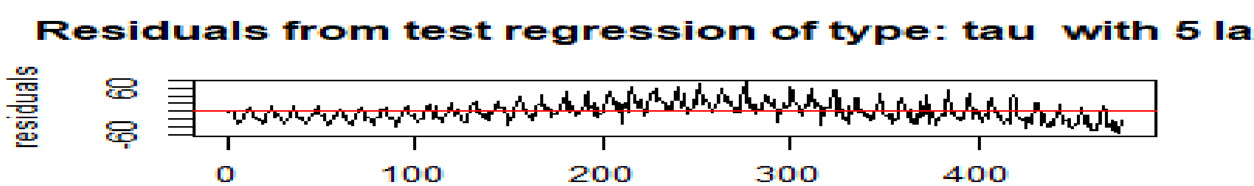
Box Jenkins Modeling:

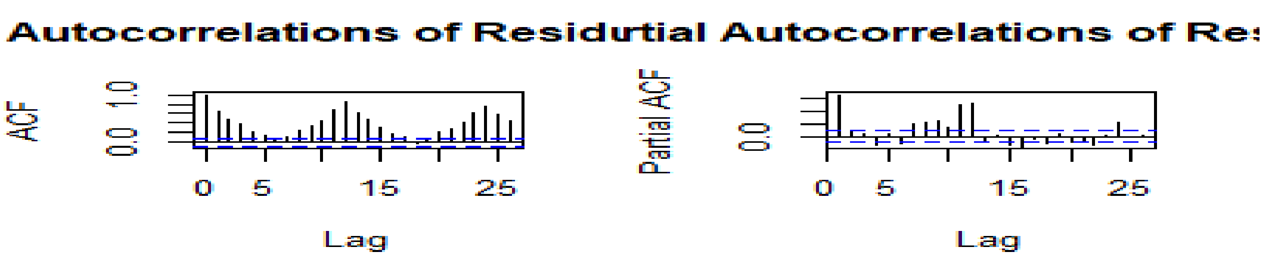
- **Identification**. Use the data and all related information to help select a sub-class of model that may best summarize the data.
- **Estimation**. Use the data to train the parameters of the model (i.e. the coefficients).
- **Diagnostic Checking**. Evaluate the fitted model in the context of the available data and check for areas where the model may be improved.

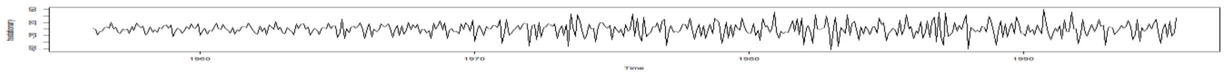
- **Unit Root Tests**. Use unit root statistical tests on the time series to determine whether or not it is stationary. Repeat after each round of differencing.

The code for unit root test:

```
library("fUnitRoots")
urkpssTest(mydata, type = c("tau"), lags = c("short"),use.lag = NULL, doplot = TRUE)
tsstationary = diff(mydata, differences=1)
plot(tsstationary)
```







Configuring AR and MA and Fitting the model

Two diagnostic plots can be used to help choose the *p* and *q* parameters of the ARMA or ARIMA. They are:

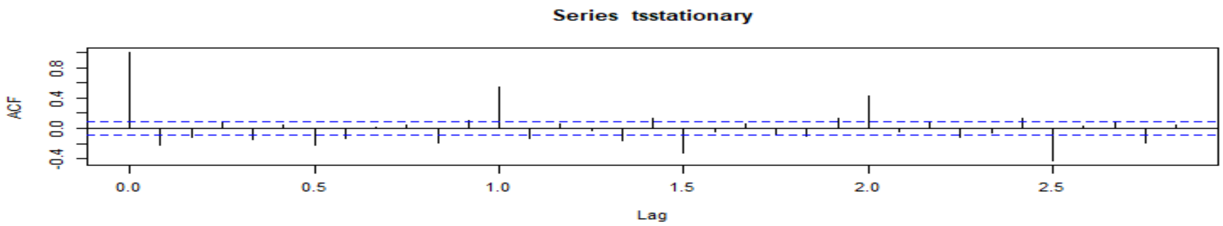
- **Autocorrelation Function (ACF)**. The plot summarizes the correlation of an observation with lag values. The x-axis shows the lag and the y-axis shows the correlation coefficient between -1 and 1 for negative and positive correlation.
- **Partial Autocorrelation Function (PACF)**. The plot summarizes the correlations for an observation with lag values that is not accounted for by prior lagged observations.

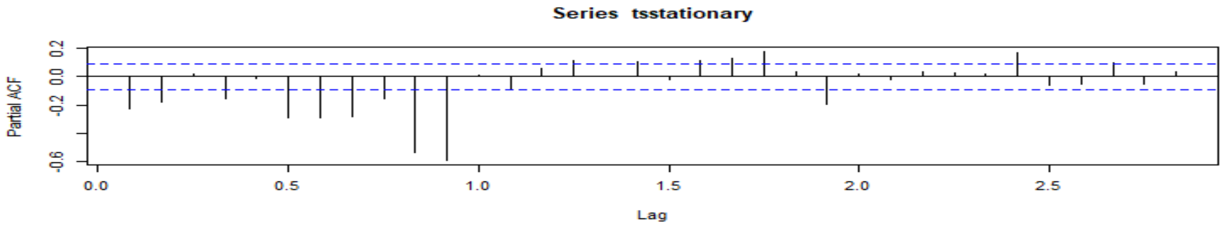
Some useful patterns we observe on these plots are:

- The model is AR if the ACF trails off after a lag and has a hard cut-off in the PACF after a lag. This lag is taken as the value for *p*.
- The model is MA if the PACF trails off after a lag and has a hard cut-off in the ACF after the lag. This lag value is taken as the value for *q*.
- The model is a mix of AR and MA if both the ACF and PACF trail off.

```
acf(tsstationary, lag.max=34)
```

```
pacf(tsstationary, lag.max=34)
```





Looking at the graphs and going through the table we can determine which type of the model to select and what will be the values of p, d and q.

Shape	Indicated Model
Exponential series decaying to 0	Auto Regressive (AR) model. pacf() function to be used to identify the order of the model
Alternative positive and negative spikes, decaying to 0	Auto Regressive (AR) model. pacf() function to be used to identify the order of the model
One or more spikes in series, rest all are 0	Moving Average(MA) model, identify order where plot becomes 0
After a few lags overall a decaying series	Mixed AR & MA model
Total series is 0 or nearly 0	Data is random
Half values at fixed intervals	We need to include seasonal AR term
Visible spikes, no decay to 0	Series is not stationary

```
fitARIMA <- arima(mydata, order=c(1,1,1),seasonal = list(order = c(1,0,0), period = 12),method="ML")

library(lmtest)

coeftest(fitARIMA)
```

```
z test of coefficients:

      Estimate Std. Error  z value Pr(>|z|)
ar1  -0.084865   0.050103  -1.6938   0.0903 .
ma1  -0.954521   0.012413 -76.8943   <2e-16 ***
sar1  0.823202   0.027229  30.2321   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Choosing the best model

R uses maximum likelihood estimation (MLE) to estimate the ARIMA model. It tries to maximize the log-likelihood for given values of p, d, and q when finding parameter estimates so as to maximize the probability of obtaining the data that we have observed.

This is a recursive process and we need to run this arima() function with different (p,d,q) values to find out the most optimized and efficient model.
The output from fitarima() includes the fitted coefficients and the standard error (s.e.) for each coefficient. Observing the coefficients, we can exclude the insignificant ones. We can use a function confint() for this purpose.
We can use a function confint() for this purpose.

```
          2.5 %      97.5 %
ar1  -0.1830652  0.01333424
ma1  -0.9788505 -0.93019085
sar1  0.7698333  0.87657072
```

Forecasting using an ARIMA model

Predict (fitARIMA,n.ahead = 5)

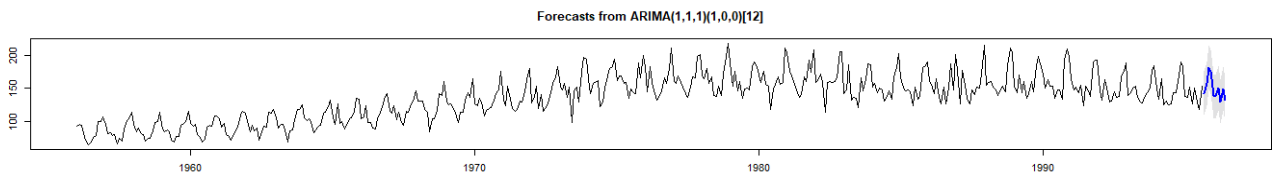
```
$pred
        Jan Feb Mar Apr May Jun Jul Aug      Sep      Oct      Nov      Dec
1995                                    142.2701 157.1751 181.7939 175.2149
1996 138.9934

$se
        Jan Feb Mar Apr May Jun Jul Aug      Sep      Oct      Nov      Dec
1995                                    11.66848 11.67753 11.69142 11.70136
1996 11.71161
```

forecast.Arima() function in the forecast R package can also be used to forecast for future values of the time series. Here we can also specify the confidence level for prediction intervals by using the level argument.

Here we are using 99.5% confidence interval.

futurVal <- forecast (fitARIMA,h=10, level=c(99.5))

plot(futurVal)



In the graph above,the blue color is the forecasted beer production.