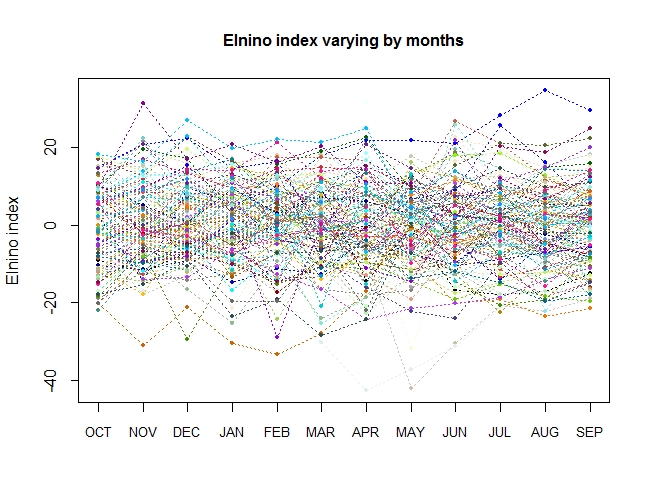
**Dataset introduction-**

The aim of this project is to predict the amount of snowfall for December 2018. The snowfall

depends on various factors like sunspots, elnino winds, average temperature and ice on lake

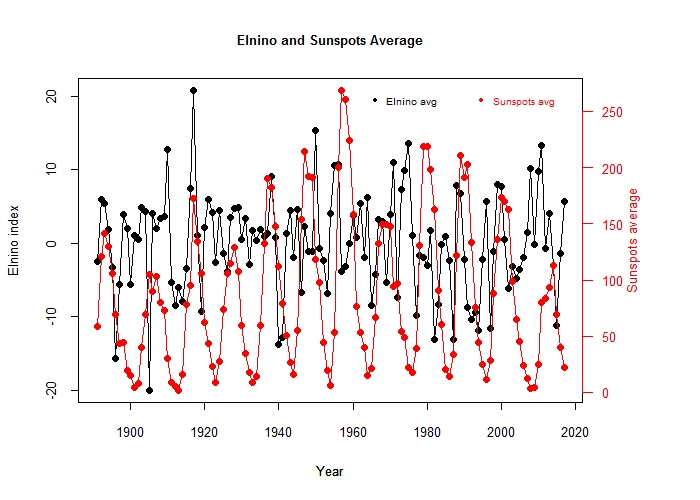
superior. These we collected over a period of 128 years i.e., 1890-91 to 2016-17. We have used ARIMA model to forecast the snowfall data for December 2018.

We have considered Elnino index observed during various months and tried to observe its impact on Snowfall data.

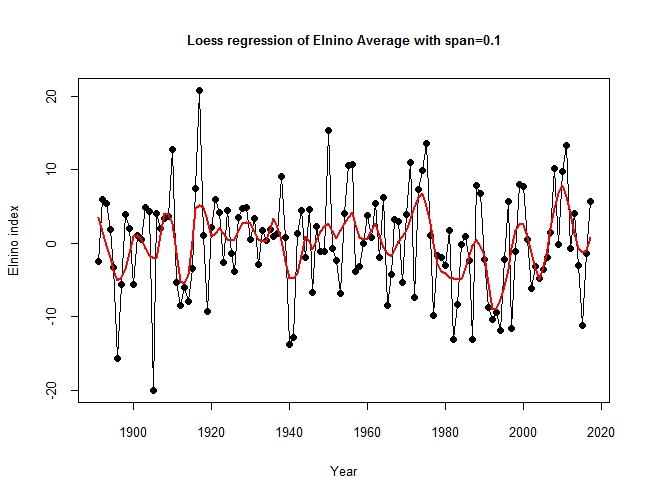


From this plot we can see a change in elnino index over the months. We can observe the trends

for every month and the outliers.



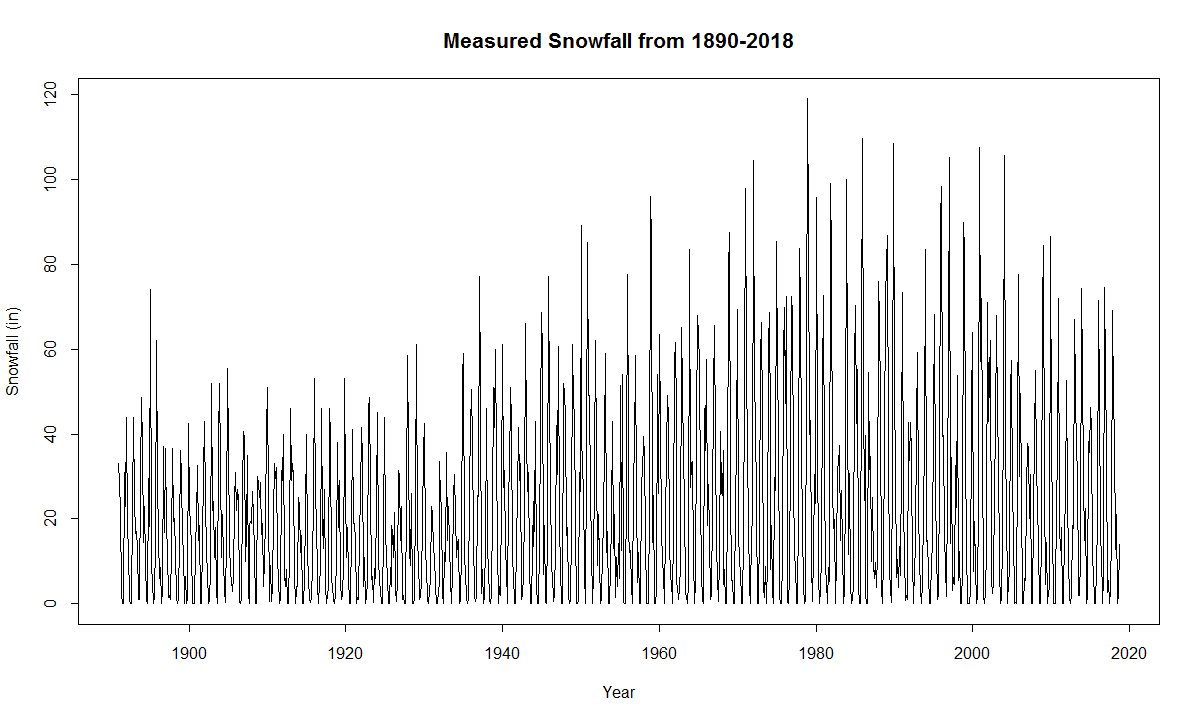
The above figure represents a average of elnino and sunspots average taken over the years. In this graph we have two y-axes, each representing elnino index and sunspots average respectively. We can see that there is possoble linear relationship between sunspots and elnino. We can observe a correlation of around 0.0331 between elnino and sunspots average.



This plot indicates a loess model with a span of 0.1, we tried to create a prediction model using loess but it didn’t predict exact values and it has low r-squared values. Hence we used forecasting models for our prediction model.

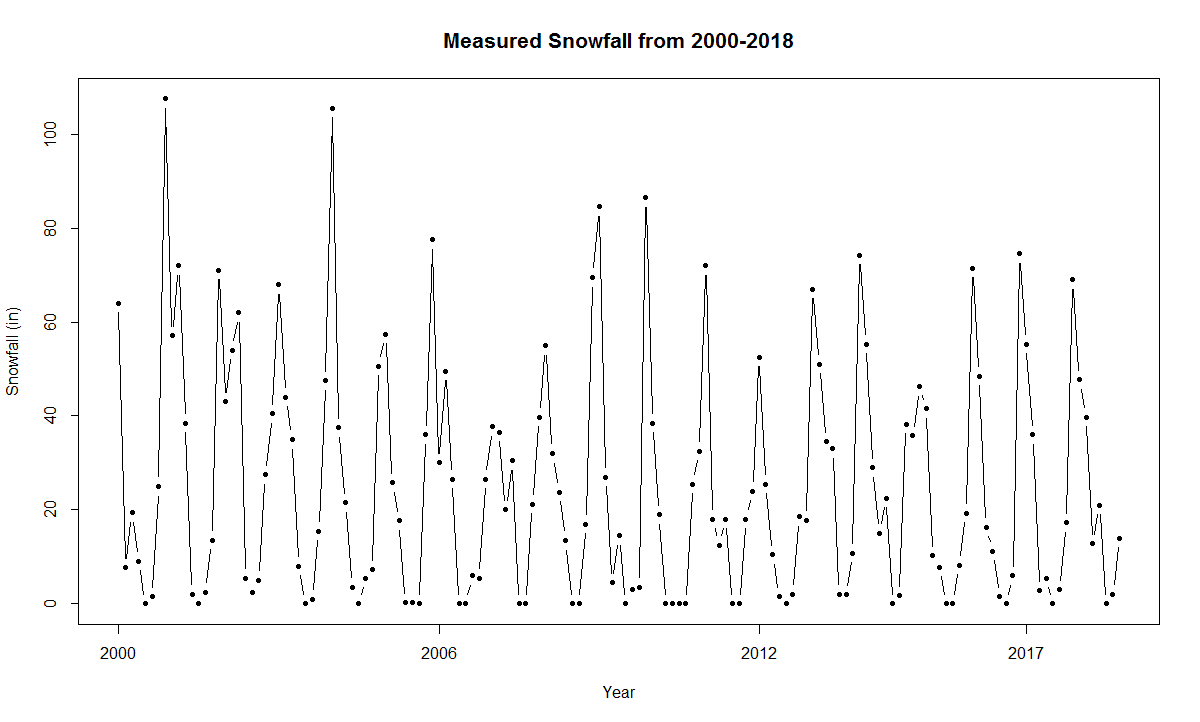
**Forecasting December month’s snowfall data:**

For our convenience we are using only Snowfall data observed in various months from 1891 to 2018 and trying to predict the amount of Snowfall during December 2018.



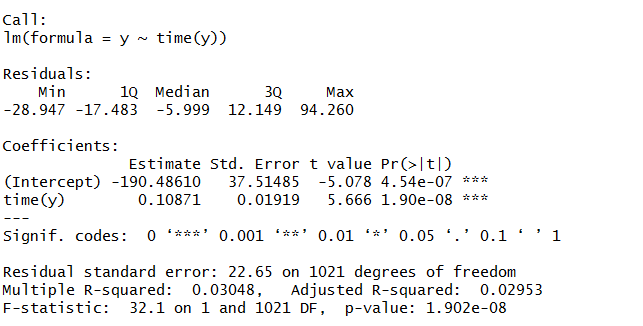
The above figure describes the amount of measured snowfall in inches from 1891 to 2018 over various months. This can be observed by converting the given data into a time series dataset.

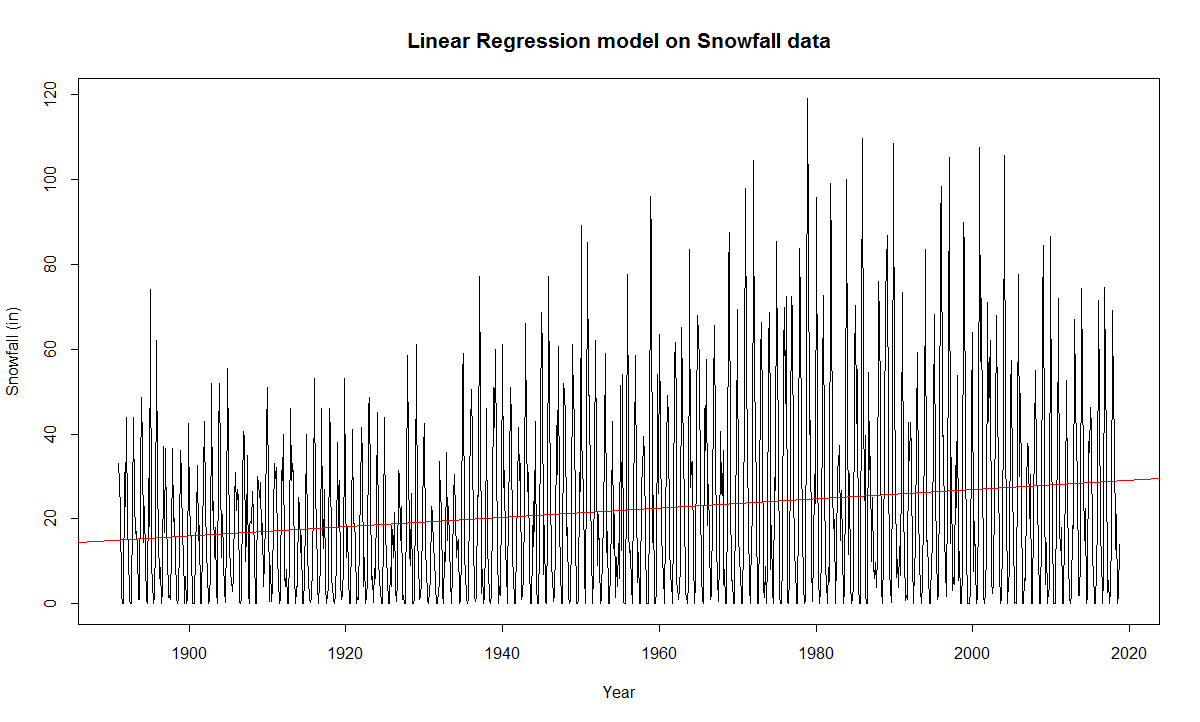
We can take a closer look of the data and can observe the variations in snowfall from 2000 to 2018.



Linear Regression:

On applying Linear Regression on the snowfall data with respect to time we can observe that the R squared value is 0.03048 which is very less and is not appropriate for the given data.



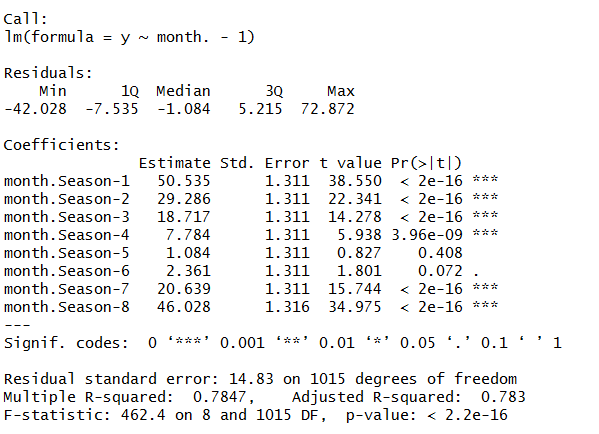


**Seasonal Means model:**

On applying deterministic seasonal means model on the given snowfall data we can observe that the R square is around 0.78 which is very high when compared with the linear regression model. Hence, in this model we can determine the β coefficients due to every month in the data. Here season-1, season-2, season-3, season-4, season-5, season-6, season-7 and season-8 represents January, February, March, April, May, October, November and December respectively.

Here the output of the model is fully determined by the parameter values and the initial conditions.

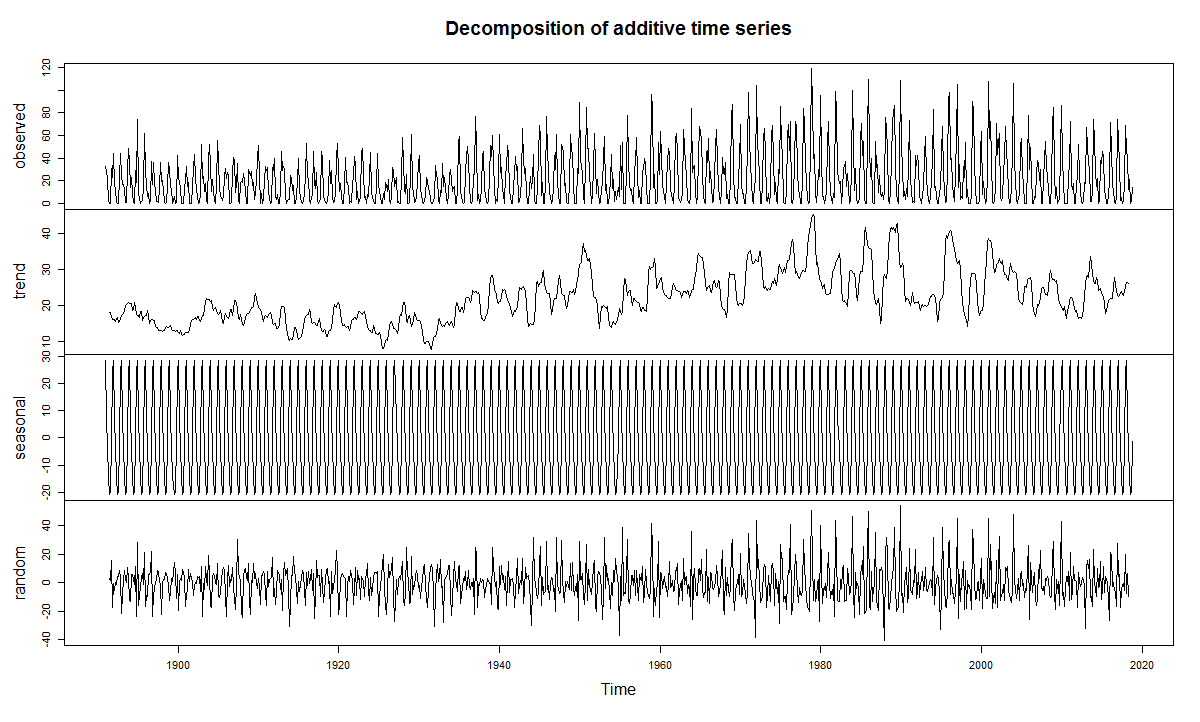
But in reality, the natural world is buffeted by stochasticity. Hence, we need to consider stochastic models such as ARIMA to appropriately predict the data.



**ARIMA model:**

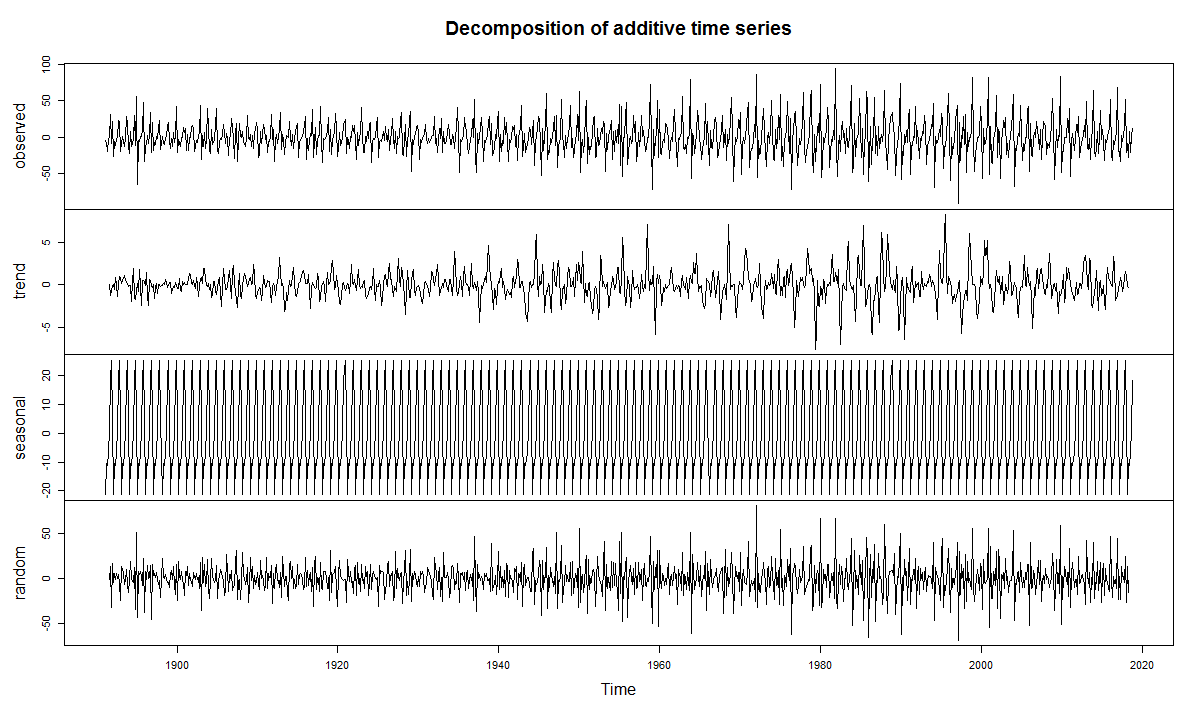
Stochastic models possess some inherent randomness. The same set of parameter values and initial conditions will lead to an ensemble of different outputs. ARIMA is one such stochastic model.

On decomposing the given snowfall data we can observe that there exists a linear trend and seasonality in the data.

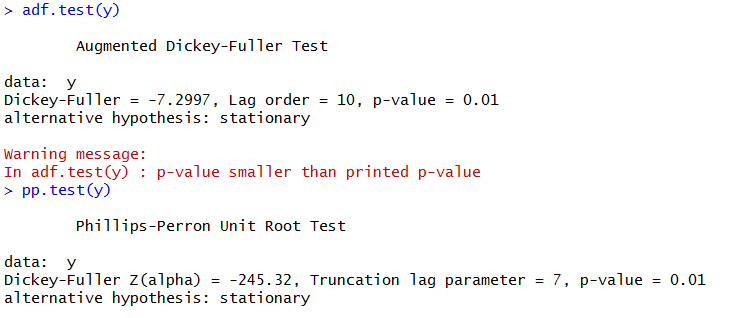


**After applying difference:**

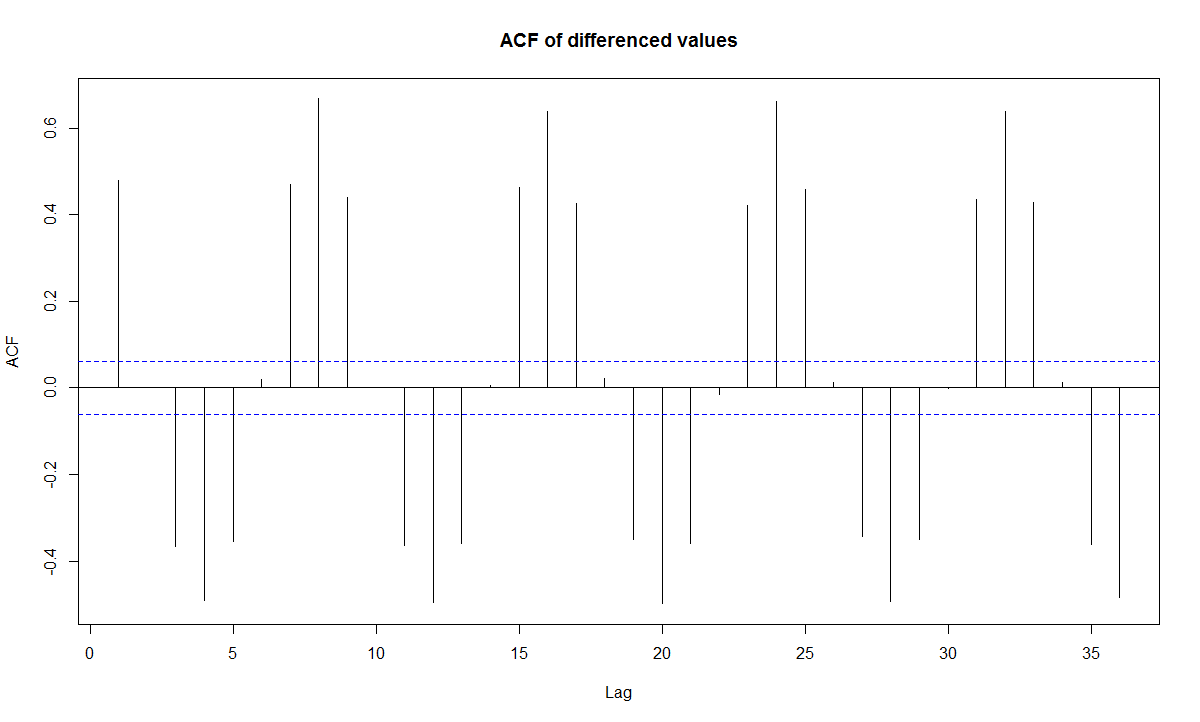
This increasing linear trend can be removed by applying difference to the data as observed in the below figure.



In order to apply ARIMA, the data needs to be stationary which can be observed using Augmented Dickey-Fuller test and Phillips-Perron Unit root test.

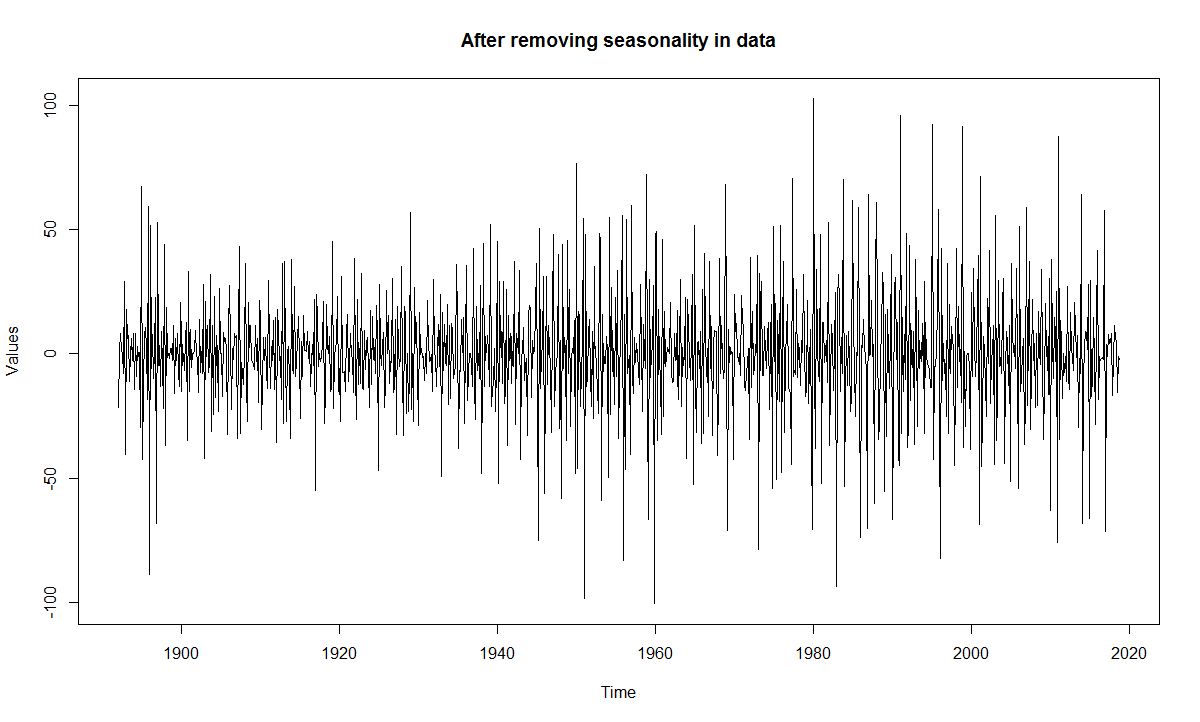


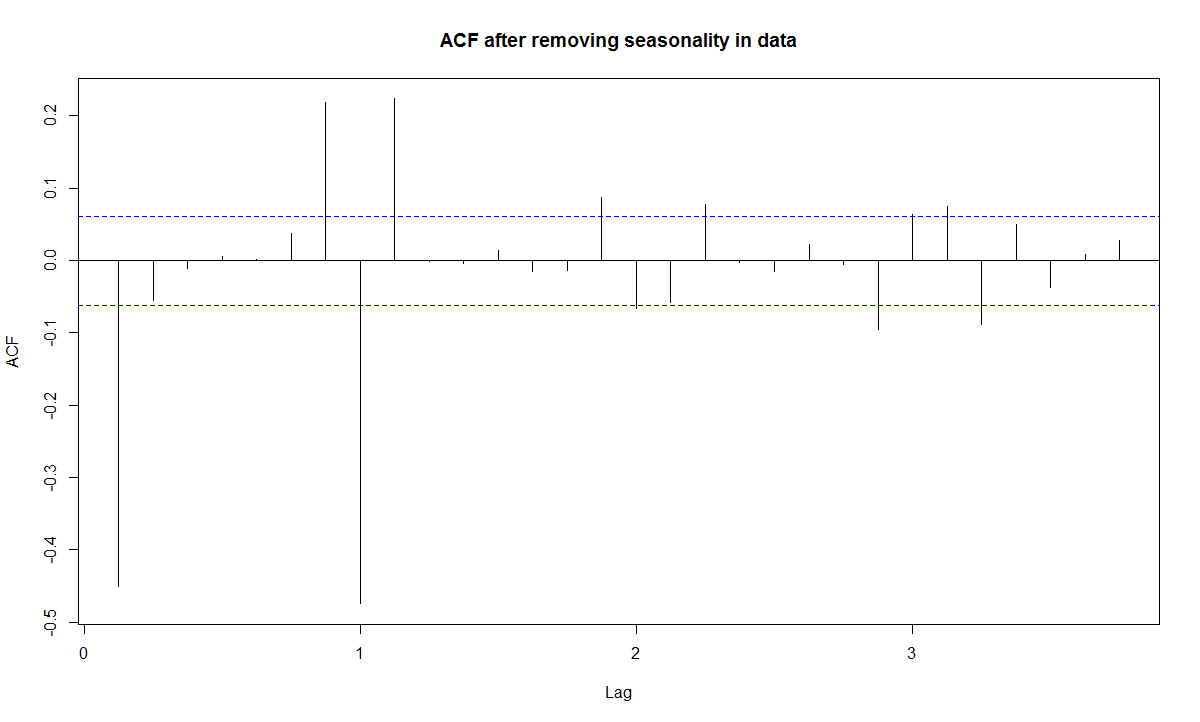
On applying, the above two test we can observe that the differenced snowfall data is stationary.



On applying autocorrelation function (ACF) on the differenced snowfall data we can observe that there is some seasonality present in the data and is repeating for every 8 lags. Hence, this seasonality needs to be removed from the data in order to get appropriate results.

The seasonality can be removed using SARIMA model.

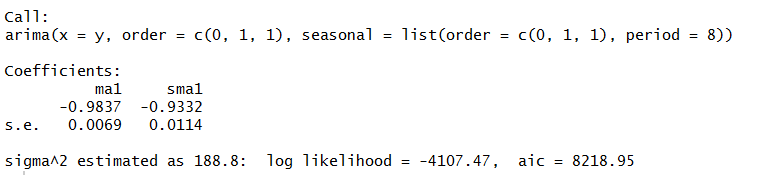




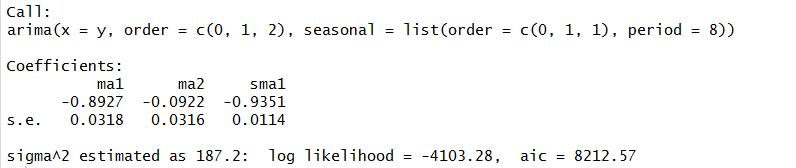
Now we can observe that there is no seasonality present in the data from the above graph.

**Comparing various ARIMA models:**

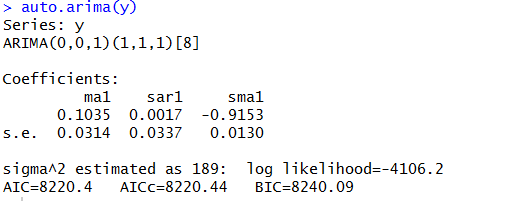
**ARIMA (0,1,1) (0,1,1)8**



**ARIMA (0,1,2) (0,1,1)8**



**Auto.arima( )**

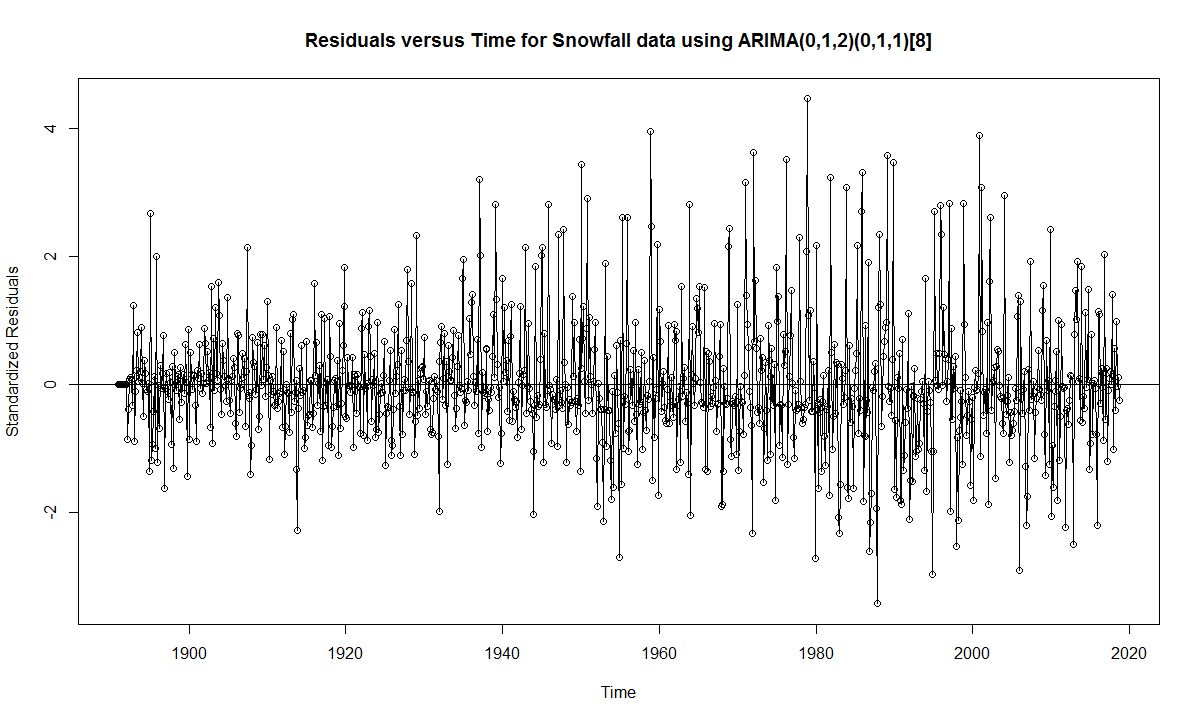


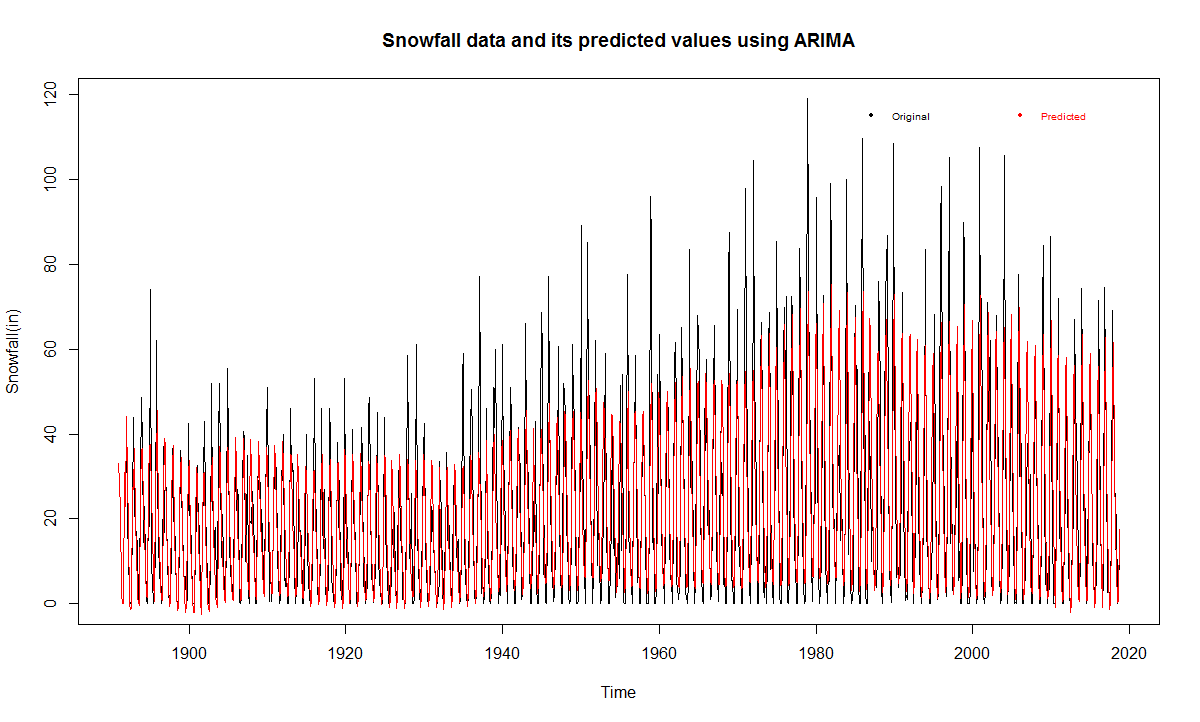
|  |  |
| --- | --- |
| Model | AIC |
| ARIMA (0,1,1) (0,1,1)8 | 8218.95 |
| ARIMA (0,1,2) (0,1,1)8 | 8212.57 |
| Auto.arima  ARIMA (0,0,1) (1,1,1)8 | 8220.4 |

From the above three models we can observe that ARIMA (0,1,2) (0,1,1)8 consists of least AIC value.

Hence we consider ARIMA (0,1,2) (0,1,1)8 to predict the snowfall data for December month.

The residuals versus Time for Snowfall data using ARIMA (0,1,2) (0,1,1)8 is as follows.

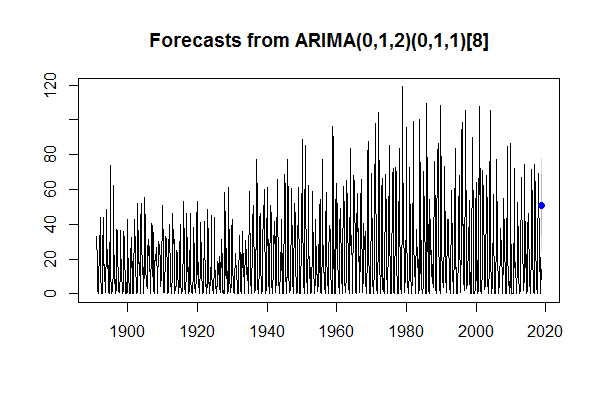




We can observe an R square value of 0.805537 using ARIMA (0,1,2) (0,1,1)8. Hence from the R square value we can determine that this model produces 80.5% accurate results.



**Forecasting the Snowfall of December 2018:**



The above graph forecasts the 2018 December month’s snowfall which is represented by a blue dot along with its upper and lower confidence levels represented by grey color.



Here with ARIMA (0,1,2) (0,1,1)8 the forecasted snowfall value is 50.78987 and a standard error of 13.68125.

We can observe that at 95% confidence interval of the snowfall measured in December 2018 will be in the range from [23.93537, 77.64438]

**Appendix (Code):**

library(gplots)

filter <- 1:128 ##ignor the last year, update this if you get more data:

alldat <- read.csv("weather.csv")[filter,]

elnino\_c <- read.csv("Book2.csv")[16:142,]

sun\_c <- read.csv("Book3.csv")

years <- alldat[,1]

weather <- alldat[,1:9]

elnino <- alldat[,12:23]

sunspots <- alldat[,25:36]

avgtemp <- alldat[,38:49]

ice <- alldat[,50:57]

matplot(t(elnino),type="o",pch=16,xaxt= "n",lty=3,cex=0.5,tcl=-.4,cex.axis=1,cex.main=1,cex.lab=1,

ylab="Elnino index",main="Elnino index varying by months",col=colors())

axis(1,1:12,c("OCT","NOV","DEC","JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP"),cex.axis=0.8)

for(i in 1:nrow(elnino\_c)){

elnino\_c$avg[i] <- sum(elnino\_c[i,2:13])/12

}

for(i in 1:nrow(sun\_c)){

sun\_c$avg[i] <- sum(sun\_c[i,2:13])/12

}

par(mar=c(5, 4, 4, 6) + 0.1)

plot(elnino\_c$Year,elnino\_c$avg,type="o",xlab = "Year",ylab="Elnino index"

,main="Elnino and Sunspots Average",pch=16,tcl=-.4,cex.axis=0.8,cex.main=0.8,cex.lab=0.8)

par(new=TRUE)

plot(sun\_c$Year,sun\_c$avg,type="o",xlab = "Year",ylab="",axes=F

,pch=16,tcl=-.4,cex.axis=0.8,cex.main=0.8,cex.lab=0.8,col="red")

mtext("Sunspots average",side=4,col="red",line=2,cex=0.8)

axis(4, ylim=c(0,300), col="red",col.axis="red",las=1,cex.axis=0.8)

cor(elnino\_c$avg,sun\_c$avg)

legend("topright",legend=c("Elnino avg","Sunspots avg"),bty ="n",y.intersp=0.8,xpd=TRUE,cex=0.6,horiz=TRUE,

text.col=c("black","red"),pch=16,col=c("black","red"))

l1 <- loess(avg~Year,span=.1,data=elnino\_c)

summary(l1)

l1

cor(elnino\_c$avg,predict(l1,elnino\_c))^2

plot(elnino\_c$Year,elnino\_c$avg,type="o",xlab = "Year",ylab="Elnino index"

,main="Loess regression of Elnino Average with span=0.1",pch=16,tcl=-.4,cex.axis=0.8,cex.main=0.8,cex.lab=0.8)

lines(elnino\_c$Year,predict(l1),type="l",col="red",lwd=2)

library(gplots)

library(TSA)

library(tseries)

library(forecast)

snow <- read.csv("ts\_snow.csv",header=F)

t <- snow[-c(1:3),]

y <- ts(t,start=c(1891,1),end=c(2018,7),frequency = 8)

print(y, calendar = T)

plot(y,xlab="Year",ylab="Snowfall (in)",main="Measured Snowfall from 1890-2018",

cex=0.5,tcl=-.4,cex.axis=1,cex.main=1.3,cex.lab=1)

plot(y[873:1023],type="b",xaxt="n",main = "Measured Snowfall from 2000-2018"

,xlab="Year",ylab="Snowfall (in)",pch=16,cex=0.7,tcl=-.4,cex.axis=1,cex.main=1.3,cex.lab=1)

axis(side=1,at=c(1,49,97,137),labels=c(2000,2006,2012,2017),cex.axis=1)

model1=lm(y~time(y))

summary(model1)

plot(y,type="l",xlab="Year",ylab="Snowfall (in)",main="Linear Regression model on Snowfall data",

cex=0.5,tcl=-.4,cex.axis=1,cex.main=1.3,cex.lab=1)

abline(model1,col="red")

month.=season(y)

model2=lm(y~month.-1)

summary(model2)

c <- diff(y)

plot(c,ylab='Change in Log(Snowfall)',type='l')

adf.test(y)

pp.test(y)

acf(as.vector(y),lag.max=36,main="ACF of differenced values")

plot(decompose(y))

plot(decompose(c))

plot(diff(diff(y),lag=8),main="After removing seasonality in data",ylab="Values")

acf(diff(diff(y),lag=8),main="ACF after removing seasonality in data")

pacf(diff(diff(y),lag=8))

m1.s=arima(y,order=c(0,1,2),seasonal=list(order=c(0,1,1),

period=8))

m1.s

plot(window(rstandard(m1.s),start=c(1891,1)),end=c(2018,7),frequency = 8,

ylab='Standardized Residuals',type='o',main = "Residuals versus Time for Snowfall data using ARIMA(0,1,2)(0,1,1)[8]")

abline(h=0)

acf(as.vector(window(rstandard(m1.s))))

pacf(as.vector(window(rstandard(m1.s))))

hist(as.vector(window(rstandard(m1.s))))

a <- forecast(Arima(y,order=c(0,1,2),seasonal=list(order=c(0,1,1))),1)

plot(a)

a

plot(y,main="Snowfall data and its predicted values using ARIMA",ylab="Snowfall(in)")

lines(fitted(m1.s),col="red")

legend("topright",legend=c("Original","Predicted"),bty ="n",y.intersp=0.8,xpd=TRUE,cex=0.6,horiz=TRUE,

text.col=c("black","red"),pch=16,col=c("black","red"))

cor(y,fitted(m1.s))