ARIMA & SARIMA

Agenda

- Business Problem
- ARIMA Models for Non - Stationary Time Series
- Forecasting ARIMA Models
- Seasonal ARIMA Models
- Forecasting SARIMA Models

ARIMA

Business problem: predict the amount of Carbon Dioxide (in parts per million)

- It is important for nation to develop models that accurately forecast CO2 emission and can take an primitive measures accordingly.

- This model forecast can be used to create premium tables that can assist/guide the nation to control the emission.

Dependent Variable

- The variable we wish to explain or predict

- Usually denoted by Y

- Dependent Variable = Response Variable = Target Variable

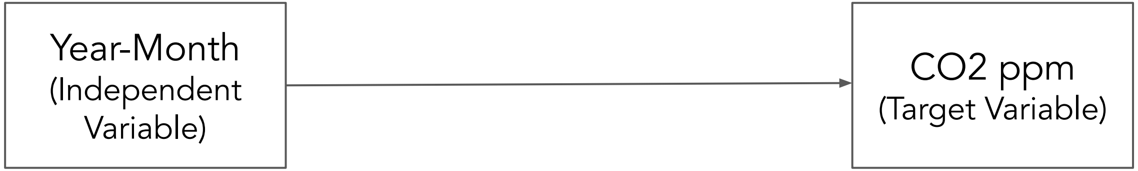
- Here ‘CO2 ppm’ is our target variable

Independent Variable

- The variables used to explain the dependent variable
- Usually denoted by X
- Independent Variable = Predictor Variable
- In our example, Year-Month are the independent variables

Visiting Basics

Variable that contributes to Carbon Dioxide (in parts per million)

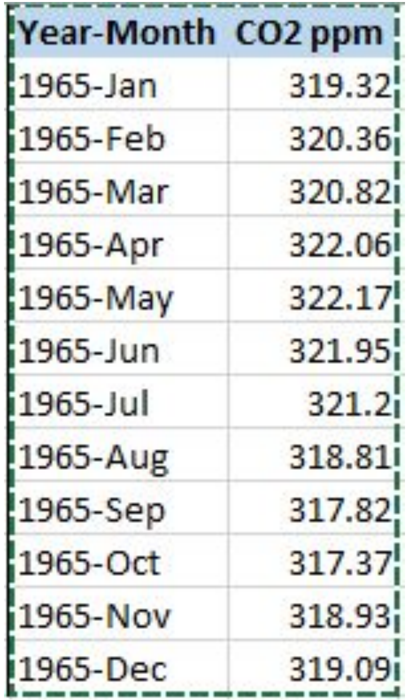


Year-Month (Independent Variable)

CO2 ppm (Target Variable)

Data

Let us consider the following data.



Year-Month	CO2 ppm
1965-Jan	319.32
1965-Feb	320.36
1965-Mar	320.82
1965-Apr	322.06
1965-May	322.17
1965-Jun	321.95
1965-Jul	321.2
1965-Aug	318.81
1965-Sep	317.82
1965-Oct	317.37
1965-Nov	318.93
1965-Dec	319.09

ARIMA Models for Non-stationary Time Series

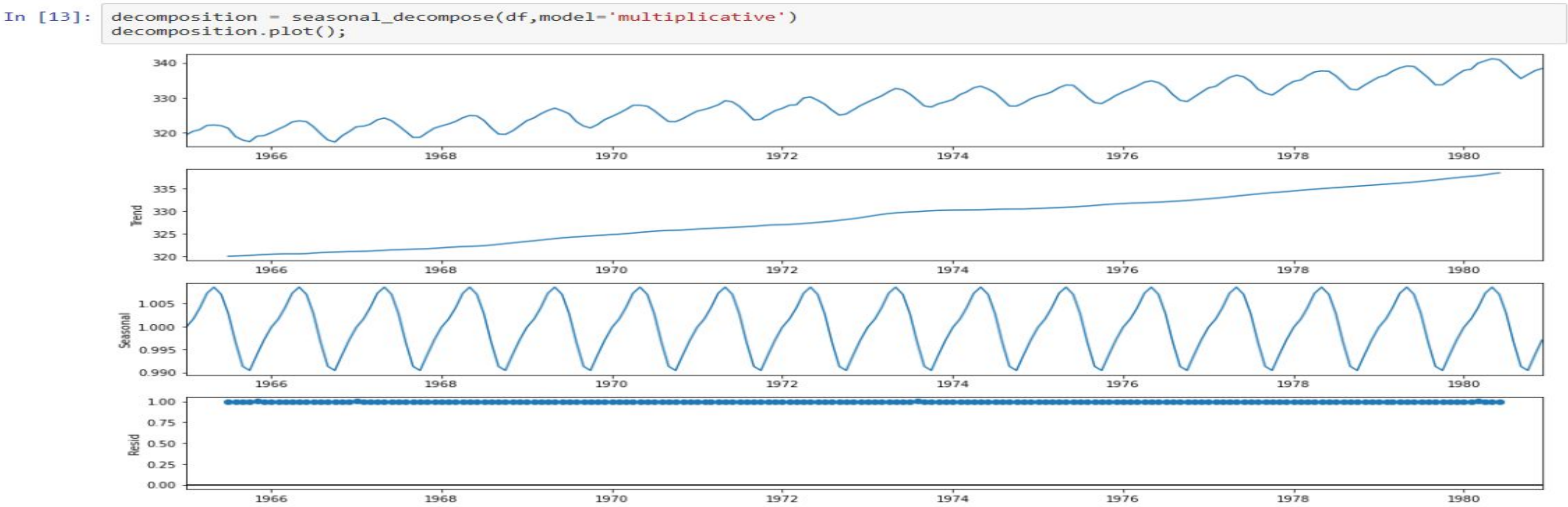
- Stationary Process :-

  - Autoregressive Process: AR (p)
  - Moving Average Process: MA (q)
  - ARMA (p, q)

- Integrated Non-stationary Process :-

  - ARIMA (p, d, q)

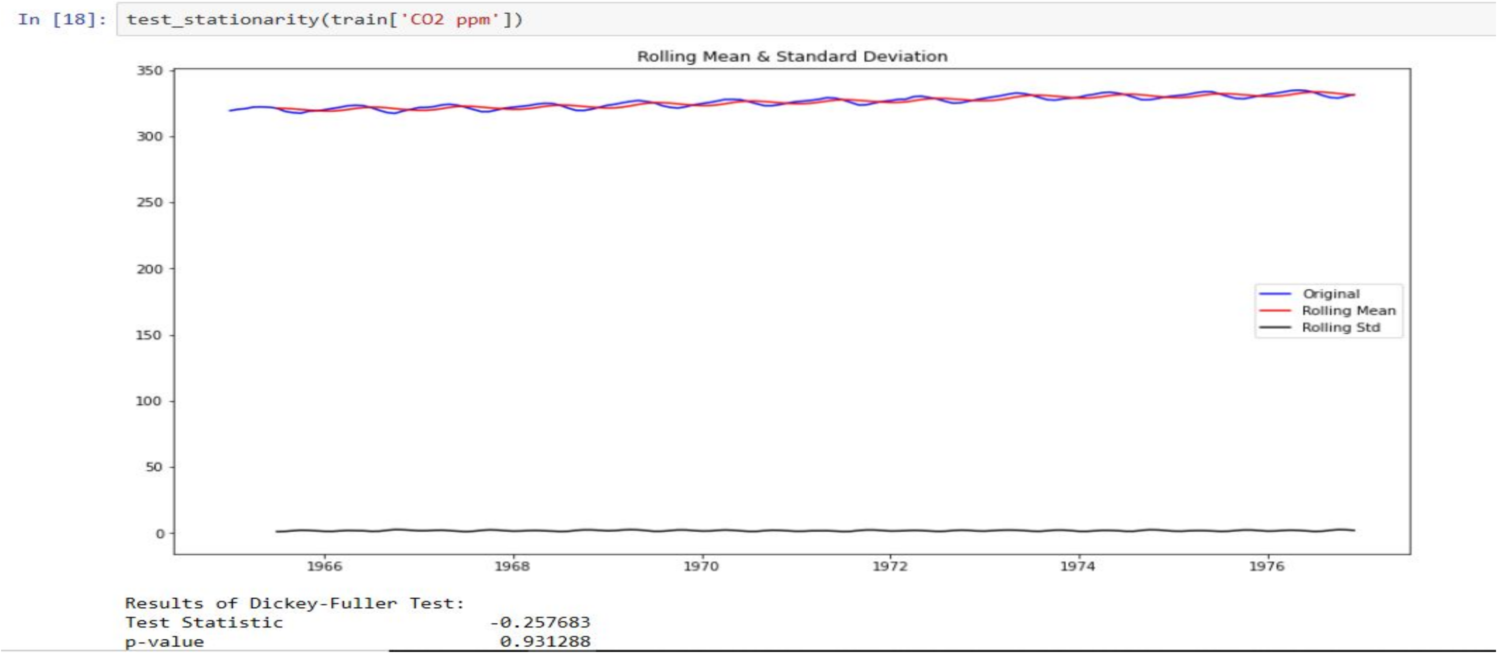
Decomposition of Time-Series Data



```
In [13]: decomposition = seasonal_decompose(df,model='multiplicative')
         decomposition.plot();
```

- We can infer that the existence of trend and seasonality in the data, hence stationary data doesn't have trend, seasonality.

Stationary test - (Dickey-Fuller Test)



```
In [18]: test_stationarity(train['CO2 ppm'])
```

```
Results of Dickey-Fuller Test:
Test Statistic                -0.257683
p-value                        0.931288
```

- We can infer that p-value greater than 0.5, hence null hypothesis rejected the data is non stationary.

ARIMA Model

- ARIMA :- autoregressive integrated moving average. it is an a way of modelling time series data for forecasting or predicting future points in a series.

- ARIMA models consists of three components in it :-

  - AR model

  - Integrated component

  - MA model

ARIMA Model

- A pattern of growth/decline in the data is accounted for (hence the ‘autoregressive’/’AR’ part).
- The rate of change of the growth/decline in the data is accounted for (hence the “integrated”/’I’ part)
- The noise between the consecutive time points is accounted for (hence the “moving average” part)

Note :- time series data is an data that is made up of a sequence of data points taken at successive equally spaced points in time.

ARIMA Model

- Few Key notes to be known about ARIMA models:

  - The ARIMA model is denoted ARIMA as (p , d , q).

  - p : order of AR model.

  - d : times to difference the data.

  - q : order of MA model.

  - p, d, and q are non-negative integers.

ARIMA (d) - Differencing

- Differencing is an non-stationary time series data one or more times that can convert it into stationary. Hence the integrated ' I ' is an component of ARIMA.

- d is an number of times to perform an lag-1 to the difference on data.
  - d = 0: no differencing
  - d = 1: difference of once
  - d = 2: difference of twice

$$Y_i = Z_i - Z_i - 1$$

ARIMA (d) - Differencing

The second component of ARIMA model, where I as ‘integration’ , is used to replace the series with the difference between their current values and the previous values and this differencing process can be performed more than once as per the requirement .

- Equation of first order differencing is : $y_t = y_t - y_{t\text{-}1}$
- Hence, for y_t =2 and $y_{t\text{-}1}$= 1 ; y_t will be 1
- As same, second order differencing , $y_t = (y_t - y_{t\text{-}1}) - (y_{t\text{-}1} - y_{t\text{-}2})$
- Order of this component (order of differencing) is applied by parameter d while fitting a model : ARIMA (p, d, q)

ARIMA (d) - Differencing

- Differencing (lag difference) is the process of transforming a time series to stabilize the mean.

- We have several ways to identify the time series, such as a line plot, which represents the series over time.

- Within these trends, seasonality and random walk can be observed, which is a change over a period of time, and this behavior is considered a nonstationary time series.

- The clear rule states that it needs to remove trends and seasonality in the data from the training time series forecasting model.

ARIMA (d) - Differencing

The following are the reasons for differencing:

- To convert non-stationary data into a stationary time series.
- To remove seasonal trends.

ARIMA (d) - Differencing

FIRST-ORDER DIFFERENCING (TREND DIFFERENCING):

- Change between two consecutive observations in the time series can be written as follows:

  $$Y'_t = Y_t - Y_{t-1}$$

- When the differenced series contains white noise (εt), the formula can be written as follows:

  $$Y_t - Y_{t-1} = \varepsilon_t$$
  $$Y_t = Y_{t-1} + \varepsilon_t$$

  where ε_t = white noise. This is known as a *normal random walk* .
- A random walk represents that a time series is nonstationary.
- This is mostly seen in financial, economic, and microeconomics data. A random walk has mostly long durations of trend ups and down and uncertain and unpredictable changes.

ARIMA (d) - Differencing

FIRST-ORDER DIFFERENCING (TREND DIFFERENCING):

- For instance, the stock price of some XYZ Company has gone up and down for the last six months. This is random walk behavior. Any future moment is not predictable because of the haphazard ups and downs.
- So, here it is clear that the random time series has nonzero mean values. In that case, the final formula is known as *random walk with drift*, as shown here:

$$Y_t = c + Y_{t-1} + \boldsymbol{\varepsilon}_t$$

where c is the average of change between two observations.

- If c is positive, then the average change will rise in Y_t, and Y_t will move upward.
- If c is negative, the *Yt* value will move downward.

ARIMA (d) - Differencing

- Few Key notes to be known about Differencing:
    - First-order differencing in a time series will remove a linear trend (i.e., differences=1).
    - Second-order differencing will remove a quadratic trend (i.e., differences=2).
    - In addition, first-order differencing in a time series at a lag equal to the period will remove a seasonal trend.

ARIMA (d) - Differencing

SECOND-ORDER DIFFERENCING (TREND DIFFERENCING)

- Second-order differencing is a technique used to make first-order differencing data stationary when the first-order differencing has failed.

- So, it’s necessary to apply second-order differencing to obtain a stationary series.

$$Y''_t = Y'_t – Y'_{t-1}$$
$$= (Y_t - Y_{t-1}) – (Y_{t-1} - Y_{t-2})$$
$$= Y_t - 2Y_{t-1} + Y_{t-2}$$

- This is second-order differencing, Y”, which has t-2 values. This is known as *double changes* in the original series.

Distinguish p, d, q values

- The Values of p, q are determine based on the auto-correlation (ACF) and partial autocorrelation (PACF) plots and values of d depends on the level of stationarity in data.

- Where, as in PACF plot the number of peaks indicates the order of the autoregression/AR (value of p in ARIMA(p, d, q)).

- As we can see in the right figure (Next slide), As there is an one peak falling out of range, hence, the order of AR , i.e. value of p would be 1.

- As per the ACF plot, number of peaks indicates the order of the moving average (value of q in ARIMA (p, d, q)) .

Distinguish p, d, q values

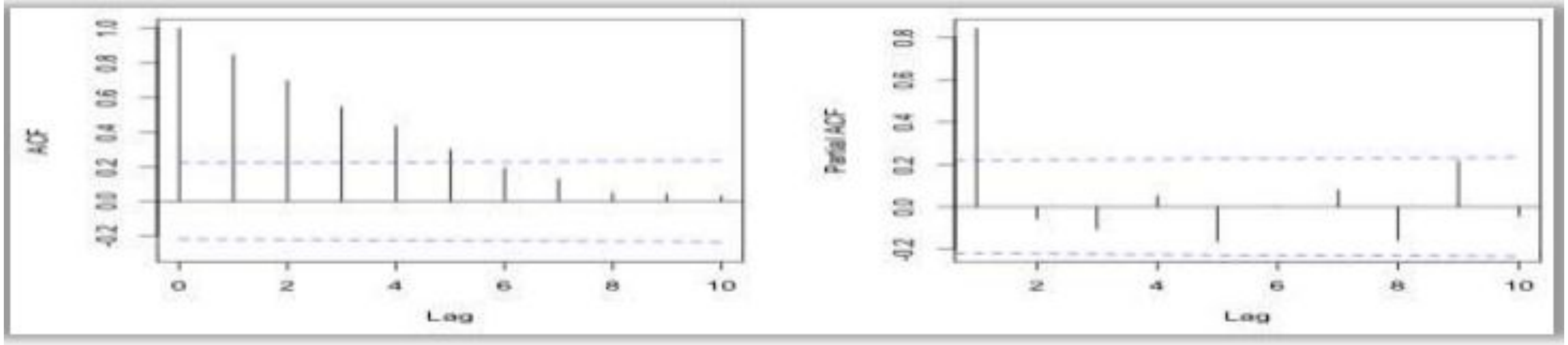
- Where, as in p, d and q parameters in ARIMA (p , d , q) are substituted with integer values where p and q take any values between 0 to 5 and value of d is set between 0 to 2

- For example, ARIMA(2,1,1) means that you have a second order autoregressive model with a first order moving average component and series has been differenced once to induce stationarity

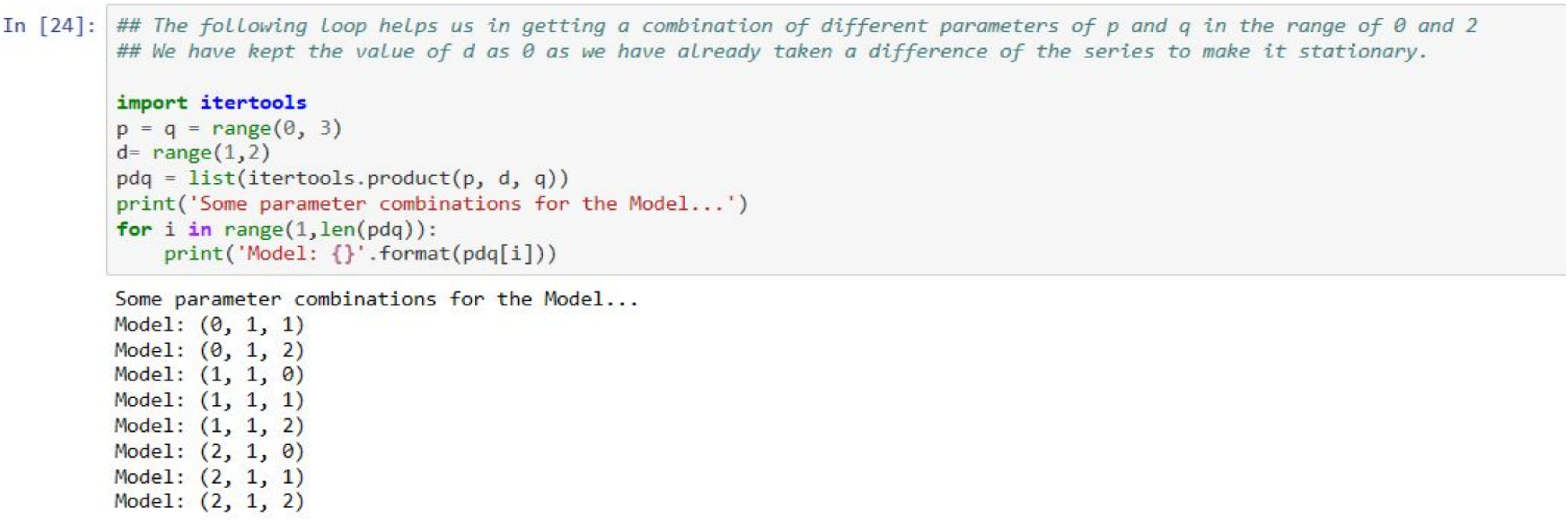
- A value of 0 can be used for any of the above mentioned parameters indicating that particular component (AR/ I/ MA) should not be used. This way, the ARIMA model can be configured to perform the function of an ARMA model, and even a simple AR, I, or MA model depending on the data

Distinguish p, d, q values

- We can observe that in the left figure there was five raised peaks falling out of range, hence, the order of MA i.e. value of q would be 5.



Parameter combinations of p, d, q values of the Model



In [24]:

```
## The following loop helps us in getting a combination of different parameters of p and q in the range of 0 and 2
## We have kept the value of d as 0 as we have already taken a difference of the series to make it stationary.

import itertools
p = q = range(0, 3)
d= range(1,2)
pdq = list(itertools.product(p, d, q))
print('Some parameter combinations for the Model...')
for i in range(1,len(pdq)):
    print('Model: {}'.format(pdq[i]))
```

```
Some parameter combinations for the Model...
Model: (0, 1, 1)
Model: (0, 1, 2)
Model: (1, 1, 0)
Model: (1, 1, 1)
Model: (1, 1, 2)
Model: (2, 1, 0)
Model: (2, 1, 1)
Model: (2, 1, 2)
```

Akaike Information Criterion (AIC)

- The best fit model is selected based on Akaike Information Criterion (AIC) , and Bayesian Information Criterion (BIC) values. The idea is to choose a model with minimum AIC and BIC values.

- AIC is an effort to balance the model between goodness-of-fit and number of parameters used in the model, This is similar to the balancing act between income and cost of a company so that the debs of the company is optimized (Debt = Cost - Income).

- As a modeler, we care about the maximum goodness of fit (income) with the minimum number of parameters (cost).

$$AIC = 2K - 2\ln(L)$$

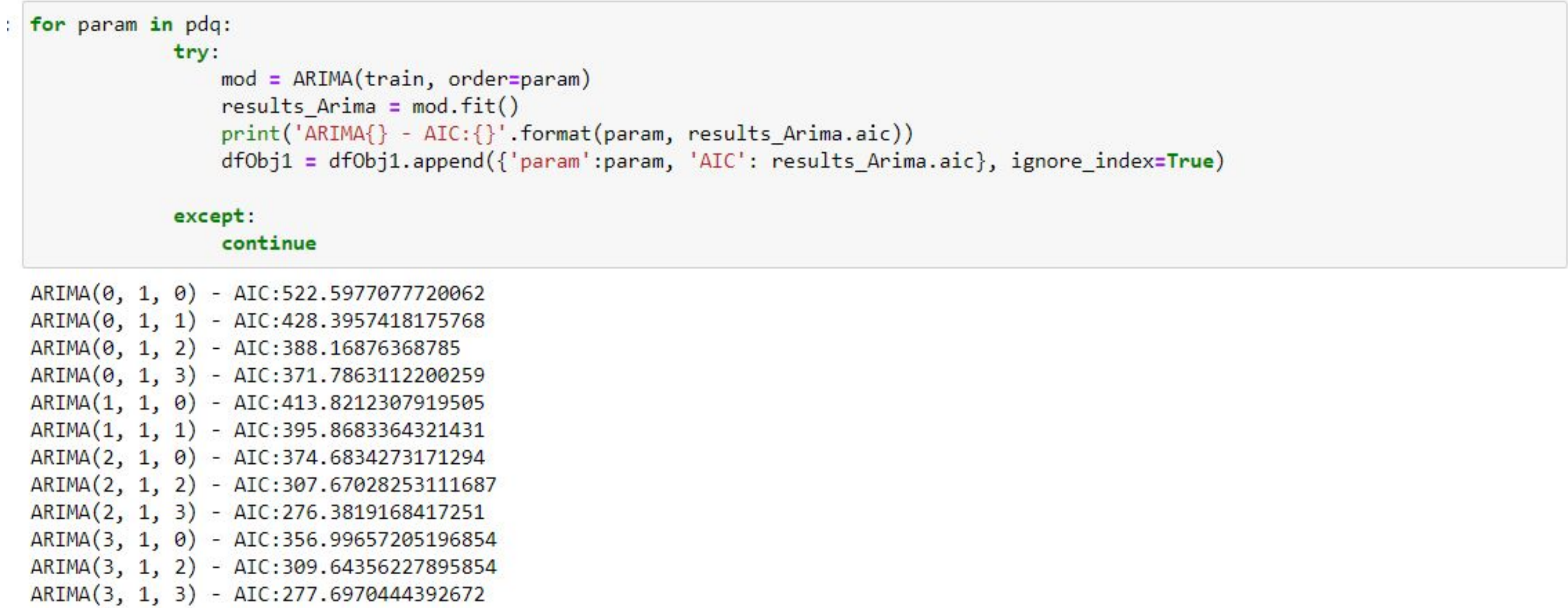
Akaike Information Criterion (AIC)

- For the given model, L in the above formula is the maximized value of the likelihood function representing goodness-of-fit, and K the number of estimated parameters. Like our debts, we want to keep AIC value at the minimum to choose the best possible model.

- Bayesian Information Criterion (BIC) is another variant of AIC and is used for the same purpose of best fit model selection. For the best possible model selection, we want to look at AIC, BIC, and AICc (AIC with sample correction) if all these values are minimum for a given model.

- With increasing parameters K will increase and hence AIC increases. While with the goodness of the fit L increases thus decreasing AIC.

Best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC)

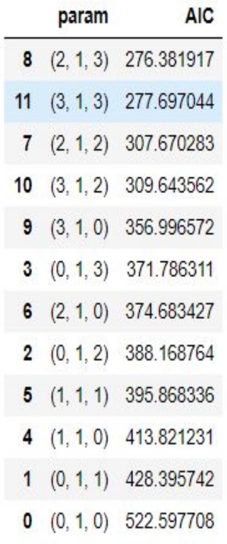


```
for param in pdq:
            try:
                mod = ARIMA(train, order=param)
                results_Arima = mod.fit()
                print('ARIMA{} - AIC:{}'.format(param, results_Arima.aic))
                dfObj1 = dfObj1.append({'param':param, 'AIC': results_Arima.aic}, ignore_index=True)

            except:
                continue
```

```
ARIMA(0, 1, 0) - AIC:522.5977077720062
ARIMA(0, 1, 1) - AIC:428.3957418175768
ARIMA(0, 1, 2) - AIC:388.16876368785
ARIMA(0, 1, 3) - AIC:371.7863112200259
ARIMA(1, 1, 0) - AIC:413.8212307919505
ARIMA(1, 1, 1) - AIC:395.8683364321431
ARIMA(2, 1, 0) - AIC:374.6834273171294
ARIMA(2, 1, 2) - AIC:307.67028253111687
ARIMA(2, 1, 3) - AIC:276.3819168417251
ARIMA(3, 1, 0) - AIC:356.99657205196854
ARIMA(3, 1, 2) - AIC:309.64356227895854
ARIMA(3, 1, 3) - AIC:277.6970444392672
```

Best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC)

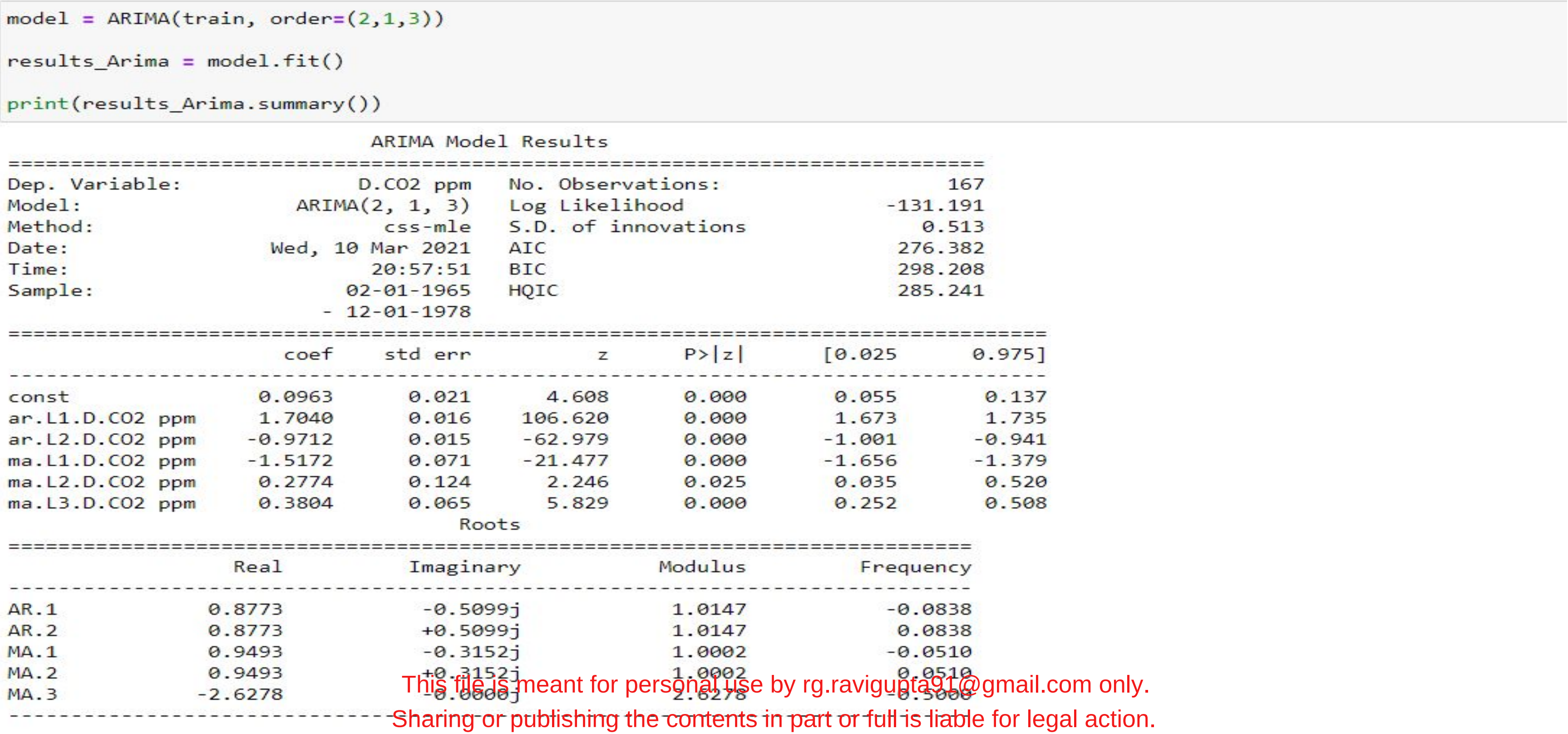
- We can interpret that the parameters (p,d,q) (2,1,3) are the optimal with the lowest AIC to treat the non stationary data and predict the time-series data.



	param	AIC
8	(2, 1, 3)	276.381917
11	(3, 1, 3)	277.697044
7	(2, 1, 2)	307.670283
10	(3, 1, 2)	309.643562
9	(3, 1, 0)	356.996572
3	(0, 1, 3)	371.786311
6	(2, 1, 0)	374.683427
2	(0, 1, 2)	388.168764
5	(1, 1, 1)	395.868336
4	(1, 1, 0)	413.821231
1	(0, 1, 1)	428.395742
0	(0, 1, 0)	522.597708

Model Evaluation

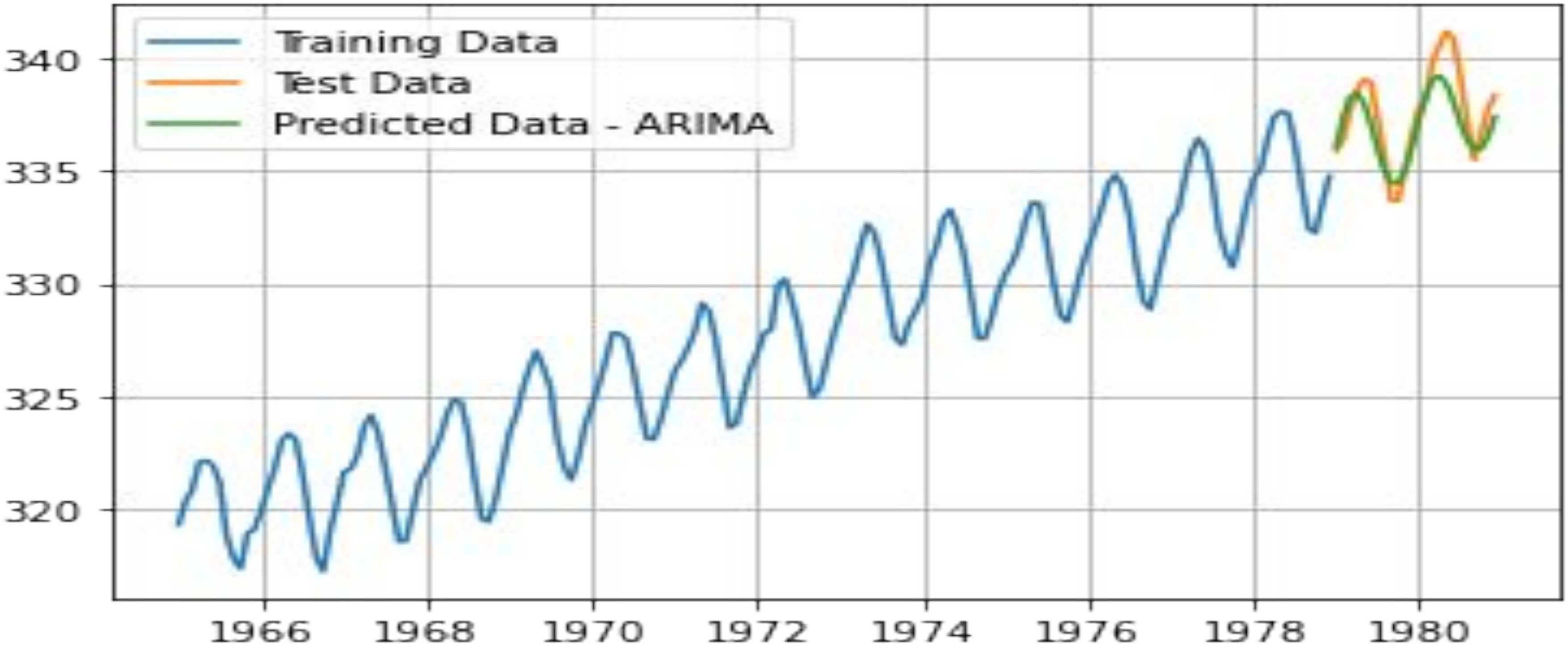
Model Evaluation



```
model = ARIMA(train, order=(2,1,3))

results_Arima = model.fit()

print(results_Arima.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:              D.CO2 ppm   No. Observations:                  167
Model:                 ARIMA(2, 1, 3)   Log Likelihood                -131.191
Method:                       css-mle   S.D. of innovations              0.513
Date:                Wed, 10 Mar 2021   AIC                            276.382
Time:                        20:57:51   BIC                            298.208
Sample:                    02-01-1965   HQIC                           285.241
                         - 12-01-1978
===================================================================================
                      coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------
const               0.0963      0.021      4.608      0.000       0.055       0.137
ar.L1.D.CO2 ppm     1.7040      0.016    106.620      0.000       1.673       1.735
ar.L2.D.CO2 ppm    -0.9712      0.015    -62.979      0.000      -1.001      -0.941
ma.L1.D.CO2 ppm    -1.5172      0.071    -21.477      0.000      -1.656      -1.379
ma.L2.D.CO2 ppm     0.2774      0.124      2.246      0.025       0.035       0.520
ma.L3.D.CO2 ppm     0.3804      0.065      5.829      0.000       0.252       0.508
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.8773           -0.5099j            1.0147           -0.0838
AR.2            0.8773           +0.5099j            1.0147            0.0838
MA.1            0.9493           -0.3152j            1.0002           -0.0510
MA.2            0.9493           +0.3152j            1.0002            0.0510
MA.3           -2.6278           -0.0000j            2.6278           -0.5000
-----------------------------------------------------------------------------
```

Model Evaluation



Limitations of ARIMA

- For an ARIMA model, we can see the predictions(previous slide) with 95% confidence interval bands. The seasonality was unable to be captured. Let us try out a SARIMA model.

- Seasonality in a time series data can be captured with SARIMA Model.

Summary of ARIMA

- ARIMA is a method among several used for forecasting univariate variables, which uses information obtained from the variable itself to predict its trend. The variables are regressed on its own past values.

- AR(p) is where p equals the order of autocorrelation (designates weighted moving average over past observations) z I (d), where d is the order of integration (differencing), which indicates linear trend or polynomial trend z.

- MA(q) is where q equals the order of moving averages (designates weighted moving average over past errors).

- ARIMA is made up of two models: AR and MA.

SARIMA

SARIMA Model

- In general the economic, agricultural and geophysical time series have cycle components within a specific time period.
- The smallest time period for this repetitive phenomenon is called a seasonal period (s).
- For example,

  monthly temperature have a 12-month cycle, s = 12.

  the quarterly ice cream sales have a 4-quarterly cycle, s = 4.

  It may be useful to use a s-fold difference operator

  with s = 4 to remove the cycle component from quarterly data,

  s = 12 to remove annual fluctuations from monthly data.

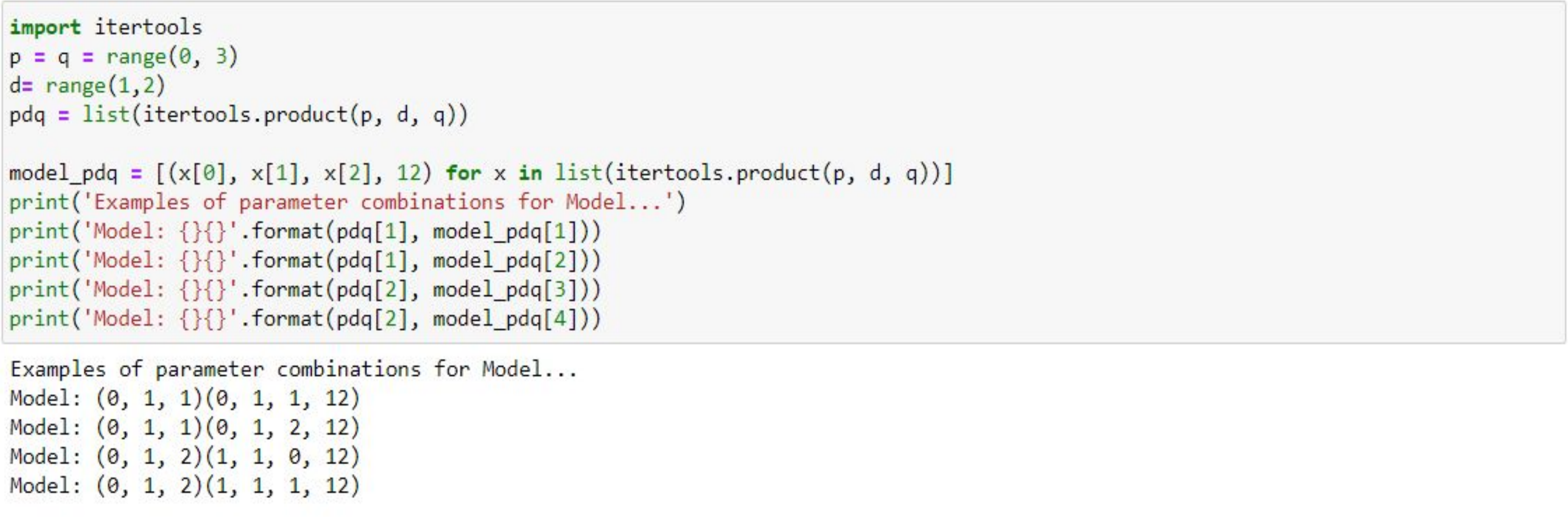
SARIMA Model

- The ARIMA models can be extended to handle seasonal components of a data series.

- The multiplicative seasonal autoregressive moving average model, SARIMA (p, d, q)(P, D, Q)s is given by where { }is Gaussian white noise is ordinary autoregressive and moving average components; and are seasonal autoregressive and moving average components, respectively, and are the ordinary and seasonal difference component of order d and D.

- ARIMA(p, d, q)x(P, D, Q)[freq]
- Seasonal difference = D
- Appropriate for seasonal series

Parameter combinations of (p, d, q) x (P, D, Q)[frequency/s] values of the Model

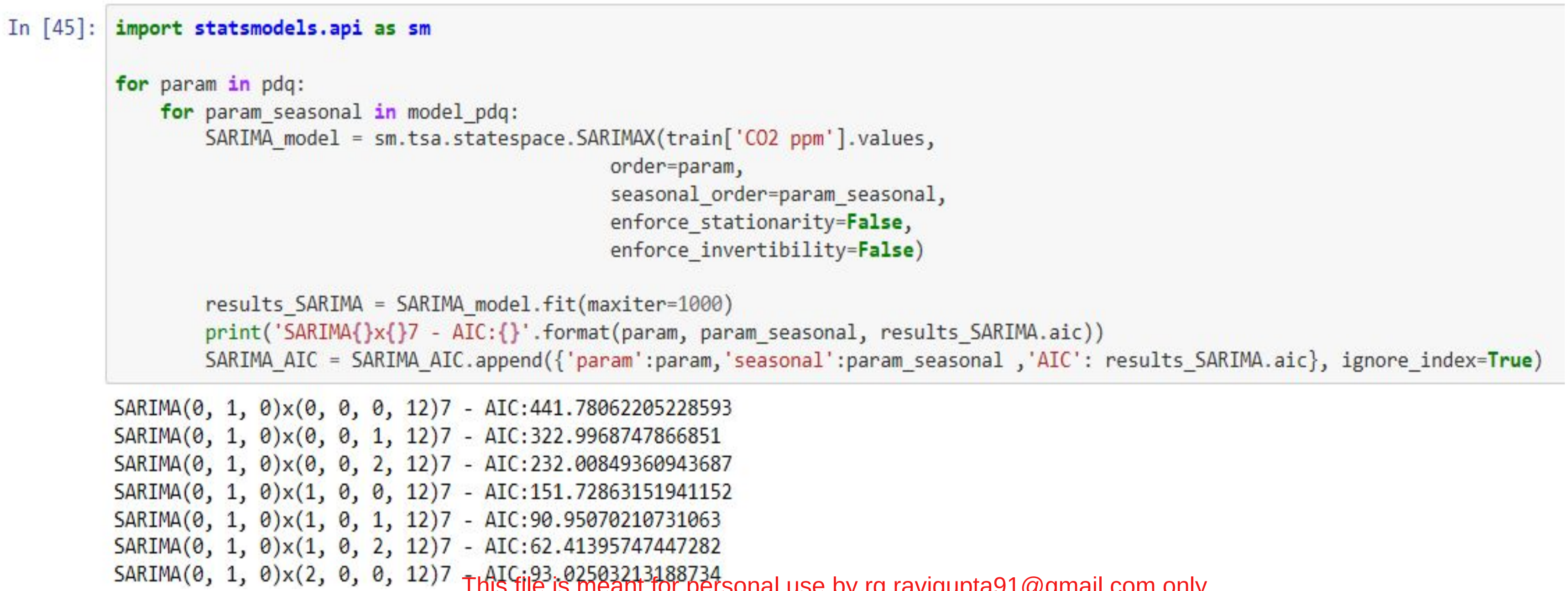


```
import itertools
p = q = range(0, 3)
d= range(1,2)
pdq = list(itertools.product(p, d, q))

model_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
print('Examples of parameter combinations for Model...')
print('Model: {}{}'.format(pdq[1], model_pdq[1]))
print('Model: {}{}'.format(pdq[1], model_pdq[2]))
print('Model: {}{}'.format(pdq[2], model_pdq[3]))
print('Model: {}{}'.format(pdq[2], model_pdq[4]))
```

```
Examples of parameter combinations for Model...
Model: (0, 1, 1)(0, 1, 1, 12)
Model: (0, 1, 1)(0, 1, 2, 12)
Model: (0, 1, 2)(1, 1, 0, 12)
Model: (0, 1, 2)(1, 1, 1, 12)
```

Hence, as per the business problem we have a data for an year-wise split which composed of months .so, our frequency or (s) is 12

Best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC)

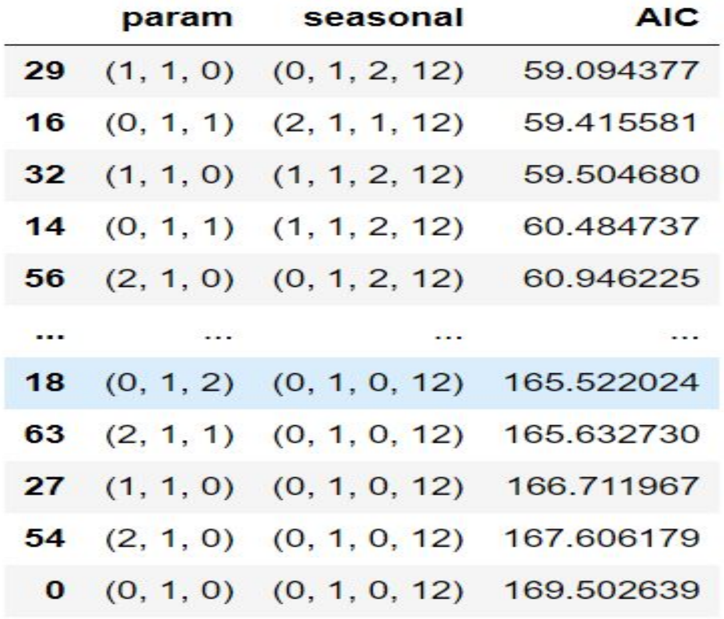


```
In [45]: import statsmodels.api as sm

         for param in pdq:
             for param_seasonal in model_pdq:
                 SARIMA_model = sm.tsa.statespace.SARIMAX(train['CO2 ppm'].values,
                                                         order=param,
                                                         seasonal_order=param_seasonal,
                                                         enforce_stationarity=False,
                                                         enforce_invertibility=False)

                 results_SARIMA = SARIMA_model.fit(maxiter=1000)
                 print('SARIMA{}x{}7 - AIC:{}'.format(param, param_seasonal, results_SARIMA.aic))
                 SARIMA_AIC = SARIMA_AIC.append({'param':param,'seasonal':param_seasonal ,'AIC': results_SARIMA.aic}, ignore_index=True)
```

```
SARIMA(0, 1, 0)x(0, 0, 0, 12)7 - AIC:441.78062205228593
SARIMA(0, 1, 0)x(0, 0, 1, 12)7 - AIC:322.9968747866851
SARIMA(0, 1, 0)x(0, 0, 2, 12)7 - AIC:232.00849360943687
SARIMA(0, 1, 0)x(1, 0, 0, 12)7 - AIC:151.72863151941152
SARIMA(0, 1, 0)x(1, 0, 1, 12)7 - AIC:90.95070210731063
SARIMA(0, 1, 0)x(1, 0, 2, 12)7 - AIC:62.41395747447282
SARIMA(0, 1, 0)x(2, 0, 0, 12)7 - AIC:93.02503213188734
```

Best parameters are selected in accordance with the lowest Akaike Information Criteria (AIC)

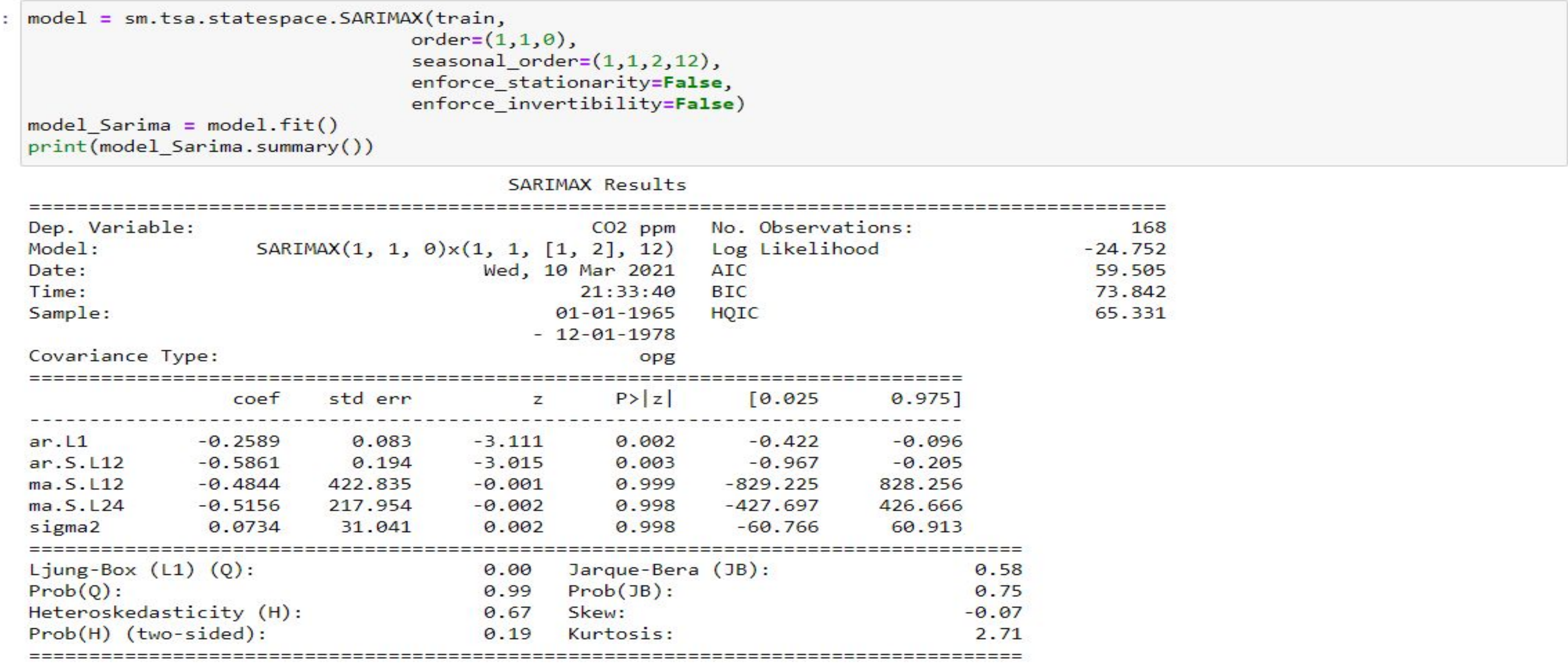
- We can interpret that the parameters (p,d,q)x(P,D,Q)[s] (1,1,0) (1,0,2,12) are the optimal with the lowest AIC to treat the non stationary data and predict the time-series data.



	param	seasonal	AIC
29	(1, 1, 0)	(0, 1, 2, 12)	59.094377
16	(0, 1, 1)	(2, 1, 1, 12)	59.415581
32	(1, 1, 0)	(1, 1, 2, 12)	59.504680
14	(0, 1, 1)	(1, 1, 2, 12)	60.484737
56	(2, 1, 0)	(0, 1, 2, 12)	60.946225
...	...	...	...
18	(0, 1, 2)	(0, 1, 0, 12)	165.522024
63	(2, 1, 1)	(0, 1, 0, 12)	165.632730
27	(1, 1, 0)	(0, 1, 0, 12)	166.711967
54	(2, 1, 0)	(0, 1, 0, 12)	167.606179
0	(0, 1, 0)	(0, 1, 0, 12)	169.502639

Model Evaluation

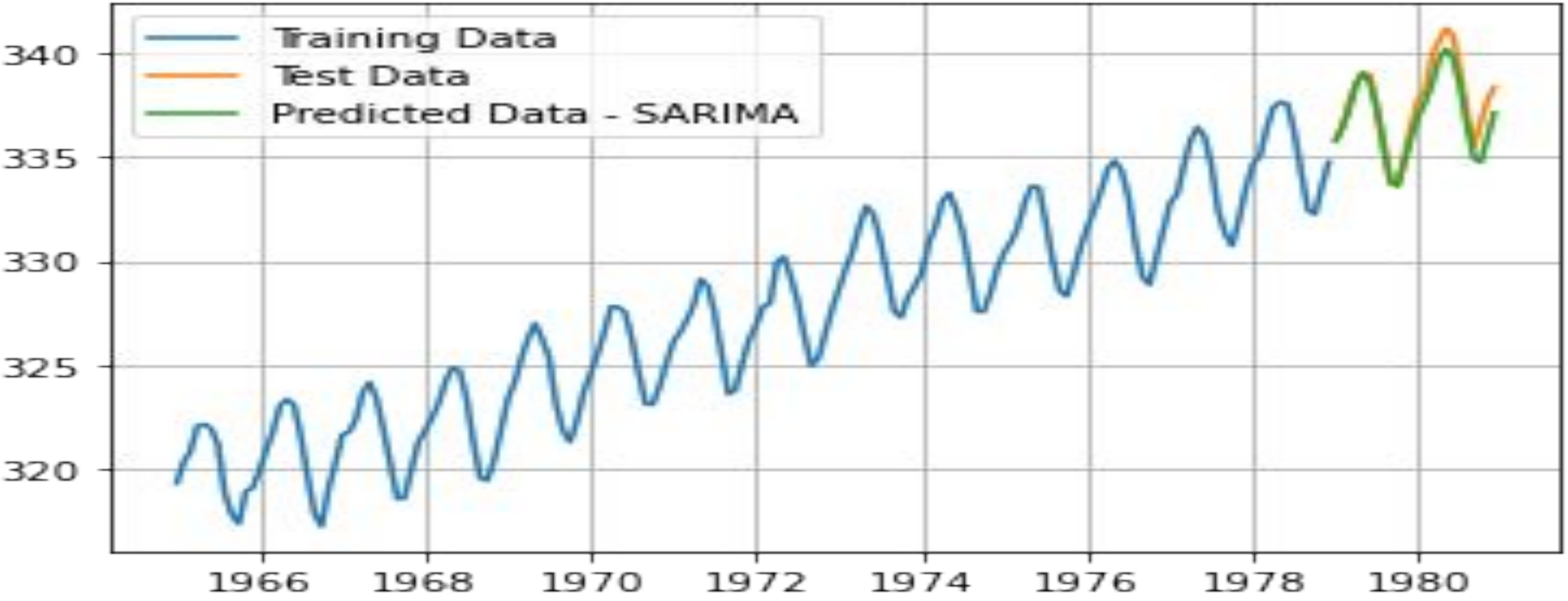
Model Evaluation



```
: model = sm.tsa.statespace.SARIMAX(train,
                                    order=(1,1,0),
                                    seasonal_order=(1,1,2,12),
                                    enforce_stationarity=False,
                                    enforce_invertibility=False)
  model_Sarima = model.fit()
  print(model_Sarima.summary())
```

```
                                     SARIMAX Results
==============================================================================================
Dep. Variable:                           CO2 ppm   No. Observations:                       168
Model:             SARIMAX(1, 1, 0)x(1, 1, [1, 2], 12)   Log Likelihood                    -24.752
Date:                           Wed, 10 Mar 2021   AIC                                  59.505
Time:                                   21:33:40   BIC                                  73.842
Sample:                               01-01-1965   HQIC                                 65.331
                                    - 12-01-1978
Covariance Type:                             opg
=============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------
ar.L1         -0.2589      0.083     -3.111      0.002      -0.422      -0.096
ar.S.L12      -0.5861      0.194     -3.015      0.003      -0.967      -0.205
ma.S.L12      -0.4844    422.835     -0.001      0.999    -829.225     828.256
ma.S.L24      -0.5156    217.954     -0.002      0.998    -427.697     426.666
sigma2         0.0734     31.041      0.002      0.998     -60.766      60.913
==================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 0.58
Prob(Q):                              0.99   Prob(JB):                         0.75
Heteroskedasticity (H):               0.67   Skew:                            -0.07
Prob(H) (two-sided):                  0.19   Kurtosis:                         2.71
==================================================================================
```

Model Evaluation



Model Evaluation

Model	RMSE	MAPE
ARIMA(2,1,3)	1.17	0.28
SARIMA(1,1,0)(0,1,2,12)	0.81	0.18

Model Evaluation

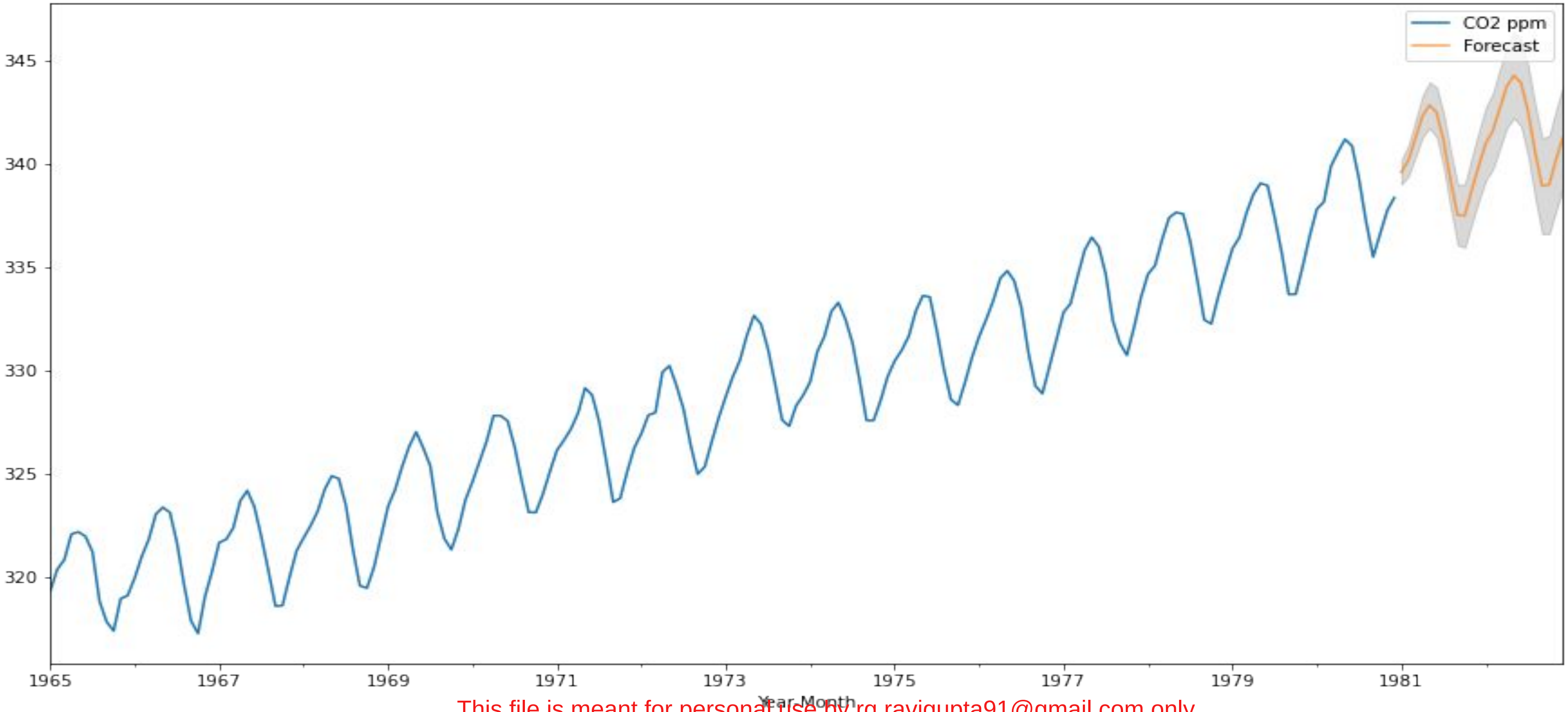
- From an SARIMA model, we can see the predictions. The seasonality was been captured.

- The evaluation metric for Autoregression models are MAPE.

- Hence, the data consist of seasonality we are able to secure the least/optimal MAPE in SARIMA model.

- It’s been evident accordingly such a way when to be used ARIMA ,SARIMA Models.

Actual forecast

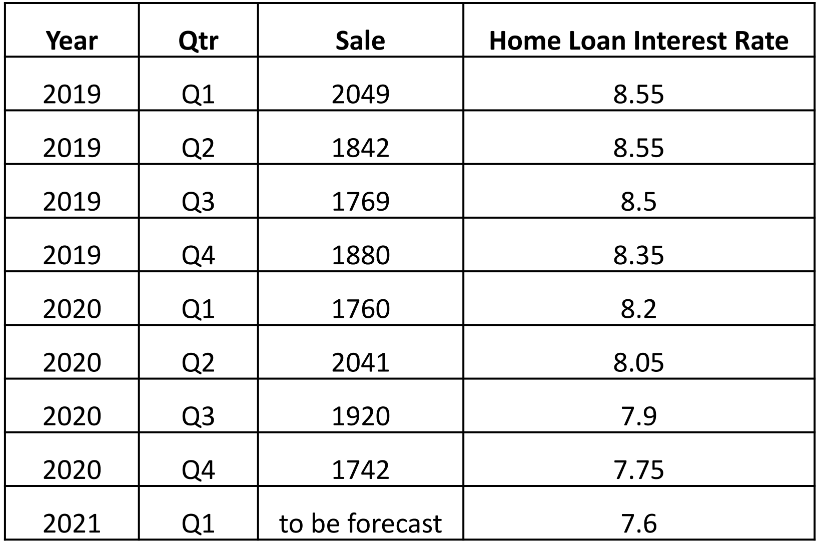


Summary

- Based on accuracy choose the model which works best
- Must have proper interpretability
- Often a simple model works better

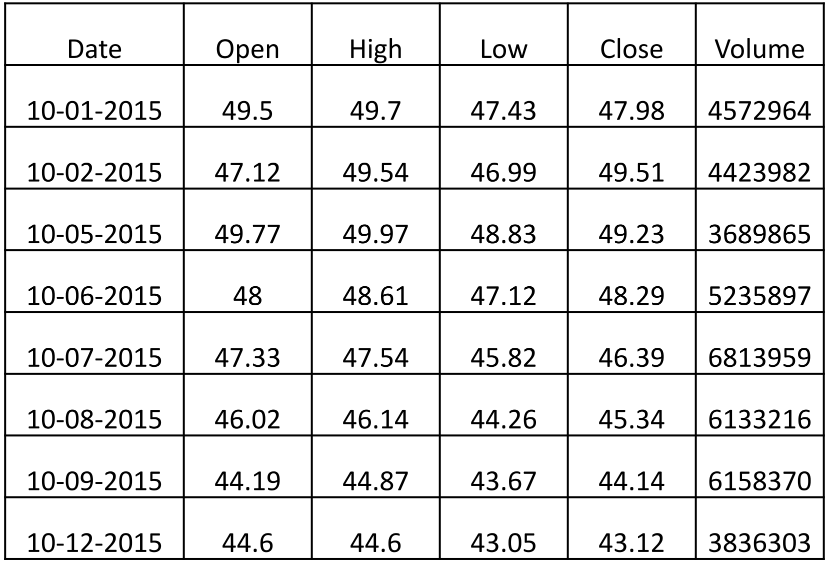
Further study

- Property sale data



Year	Qtr	Sale	Home Loan Interest Rate
2019	Q1	2049	8.55
2019	Q2	1842	8.55
2019	Q3	1769	8.5
2019	Q4	1880	8.35
2020	Q1	1760	8.2
2020	Q2	2041	8.05
2020	Q3	1920	7.9
2020	Q4	1742	7.75
2021	Q1	to be forecast	7.6

- Stock price data



Date	Open	High	Low	Close	Volume
10-01-2015	49.5	49.7	47.43	47.98	4572964
10-02-2015	47.12	49.54	46.99	49.51	4423982
10-05-2015	49.77	49.97	48.83	49.23	3689865
10-06-2015	48	48.61	47.12	48.29	5235897
10-07-2015	47.33	47.54	45.82	46.39	6813959
10-08-2015	46.02	46.14	44.26	45.34	6133216
10-09-2015	44.19	44.87	43.67	44.14	6158370
10-12-2015	44.6	44.6	43.05	43.12	3836303

Appendix

Akaike's Information Criterion:-

Akaike's information criterion (AIC) is known in the statistics trade as a **penalized log-likelihood**. If you have a model for which a log-likelihood value can be obtained, then

$$AIC = -2 \times \log\text{-likelihood} + 2(p+1),$$

where *p* is the number of parameters in the model, and 1 is added for the estimated variance (you could call this another parameter if you wanted to). To demystify AIC let's calculate it by hand. We revisit the regression data for which we calculated the log-likelihood.

Appendix

Akaike's Information Criterion:-

```
attach(regression)
names(regression)
[1] "speed"  "time"
growth

[1] 12 10 8 11 6 7 2 3 3
```

The are nine values of the response variable, growth, and we calculated the log-likelihood as −23.98941 earlier. There was only one parameter estimated from the data for these calculations (the mean value of *y*), so *p* = 1. This means that AIC should be

$$AIC = -2 \times -23.98941 + 2 \times (1+1) = 51.97882.$$

Appendix

Akaike's Information Criterion:-

Fortunately, we do not need to carry out these calculations, because there is a built-in function for calculating AIC. It takes a model object as its argument, so we need to fit a one-parameter model to the speed data like this:

```
model<-lm(speed~1)
```

Then we can get the AIC directly:

```
AIC(model)
[1] 51.97882
```

Appendix

AIC AS A MEASURE OF THE FIT OF A MODEL:

- The more parameters that there are in the model, the better the fit. You could obtain a perfect fit if you had a separate parameter for every data point, but this model would have absolutely no explanatory power.
- There is always going to be a trade-off between the goodness of fit and the number of parameters required by parsimony. AIC is useful because it explicitly penalizes any superfluous parameters in the model, by adding $2(p + 1)$ to the deviance.
- When comparing two models, the smaller the AIC, the better the fit. This is the basis of automated model simplification using step.
- You can use the function AIC to compare two models, in exactly the same way as you can use anova. Here we develop an analysis of covariance.

Appendix

AIC AS A MEASURE OF THE FIT OF A MODEL:

```
model.1<-lm(Car ~ Grazing*Root)

model.2<-lm(Car ~ Grazing+Root)

AIC(model.1, model.2)

        df   AIC

model.1  5   273.0135

model.2  4   271.1279
```

Appendix

AIC AS A MEASURE OF THE FIT OF A MODEL:

- Because model.2 has the *lower* AIC, we prefer it to model. l. The log-likelihood was penalized by 2 × (4 + 1) = 10 in model 1 because that model contained 4 parameters (2 slopes and 2 intercepts) and by 2 × (3 + 1) = 8 in model.2 because that model had 3 parameters (two intercepts and a common slope).
- You can see where the two values of AIC come from by calculation:

```
-2*logLik(model.1)+2*(4+1)
[1] 273.0135
-2*logLik(model.2)+2*(3+1)
[1] 271.1279
```

References

Aitkin, M., Anderson, D., Francis, B. and Hinde, J. (1989) *Statistical Modelling in GlIM.* Oxford: Clarendon Press.

Atkinson, A.C. (1985) *Plots, Transformations, and Regression*. Oxford: Clarendon Press.

Box, G.E.P. and Jenkins, G.M. (1976) *Time Series Analysis: Forecasting and Control*. Oakland, CA: Holden-Day.

Hicks, C.R. (1973) *Fundamental Concepts in the Design of Experiments*. New York, Holt: Rinehart and Winston.

Johnson, N.L. and Kotz, S. (1970) *Continuous Univariate Distributions. Volume 2.* New York: John Wiley.

Priestley, M.B. (1981) *Spectral Analysis and Time Series*. London: Academic Press.