Forecasting

ARIMA, SARIMA and MARKOV CHAIN MODEL

Bipin karki - bipinkarki.nep@gmail.com

20/10/2019

1. BJSALES Dataset

BJsales is a time series sales dataset that has 150 observations. The following is the plot of BJsales.

```
plot(BJsales, col = "dark green", xlab = "Observations", ylab = "BJSALES",
     main = "BJsales dataset", xaxt='n', lwd=2)+
axis(1, seq(0,150,5))+
abline(v=seq(0,150,5), lty=3, col="gray")
```

data-1.bb

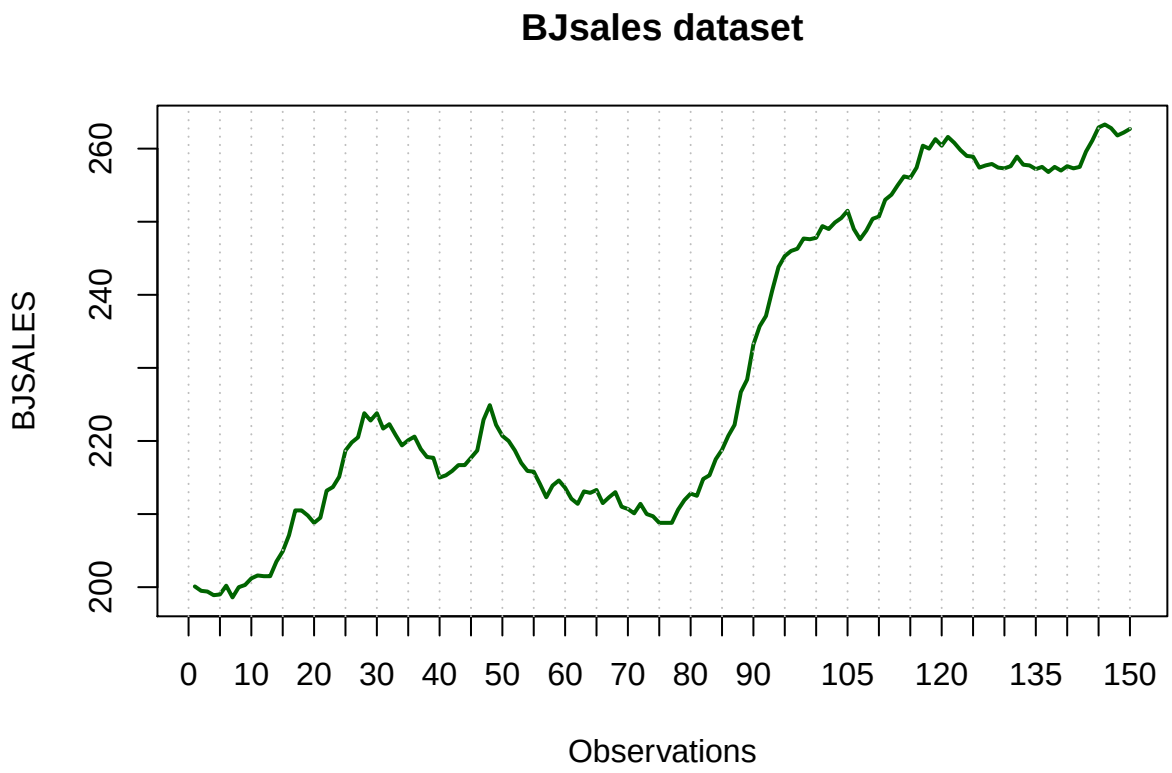

Figure 1: BJsales dataset

```
## numeric(0)
```

From the plot we can observe that the mean is changing over time and the time series is not statioary. We can see the change point at 78th observation is significant and considering dataset after 78th observation and observe the plot.

```
BJsales_data = window(BJsales, start=78)
plot(BJsales_data, col = "dark green", xlab = "Observations", ylab = "BJSALES",
     main = "BJsales dataset", xaxt='n', lwd=2, type="o")+
axis(1, seq(0,150,5))+
abline(v=seq(0,150,5), lty=3, col="gray")
```

data change point-1.bb

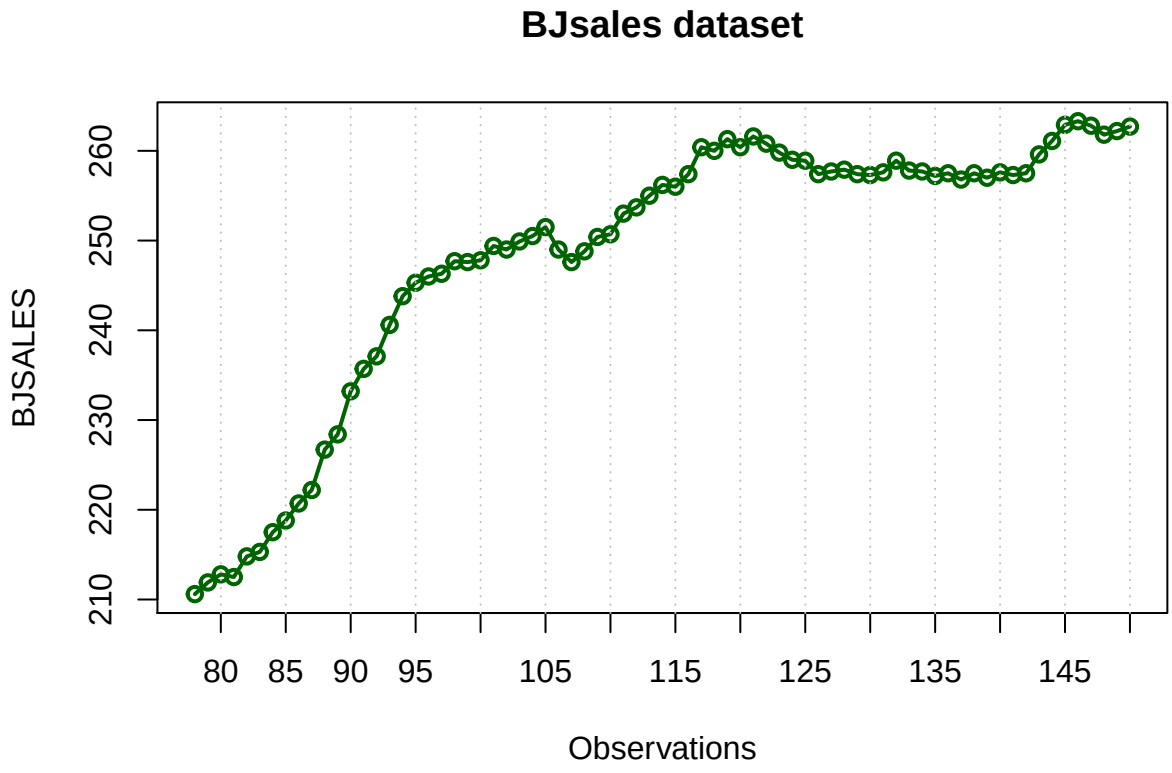

Figure 2: BJsales dataset after change point

```
## numeric(0)
```

From the plot we can see that the dataset has upward trend and not staionary. Lets take a first order difference and observe if we can acheive the stationarity of the data.

```
plot(diff(BJsales_data), col = "dark blue", xlab = "BJsales", ylab = "Observations",
     main = "First order difference
of BJsales dataset", lwd=2, type="o")
```

order difference-1.bb

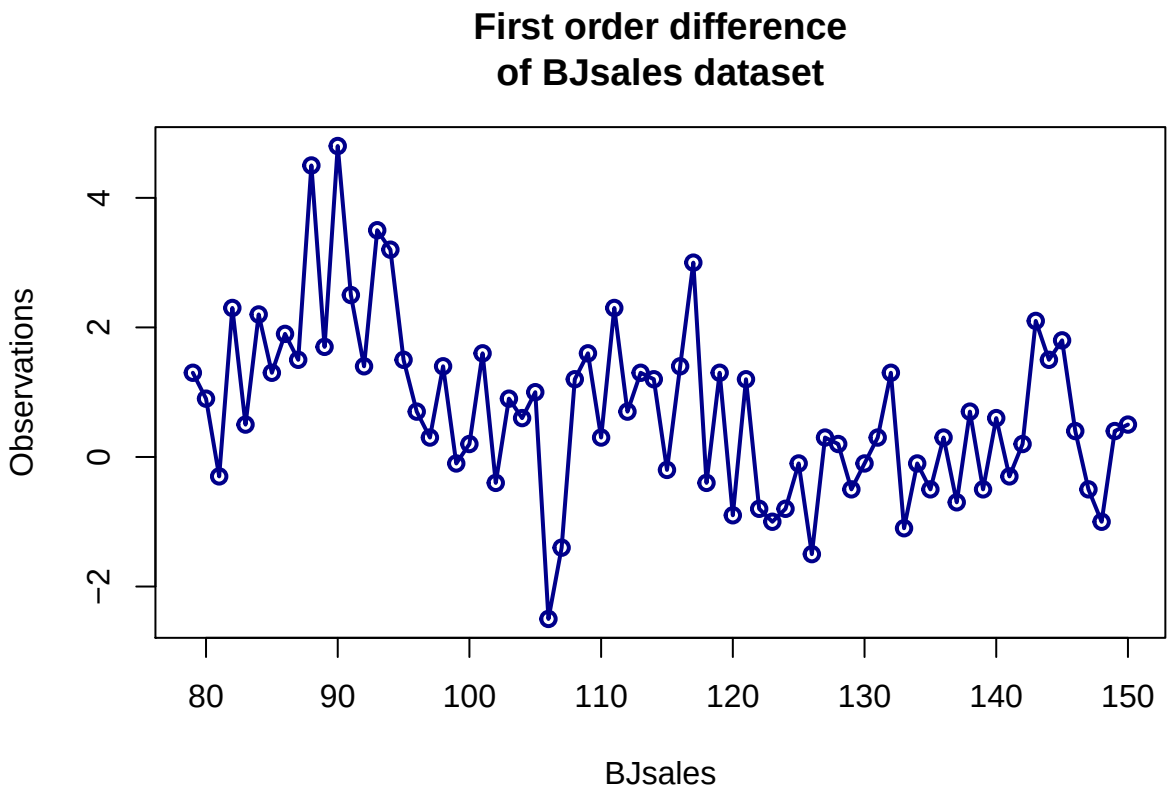

Figure 3: First order difference of BJsales dataset

We can observe that the time series plot shown above is still not staionary. Further taking seond order difference of the dataset to make it stationary.

```
par(mfrow = c(1, 1))
plot(diff(diff(BJsales_data)), col = "dark blue", xlab = "BJsales", ylab = "Observations",
     main = "Second order difference
of BJsales dataset", lwd=2, type="o")
```

order difference-1.bb

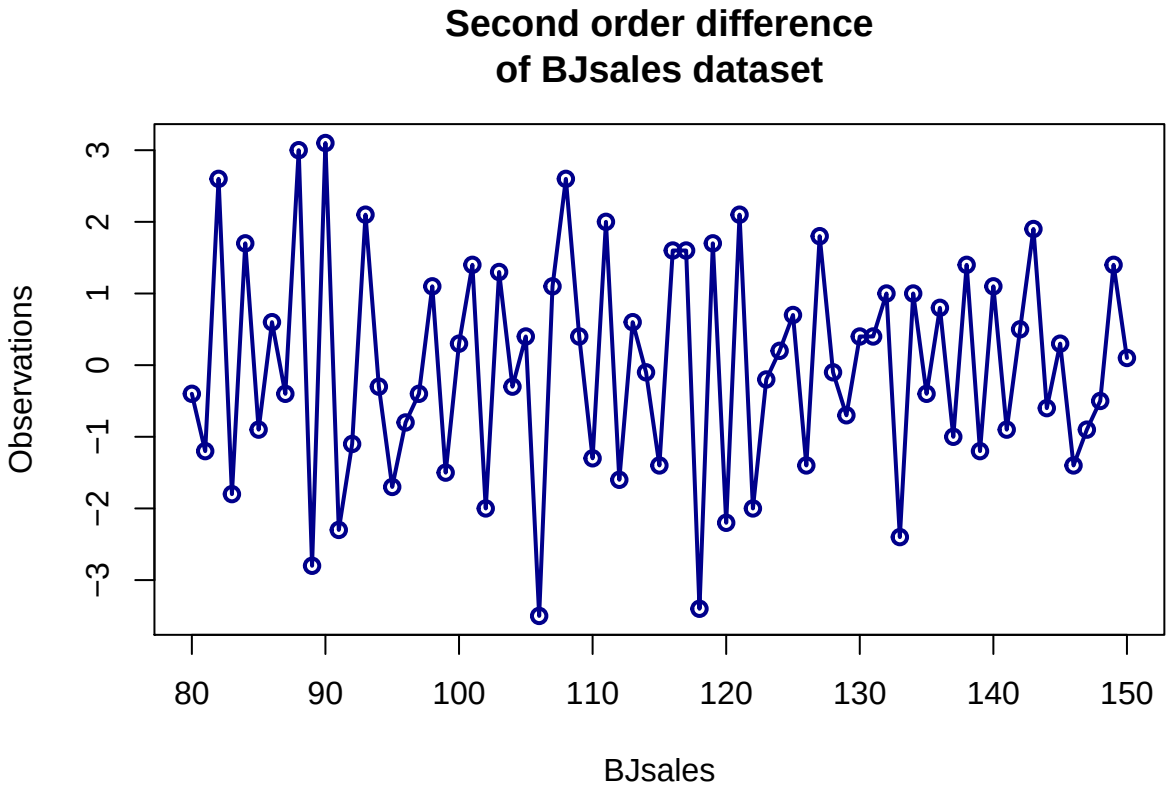

Figure 4: Second order difference of BJsales dataset

From the above figure we can clearly see that the dataset is stationary and does not follow any trends. We can fit the ARIMA model after observing ACF and PACF plots.

```
par(mfrow = c(1, 2))
acf(diff(diff(BJsales_data)), col="purple", lwd = 2, main = "ACF plot")
pacf(diff(diff(BJsales_data)), col = "purple", lwd = 2, main = "PACF plot")
```

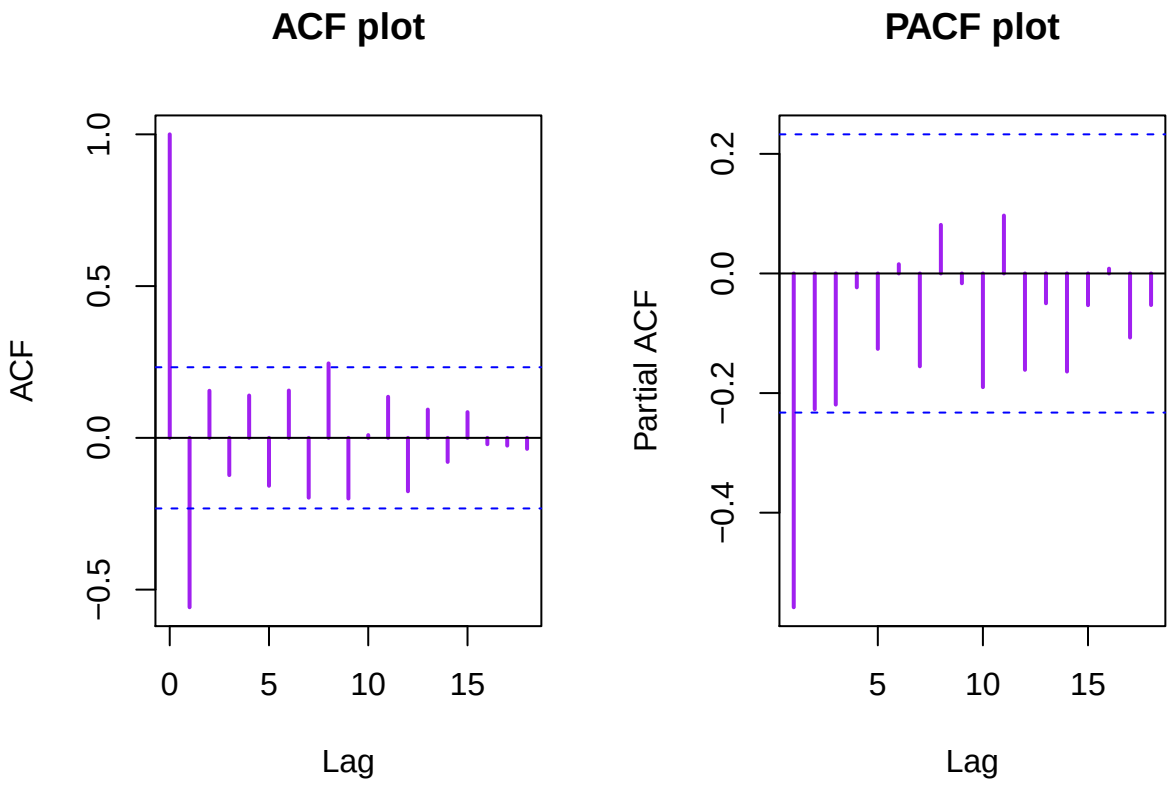

Figure 5: ACF/PACF of 2nd order difference dataset

From ACF lag 1 and 9 seems significant, while from PACF only lag 1 seems significant. Let's try building different ARIMA model and finalise the model based on value of AIC.

```
model <- arima(BJsales_data, order = c(0, 2, 1))
model
```

```
##
## Call:
## arima(x = BJsales_data, order = c(0, 2, 1))
##
## Coefficients:
##           ma1
##       -0.7080
## s.e.   0.0948
##
## sigma^2 estimated as 1.436:  log likelihood = -113.94,  aic = 231.88
```

Here, we found that ARIMA model with the order of (0, 2, 1) has smallest AIC value of 231.88.

Following is the equation of the model

equation one is obtained from the coefficient of the model

$\nabla^2 x_t = w_t - 0.708bw_{t-1}$ ..... eq(1)

deriving above equation 1, we get

$x_t = 2x_{t-1} - x_{t-2} + w_t 0.7080 w_{t-1}$

```
par(mfrow = c(2, 2))
plot(model$residuals, ylab="Residual", main="Residual plot", col="brown")
acf(model$residuals, main="Residual ACF plot", col="brown")
pacf(model$residuals, main="Residual PACF plot", col="brown")
qqnorm(model$residuals, main="Residual Q-Q plot", col="brown")
```

residuals-1.bb

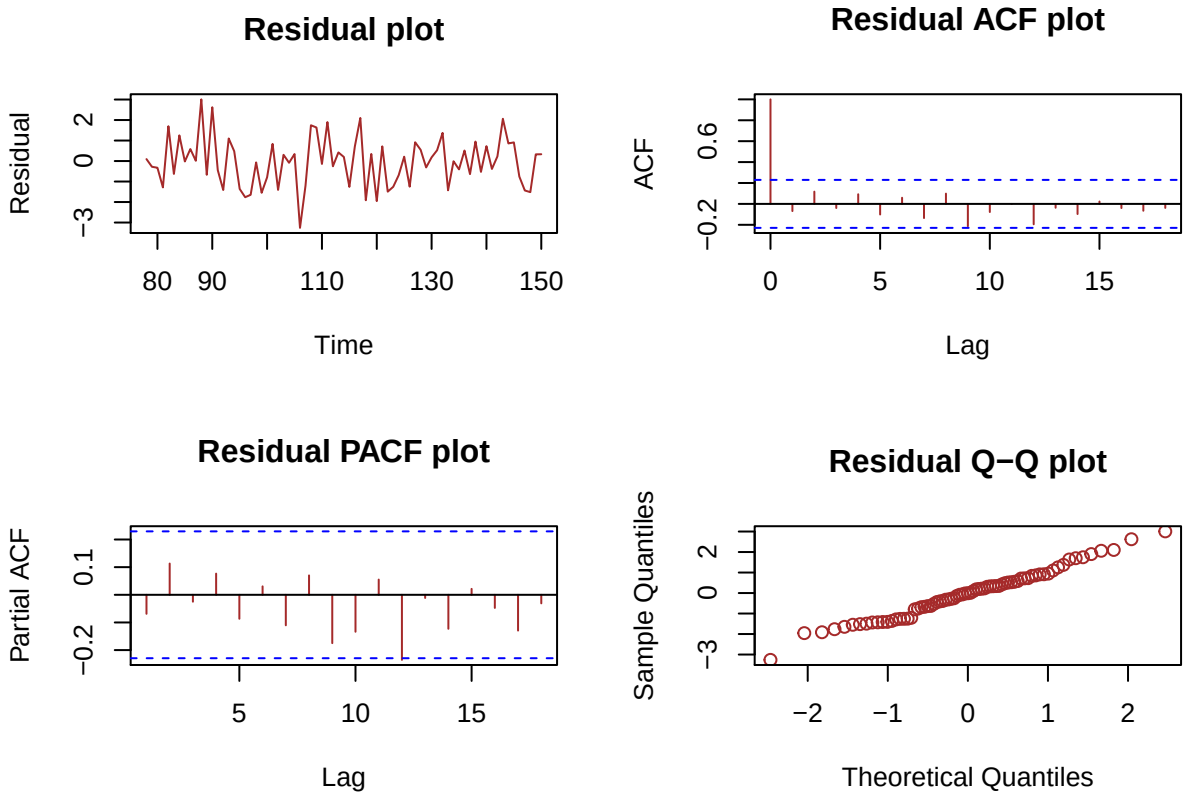

Figure 6: Residual plots

From above residual plots we can see no significant correlation as well as QQ-plot also looks normal. It does not show any trends or seasionality and neither consist of pure white noise. Thus, we can conclude that the model meets the assumption that the residuals are independent.

Now let's plot the forecast for 5 observation.

```
BJsales.Future <- forecast(model, head(5))
BJsales.Future.df <- BJsales.Future
kable(BJsales.Future.df, caption = "Predicted sales value for 5
      observation with confidence intervals")
```

Table 1: Predicted sales value for 5 observation with confidence intervals

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
151	262.9657	261.4300	264.5015	260.6170	265.3144
152	263.2314	260.7224	265.7405	259.3942	267.0687
153	263.4972	260.0026	266.9918	258.1526	268.8417
154	263.7629	259.2339	268.2919	256.8364	270.6894
155	264.0286	258.4076	269.6496	255.4321	272.6252

```
plot(BJsales.Future)
```

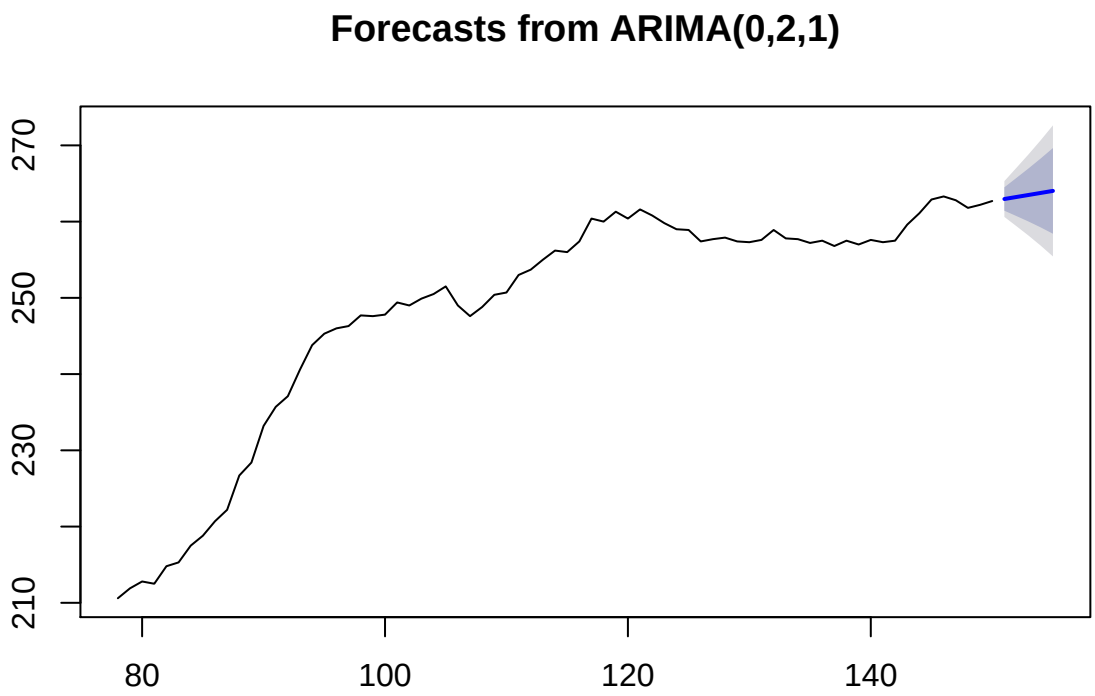

Figure 7: Forcasting five observation points

2. Mauna Loa Atmospheric CO2 Concentration Dataset

CO2 dataset is a time series of 468 monthlyu observations from 1959 to 1997. Atmospheric concentrations of CO2 are expressed in parts per million (ppm) and reported in the preliminary 1997 SIO manometric mole fraction scale.

We will try to fit a seasional ARIMA model to predict the level of co2 in 1998.

```
plot(co2, col = "dark green", xlab = "Year", ylab = "parts per million (ppm)",
     main = "CO2 Concentration", lwd=1)
```

Concentration data-1.bb

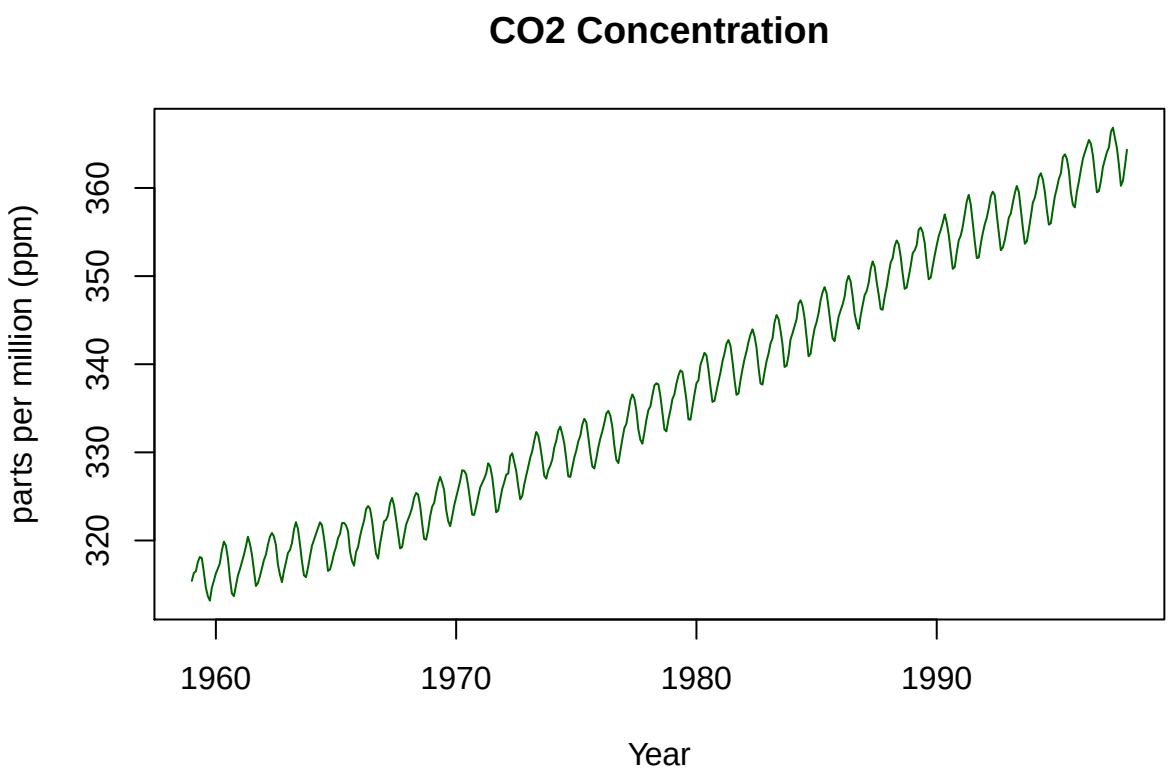

Figure 8: BJsales dataset

From the plot we can observe that the trend of co2 concentration is linear has seasional pattern over every month. Thus, the dataset is not stationary. To fit the model we need to acheive the stationarity of the datset. For that lets take a first order difference and observe if we can acheive the stationarity of the data.

```
par(mfrow = c(1, 1))
plot(diff(co2, lag = 12), col = "dark blue", xlab = "Year", ylab = "parts per million (ppm)",
     main = "First order difference of
     CO2 Concentration", lwd=1)
```

order difference co2-1.bb

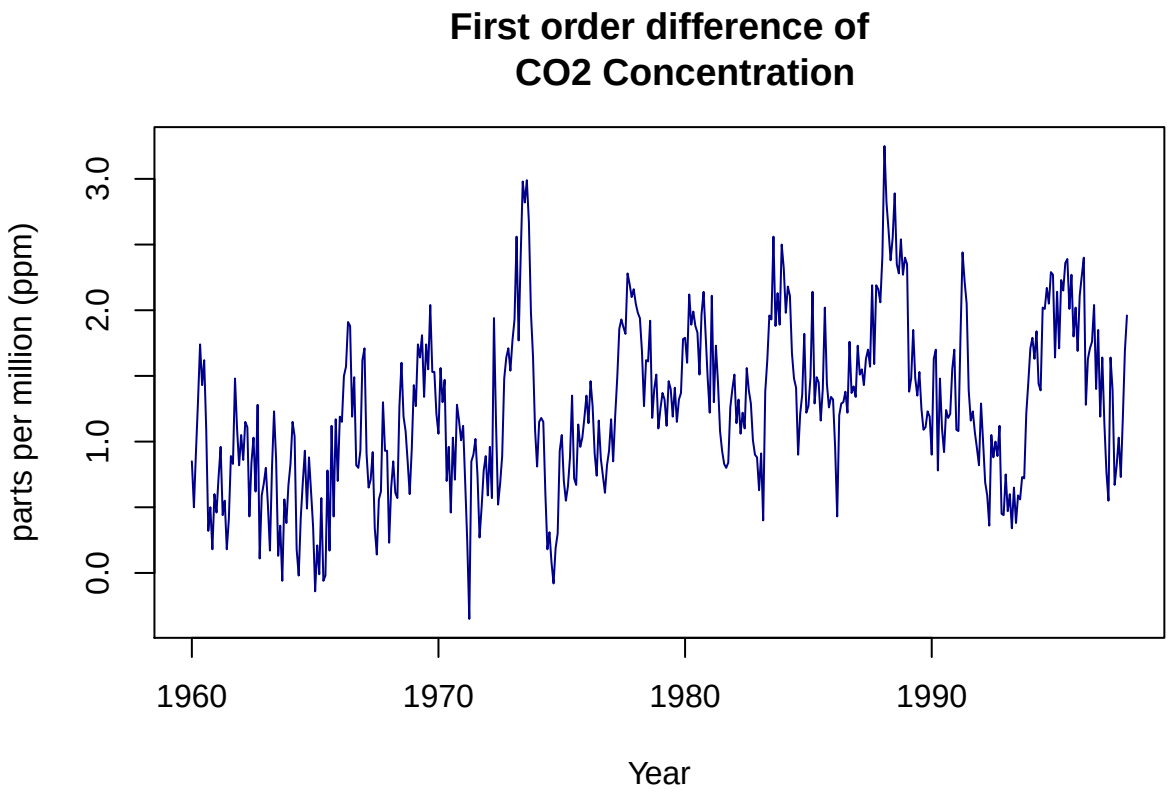

Figure 9: First order difference of CO2 Concentration dataset

We can observe that the time series plot shown above is still not staionary. Further taking second order difference of the dataset to make it stationary.

```
plot(diff(diff(co2, lag=12)), col = "dark blue", xlab = "Year", ylab = "parts per million (ppm)",
     main = "Second order difference of
     CO2 Concentration", lwd=1)
```

order difference co2-1.bb

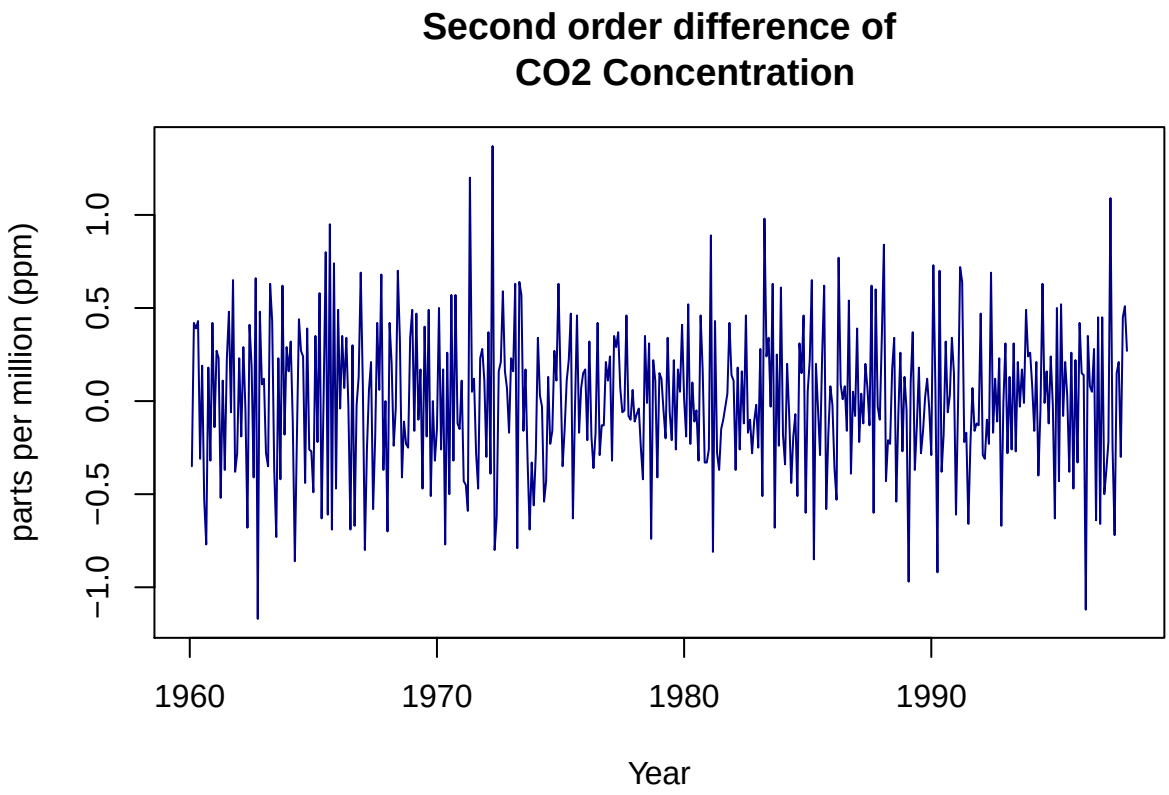

Figure 10: Second order difference of CO2 Concentration dataset

From the above figure we can clearly see that the dataset is stationary and does not follow any trends. We can fit the ARIMA model after observing ACF and PACF plots.

```
par(mfrow = c(1, 2))
acf(diff(diff(co2, lag = 12)), col="purple", lwd = 2, main = "ACF plot of co2")
pacf(diff(diff(co2, lag= 12)), col = "purple", lwd = 2, main = "PACF plot of co2")
```

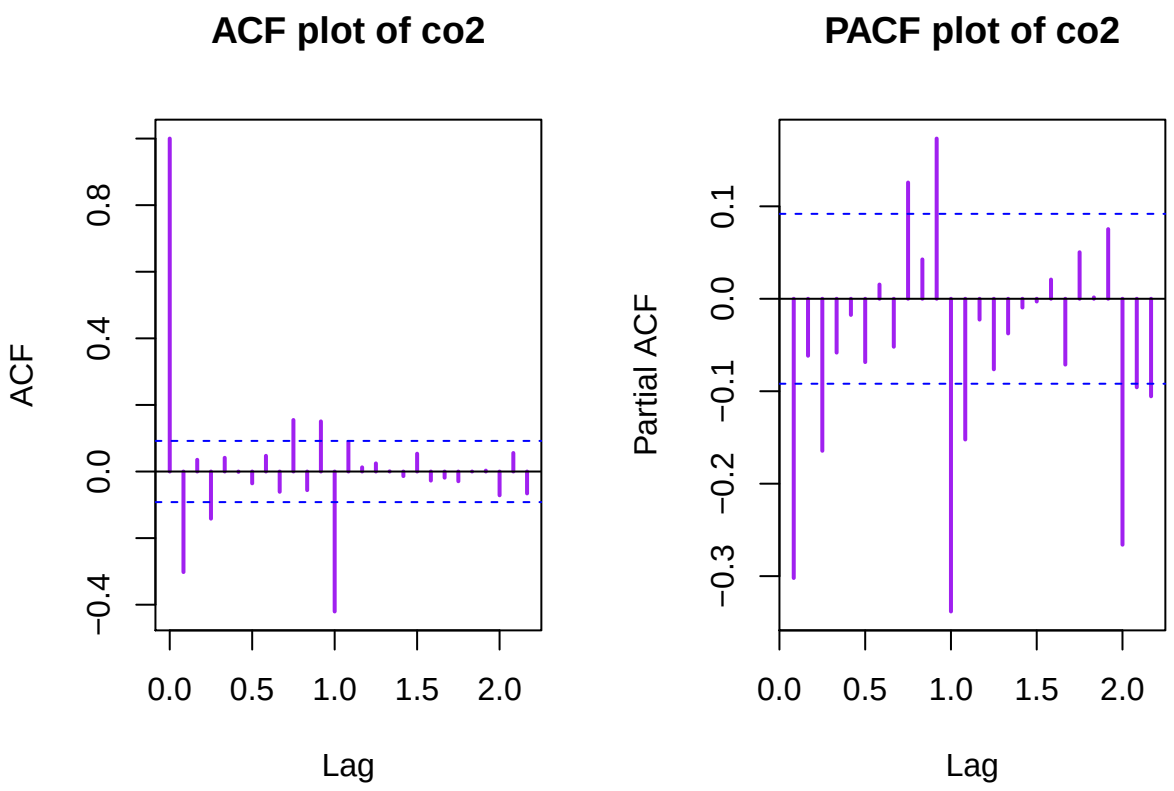

Figure 11: ACF/PACF of 2nd order difference co2 dataset

From ACF we can see that the lags are significant at '1', '3' '8', '10' and '11'. After lag '11', all the lags are insignificant and indicating no correlation. While from PACF lags '3', '9', '11', '12', '13', '24', '25' and '26' seems significant but we are uncertain to take the order of ARIMA. We will try building different ARIMA model and finalise the model based on value of AIC.

```
co2_model <- arima(co2, order = c(0,1,1), seasonal = c(0,1,1))
co2_model
```

```
##
## Call:
## arima(x = co2, order = c(0, 1, 1), seasonal = c(0, 1, 1))
##
## Coefficients:
##           ma1     sma1
##       -0.3501  -0.8506
## s.e.   0.0496   0.0257
##
## sigma^2 estimated as 0.0826:  log likelihood = -86.08,  aic = 178.16
```

Here, we found that ARIMA model with the order of (0, 1, 1)(0,1,1)[12] that has smallest AIC value of 178.16. From the model we have obtained Seasonal Moving Average coefficient which is -0.8506 and used to write the equation of the co2 model

$(1 - B_{12}) * (1 - B) * x_t = (1 - \phi_Q B_{12}) * (1 - \phi_q B) * w_t$

Where, moving average term (ma) is (ϕ_q) and seasional moving average term (sma) is (ϕ_Q)

$x_t = x_{t-1} + x_{t-12} - x_{t-13} + w_t - \phi_q wt - 1 - \phi_Q w_{t-12} + \phi_Q \phi_q w_{t-13}$

Replacing values of (ϕ_Q) and (ϕ_q), we get the final equation of prediction

$x_t = x_{t-1} + x_{t-12} - x_{t-13} + w_t 0.3501 w_{t-1} + 0.8506 w_{t-12} + 0.2977 w_{t-13}$

```
par(mfrow = c(2, 2))
plot(co2_model$residuals, ylab="Residual", main="Residual plot", col="brown")
acf(co2_model$residuals, main="Residual ACF plot", col="brown")
pacf(co2_model$residuals, main="Residual PACF plot", col="brown")
qqnorm(co2_model$residuals, main="Residual Q-Q plot", col="brown")
```

residuals-1.bb

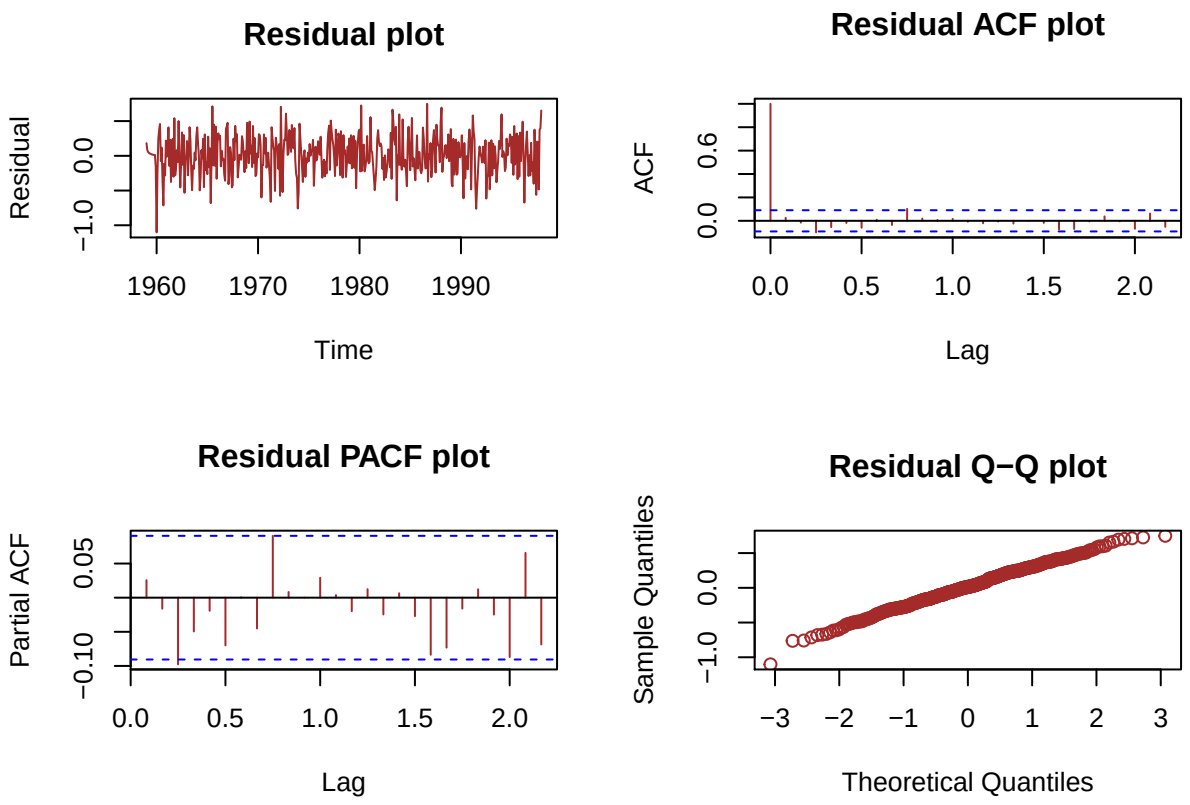

Figure 12: Residual plots

From above residual plots we can see no significant correlation as well as QQ-plot also looks normal. It does not show any trends or seasionality and neither consist of pure white noise. Thus, we can conclude that the co2_model meets the assumption that the residuals are independent.

Now let's plot the forecast of level of co2 in 1998.

```
co2.Future <- predict(co2_model,n.ahead=12,se.fit = FALSE)
co2.Future
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 1998 365.2034 366.0500 366.9133 368.2634 368.8324 368.1449 366.6424
##           Aug      Sep      Oct      Nov      Dec
## 1998 364.5871 362.7280 362.8559 364.2887 365.7025
```

plot co2-1.bb

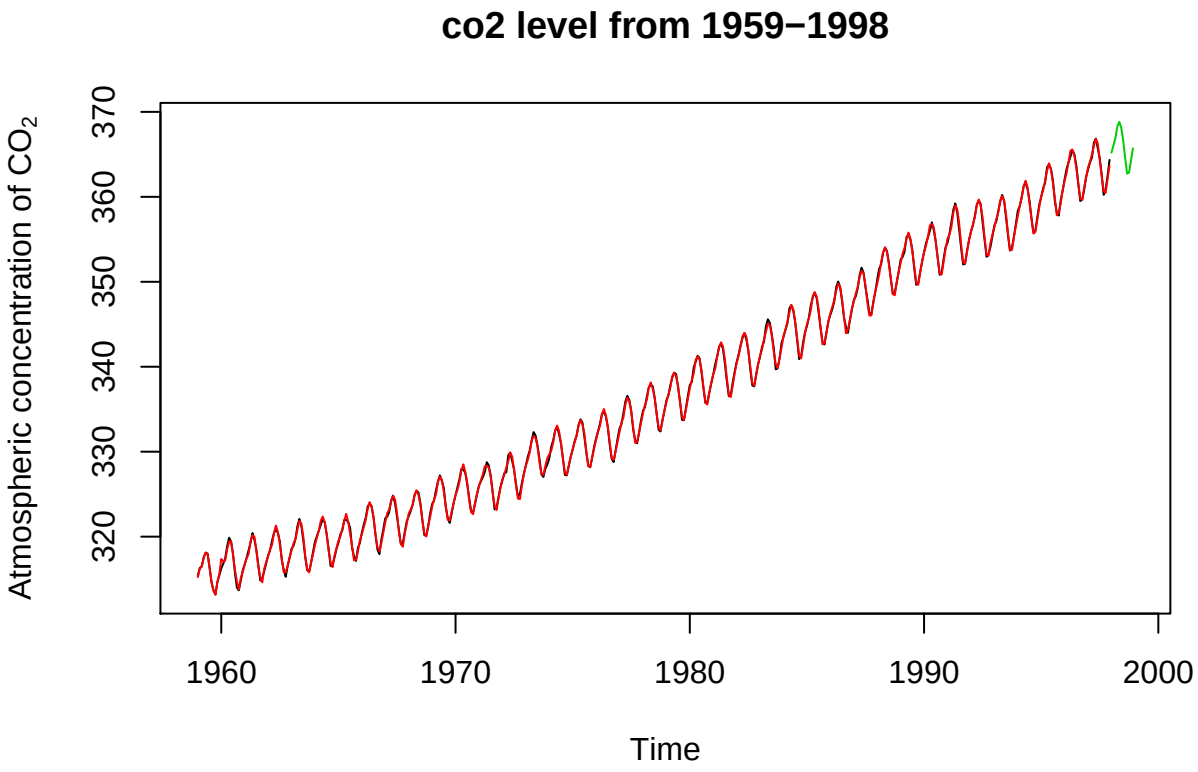

Figure 13: Forcasting level of co2 in 1998

3. Cloudy days dataset

Cloudy days dataset is simulated weather for 71 days, starting with sunny on the zeroth day and ending with rain on the 70^{th} day, where 1 denotes sunny, 2 denotes cloudy and 3 denotes rain. We are assuming that this simulated weather pattern follows a Markov Chain model.

The following is the estimated entries $\hat{p}_{ij}$ for the one step transition matrix P

```
Cloudy.data <- as.vector(readLines("C:\\Users\\Bipin Karki\\Desktop\\cloudydays.txt"))[2:72]
Fit.markovchain <- markovchainFit(data=Cloudy.data)
Fit.markovchain$estimate
```

```
## MLE Fit
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  1, 2, 3
##  The transition matrix  (by rows)  is defined as follows:
##           1         2          3
```

```
## 1 0.4516129 0.3870968 0.16129032
## 2 0.4193548 0.4838710 0.09677419
## 3 0.3750000 0.5000000 0.12500000
```

We know that on day 70 the weather is raining so we can create the row matrix as (0,0,1). Here, 1 indicates rainy day

day70 = (0,0,1)

For calculating the probability of weather of day 71, we are multiplying one step transition matrix by day70. Similary, for calculating probabilty of weather of day 72 we are multiplying two step transition matrix by day70 and so on. Probabilities of weather of day 72 and day 73 is shown below

```
Estimate.Fit.markovchain <- as.matrix(Fit.markovchain$estimate[1:3])
day_70 <- cbind(0,0,1)

day_71 <- day_70 %*% Estimate.Fit.markovchain
day_72 <- day_71 %*% Estimate.Fit.markovchain %*% Estimate.Fit.markovchain
day_73 <- day_72 %*% Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*%
  Estimate.Fit.markovchain

#merging all three matrix
day71_73 <- data.frame(rbind(day_71,day_72,day_73))
colnames(day71_73)= c("sunny","cloudy","rainy")
rownames(day71_73)= c("Day 71","Day 72", "Day 73")
kable(day71_73, caption = "Probability of weather on day 71, 72 and 73")
```

Table 2: Probability of weather on day 71, 72 and 73

	sunny	cloudy	rainy
Day 71	0.3750000	0.5000000	0.1250000
Day 72	0.4275718	0.4446621	0.1277661
Day 73	0.4274683	0.4445670	0.1279647

After the weather was rainy on day 70 from the above table we can say that the probability of weather to be cloudy on day 72 and day 73 is 0.4446621 and 0.4445670 respectively. Similarly, probability of weather to stay rainy for both day 72 and day 73 is 0.1277661 and 0.1279647 respectively. Whereas, probability of weather to be sunny for both day 72 and day 73 is 0.4275718 and 0.4274683 respectively.

Table 3: Transition matrix by row

sunny	cloudy	rainy
0.4516129	0.3870968	0.1612903
0.4193548	0.4838710	0.0967742
0.3750000	0.5000000	0.1250000

Lets calculate the probability of the weather staying rainy on day 71, then cloudy on day 72 and sunny on day 73. This probaility is calculated from transition matrix and can be written as

Rainy $\rightarrow$ Rainy $\rightarrow$ Cloudy $\rightarrow$ Sunny, or $3 \rightarrow 3 \rightarrow 2 \rightarrow 1$. So this is

$p_{33} * p_{32} * p_{21} = 0.1250 * 0.5 * 0.4193548 = 0.0262$

Thus, probability of the weather staying rainy on day 71, then cloudy on day 72 and sunny on day 73 is 0.0262

Steady State condition

As we know that sometimes Markov Chain may settle into a steady-state condition where the increment of steps of transition matrix doesnot change the probability of occurance of events. This means that we multiply P with itself until the result does not change any more. We obtained the steady state condition for this data set at p^6 which is shown in table below

```
day_74 = day_70 %*% Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*%
  Estimate.Fit.markovchain %*% Estimate.Fit.markovchain
day_75 = day_70 %*% Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*%
  Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*% Estimate.Fit.markovchain
day_76 = day_70 %*% Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*%
  Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*%
  Estimate.Fit.markovchain %*% Estimate.Fit.markovchain
day_77 = day_70 %*% Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*%
  Estimate.Fit.markovchain %*% Estimate.Fit.markovchain %*%
  Estimate.Fit.markovchain %*% Estimate.Fit.markovchain  %*%   Estimate.Fit.markovchain

day74_77 <- data.frame(rbind(day_74,day_75,day_76,day_77))
colnames(day74_77)= c("sunny","cloudy","rainy")
rownames(day74_77)= c("Day 74","Day 75", "Day 76", "Day 77")
kable(day74_77, caption = "Checking steady state probability")
```

Table 4: Checking steady state probability

	sunny	cloudy	rainy
Day 74	0.4274804	0.4445538	0.1279658
Day 75	0.4274686	0.4445659	0.1279655
Day 76	0.4274683	0.4445670	0.1279647
Day 77	0.4274683	0.4445670	0.1279647

In the table we can see that the probability of weather is exactly the same for day 76 and 77. Thus, we can say that the n step transition matrix is settled at p^6. This means that for a typical day the state of weather to be Sunny is 0.4274683, cloudy is 0.4445670 and rainy is 0.1279647.