TIME SERIES ANALYSIS OF NASDAQ STOCKS AND SEASONAL MARS WEATHER

Spring 2022 - MA 641 Time Series Analysis

Chrisia Sally Silva
Under the guidance of Prof Hadi Safari Katesari

TABLE OF CONTENTS

1 NON-SEASONAL NASDAQ COMPOSITE STOCKS

1.1 MOTIVATION AND INTRODUCTION OF THE PROBLEM

In this paper seasonal dataset is analyzed based on mean, and variance factors.

The NASDAQ Composite is a stock market index that covers all Nasdaq-listed stocks in Yahoo! Finance website. It is one of the three most widely followed stock market indices in the United States. The NASDAQ Composite is significantly weighted in favor of companies in the information technology industry. The NASDAQ Composite is a capitalization-weighted index, with its price determined by adding the closing price and index share of all the index's shares. The total is then divided by a divisor, lowering the result's order of magnitude.

This paper processes the NASDAQ Composite volatile stock data and fits an appropriate GARCH model to it. This GARCH model will accurately forecast the future variances based on the current values. The challenge here is to fit the extremely volatile data into an appropriate time series model for producing the best possible forecasts.

1.2 DATA

1.2.1 Dataset Description

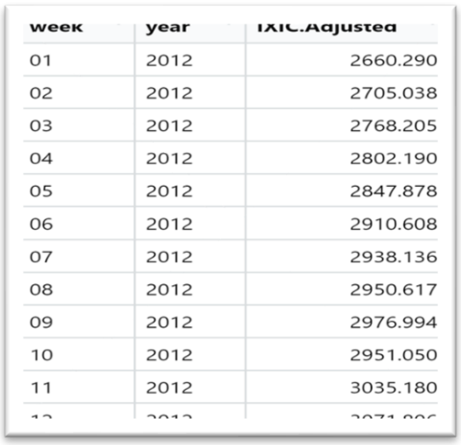
The dataset is collected from NASDAQ Composite stocks, with ticker symbol IXIC, from the Yahoo! Finance website - https://finance.yahoo.com/quote/%5EIXIC/history?p=%5EIXIC. Around 12,908 entries are present in the dataset collected from 1971-02-05 till 2022-04-08.

It consists of the following fields: Date, Open, High, Low, Close, Adj.Close, and Volume.

Date is of the yyyy-mm-dd format, Open is the opening price, High is the highest price of the day, Low is the Lowest price of the day, Close is the closing price which is the cash worth of a stock at the conclusion of the day, Adj.Close is the adjusted closing price which adjusts a stock's closing price to reflect its worth after any corporate action is taken into account, and Volume is the volume of stocks.

1.2.2 Data Preprocessing

The datasets contain NA/null values which are replaced with the previous values to maintain continuity and provide the best results. The week and year are extracted from the NASDAQ Adjusted Close Stock Data. Aggregate of Adjusted Close stock values based on week and year are taken to get 527 records.



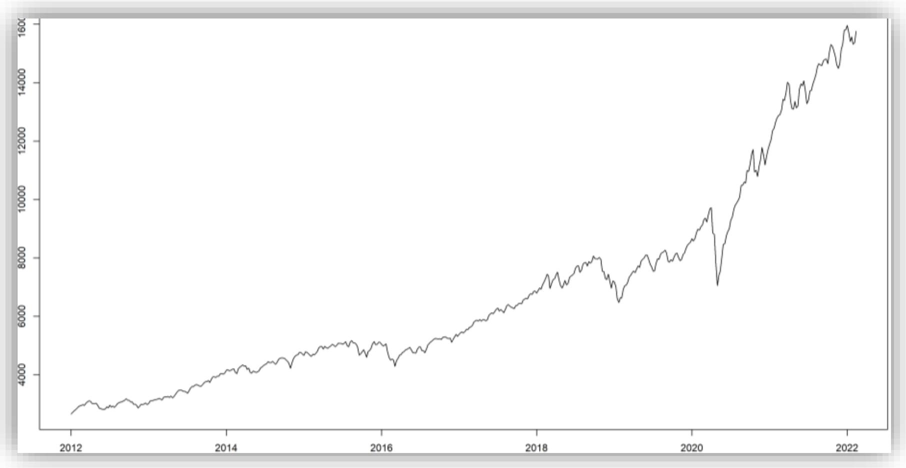
week	year	IXIC.Adjusted
01	2012	2660.290
02	2012	2705.038
03	2012	2768.205
04	2012	2802.190
05	2012	2847.878
06	2012	2910.608
07	2012	2938.136
08	2012	2950.617
09	2012	2976.994
10	2012	2951.050
11	2012	3035.180

1.3 Box-Jenkins model

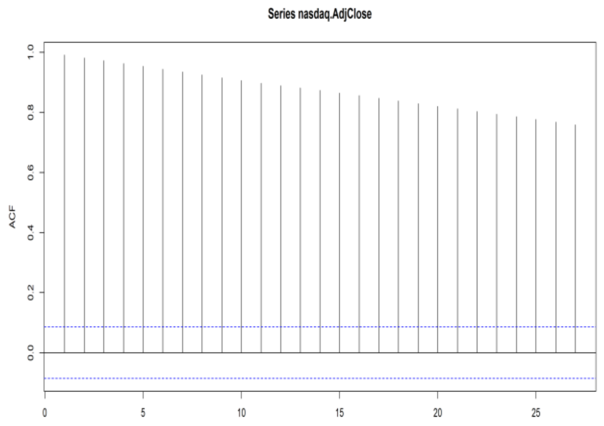
We follow the Box Jenkins model to explore the various methods to identify models used to forecast the data.

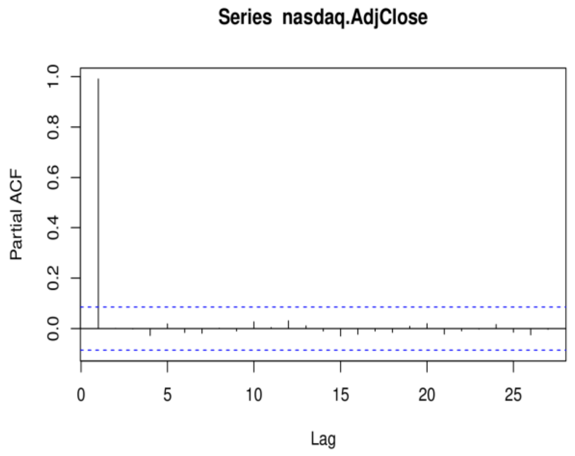
Plot of NASDAQ Composite Weekly Aggregate Adjusted Close Stock data. We observe an increasing trend and high variance.

NASDAQ Composite Stock Price Time series



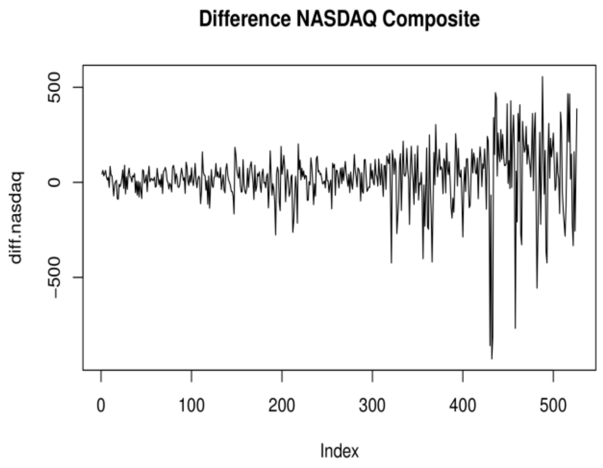
ACF decreases but does not die thus it needs differencing and PACF cuts off after lag 1, both graphs are as follows:

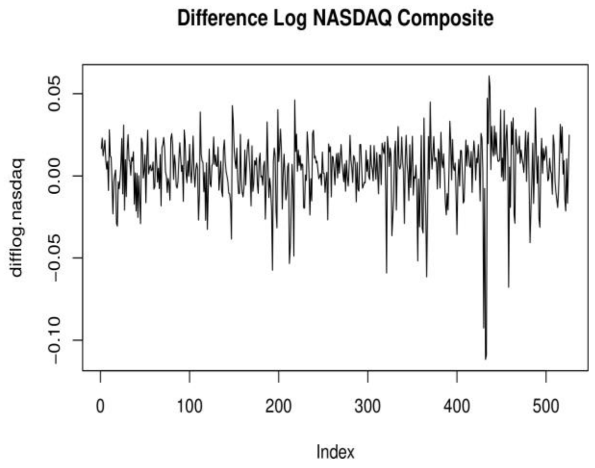


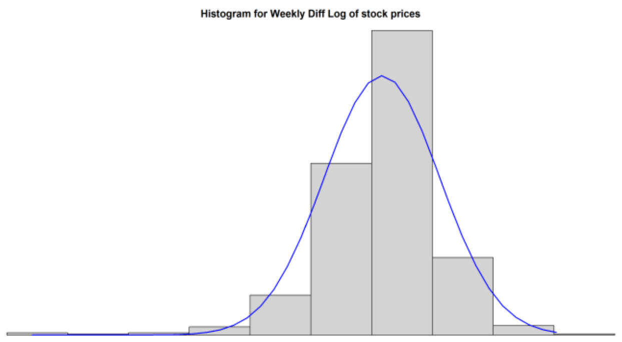


Stationary test performed is the ADF and CADF test which yields a p-value of around 0.9 which is greater than 0.05 which proves that the data is not stationary. KPSS test gives p-value of 0.01 is less than 0.05, we reject null, thus the data is not stationary.

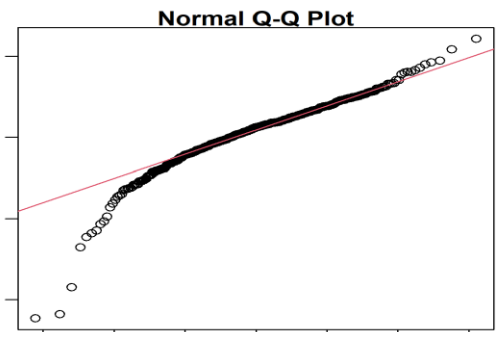
Differenced data is seen as below, and we take log of the date and take its first difference. We get the following plots:





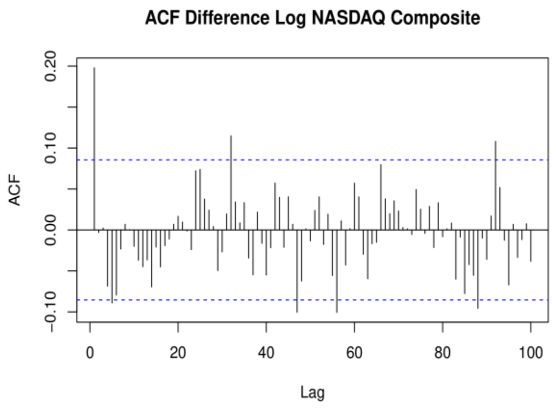


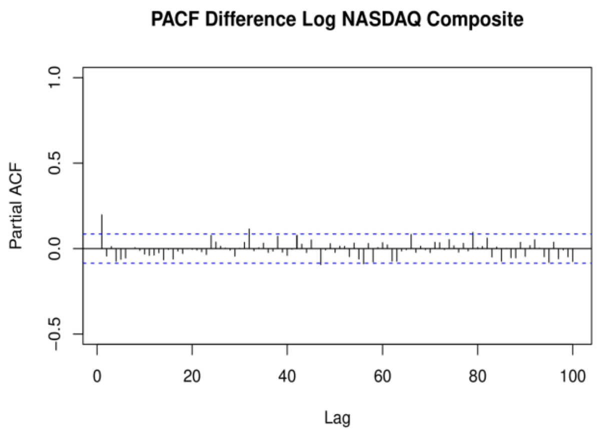
QQ plot of Differenced Log Data



ADF test yields p-value 0.01 and CADF has p-value < 2.2e-16, KPSS test is 0.1. Thus, the data is finally stationary.

ACF and PACF of log differenced data are as follows:

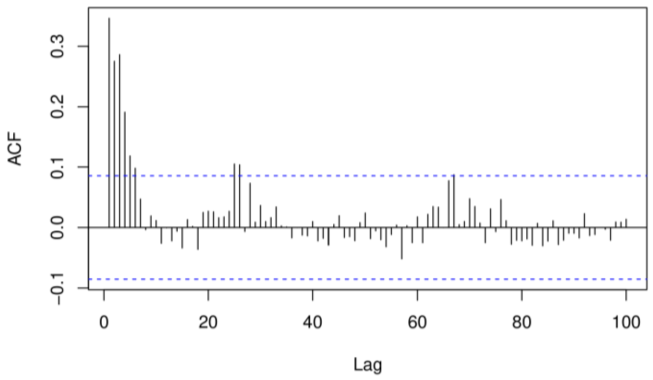




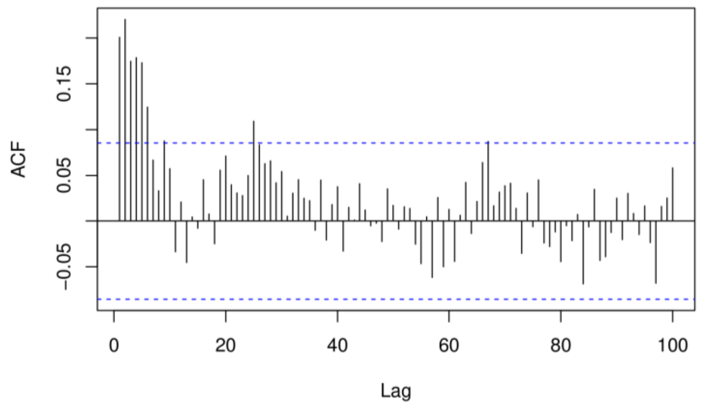
EACF

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o x o o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x x x o o o o o o o o  o  o  o
## 4 x o x o o o o o o o o  o  o  o
## 5 x o x x o x o o o o o  o  o  o
## 6 o x x x x x o o o o o  o  o  o
## 7 o o o o o x o o o o o  o  o  o
```

ACF of Square of Data



ACF of Absolute of Data



We will explore the following models:

ARIMA(0,1,0) with AIC -2657.588

ARIMA(1,1,1) with AIC -2680.652

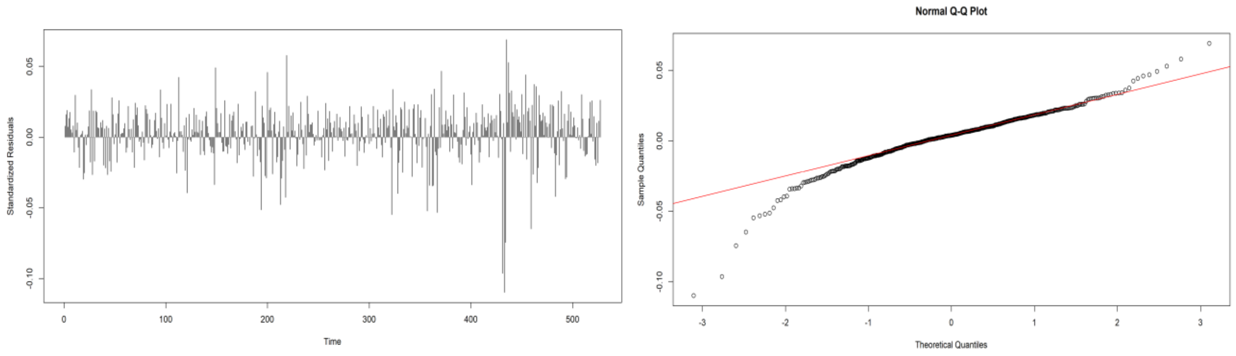
ARIMA(1,1,2) with AIC -2678.765

ARIMA(3,1,4) with AIC -2683.625

ARIMA(2,1,1) with AIC -2679.604

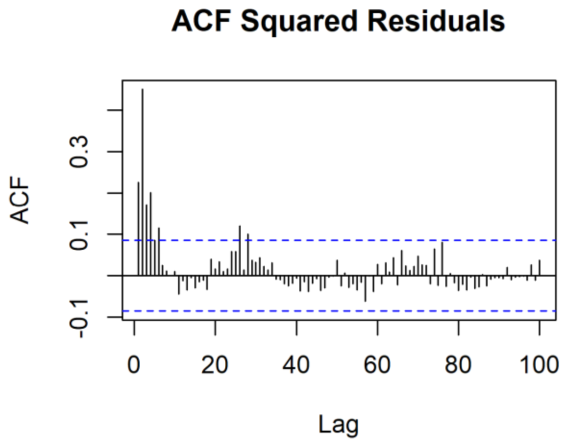
ARIMA(2,1,2) with AIC -2677.633

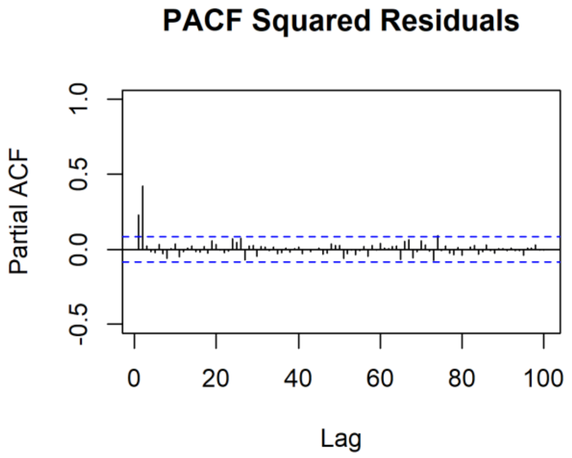
Among these we choose the model ARIMA(3,1,4) with the least AIC value and the perform modelling. The residuals obtained are plotted along with their QQ-plot as follows:



We perform the ARCH test, which displays p-value less than 0.05. We reject the null hypothesis which implies that the residuals have ARCH effect and therefore GARCH model can be fitted.

ACF and PACF of Squares of Residuals are also plotted to check for GARCH models.





1.4 Statistical Conclusions

We fit the residuals that we got from the previous ARIMA model to univariate eGARCH models with Skewed t-distribution (sstd) distribution for the following models:

eGARCH(4,1) with BIC -5.525844

eGARCH(5,1) with BIC -5.505368

eGARCH(2,1) with BIC -5.537102

eGARCH(2,2) with BIC -5.523385

eGARCH(2,3) with BIC -5.540685

We notice that above eGARCH model produces higher variances than necessary and thus we choose GJR-GARCH model that regulates this variance and works the best for the residual data. Thus, we fit the residuals that we got from the previous ARIMA model to univariate GJR-GARCH models with Skewed t-distribution (sstd) distribution for the following models which were picked for having the best BIC scores:

gjrGARCH(2,1) with BIC -5.476419

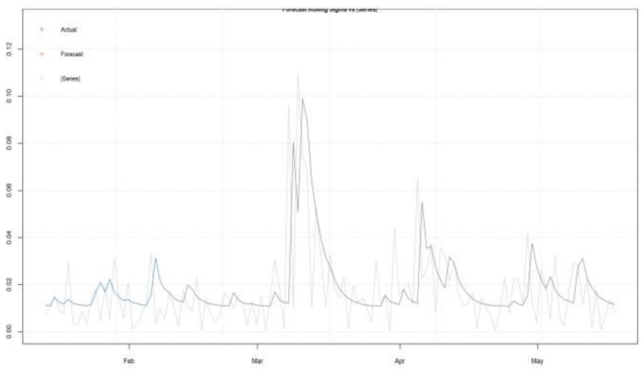
gjrGARCH(2,3) with BIC -5.468274

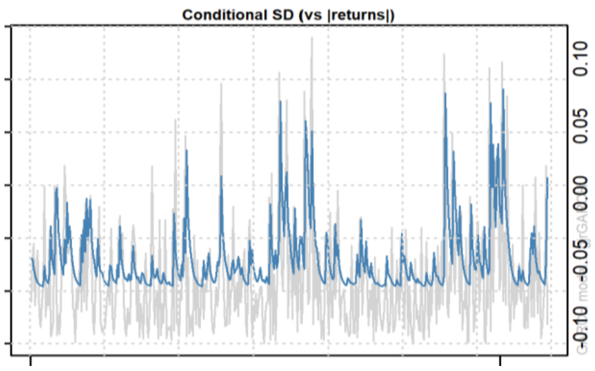
The GJR-GARCH model incorporates a stylized characteristic that the GARCH or eGARCH model does not, which is the empirically observed finding that negative shocks in the previous observation have a bigger impact on the variance at current time than positive shocks. The rise in risk is caused by higher leverage created by a negative shock, hence this imbalance is called the leverage effect.

The final gjrGARCH model chosen for the prediction based on the lowest BIC value is the gjrGARCH(2,1).

Following are the forecasts of the model which handle the variances of stock data in the right manner:

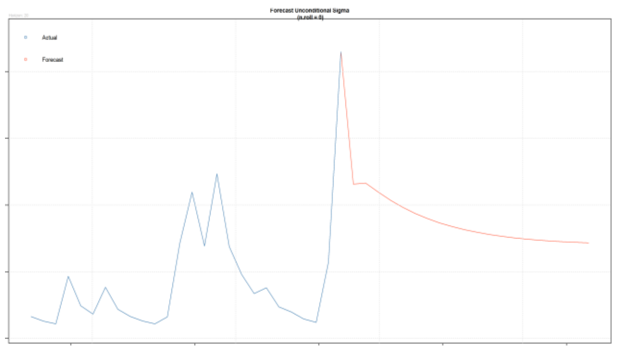
Forecast of the rolling sigma vs the series Conditional Standard Deviation vs the Diff Log of Series

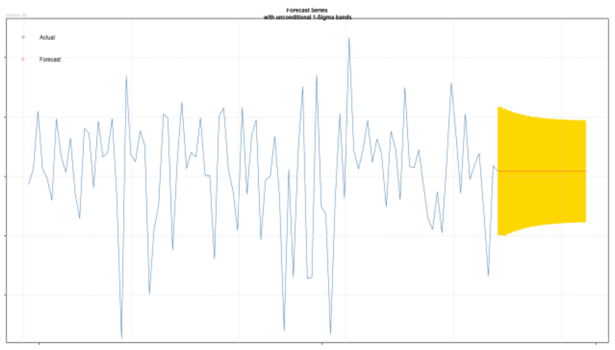




Forecast Unconditional Sigma

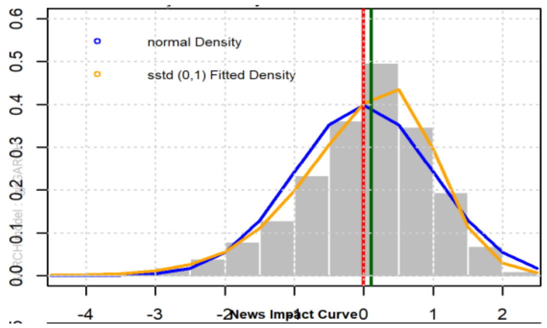
Forecast series with unconditional 1-sigma bands

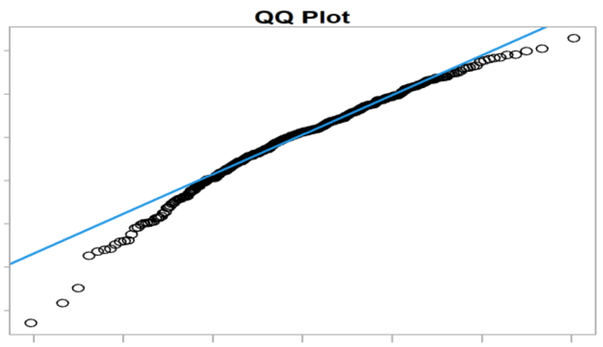




Histogram of the Residuals

QQ-plot of the Residuals





The mean of the residuals is 0.0002594118, their standard deviation is 2.449795e-07 and likelihood of the model is 1127.744.

We also find that the ACF of the residuals have no significant correlation.

The Ljung-Box test produces a p-value of 0.5932 which is greater than 0.05.

Shapiro-Wilk normality test gives a p-value of 3.022e-07 which is less than 0.05, thus the residuals are normal.

Thus, we can conclude that the model we have chosen is a good fit for the data.

1.5 CONCLUSIONS IN THE CONTEXT OF THE PROBLEM:

Stock data can be highly volatile and may require two different models to achieve the best results. We have used the ARIMA(3,1,4) model based on the least AIC scores on the log of the data, the residuals here display ARCH effects and hence we proceed with the GARCH modelling of these residuals. The GJR-GARCH(2,1) model is fitted on the residuals of the ARIMA model, based on the least BIC scores. The residuals obtained by the GARCH model are checked for goodness of fit and any further correlations. The residuals show no significant correlation and display a perfect fit as proved using the Ljung-Box test.

The conditional variance handles the various highly volatile parts of the data. GARCH models depend on the previous values and a forecast of the smaller time period provides the best results. The overall volatility of the model is based on the fact that it gives higher preference to the recent observations of the overall data. Thus the conditional variance increases when a highly volatile region of the data is encountered. Therefore, the predictions adjust accordingly to accommodate these periods of high volatility. In the future, the procedure followed in this paper can be used for most of the stock data analysis in the field of finance or any other field where highly volatile data is encountered. GARCH models provide a great way for analyzing and predicting a wide range of data that cannot be fitted using regular ARIMA models.

2 SEASONAL MARS WEATHER

2.1 MOTIVATION AND INTRODUCTION OF THE PROBLEM

In this paper seasonal dataset is analyzed based on seasonality and mean.

NASA's Curiosity Rover first arrived on Mars at the Gale Crater, near the equator of Mars' southern hemisphere. Despite the fact that its primary aim was to search for conditions where life may have existed and kept track of the weather on a daily basis. On Mars, the weather varies a lot throughout the day. The air temperature cannot be consistent every day, just like it cannot be consistent on Earth.

This paper examines the Mars Weather Data using time-series methodologies in statistics and proposes a plausible model that can explain the data's seasonal time series pattern persuasively. This time series model will be able to accurately forecast the future temperature conditions which could help plan future missions to Mars.

2.2 DATA

2.2.1 Dataset Description

The Dataset is collected from Kaggle datasets repository -
https://www.kaggle.com/datasets/imkrkannan/mars-weather-data

A Martian year is divided into 12 months, just as ours. A Martian month, on the other hand, can last anywhere from 46 to 67 sols (Martian days). The longest month is 3 (67 sols), and the shortest month is

9. (46 sols) Months distinguish seasonal changes on Mars. In the southern hemisphere (where the Curiosity rover is stationed), autumn begins in month one, followed by winter in month four, spring in month seven, and summer in month ten. 4 martian years are recorded in the given dataset.

Weather conditions on Mars from Sol 1 (August 7, 2012 on Earth) to Sol 1895 (February 27, 2018 on Earth) at the Gale Crater on the planet's southern equator. Each row shows weather data gathered at one spot on a particular sol (Martian day). This information was collected and relayed by the Curiosity Rover's Rover Environmental Monitoring Station (REMS). NASA's Mars Science Laboratory and the Centro de Astrobiologa made the information public (CSIC-INTA).

Header	**Description**	**Data Type**
id	The identification number of a single transmission	number
terrestrial_date	The date on Earth (formatted as month/day/year or m/dd/yy).	date
sol	The number of elapsed sols (Martian days) since Curiosity landed on Mars.	number
ls	The solar longitude or the Mars-Sun angle, measured from the Northern Hemisphere. In the Northern Hemisphere, the spring equinox is when ls = 0. Since Curiosity is in the Southern Hemisphere, the following ls values are of importance: <br/>• ls = 0: autumnal equinox <br/>• ls = 90 : winter solstice <br/>• ls = 180 : spring equinox <br/>• ls = 270 : summer solstice	number
month	The Martian Month. Similarly to Earth, Martian time can be divided into 12 months. Helpful information can be found here.	text
min_temp	The minimum temperature (in °C) observed during a single Martian sol.	number
max_temp	The maximum temperature (in °C) observed during a single Martian sol.	number
pressure	The atmospheric pressure (Pa) in Curiosity's location on Mars.	number
wind_speed	The average wind speed (m/s) measured in a single sol. *Note:* Wind Speed data has not be transmitted to Earth since Sol 1485. Missing values are coded as NaN.	number
atmo_opacity	Description of the overall weather conditions on Mars for a given sol based on atmospheric opacity (e.g., Sunny).	text

2.2.2 Data Preprocessing

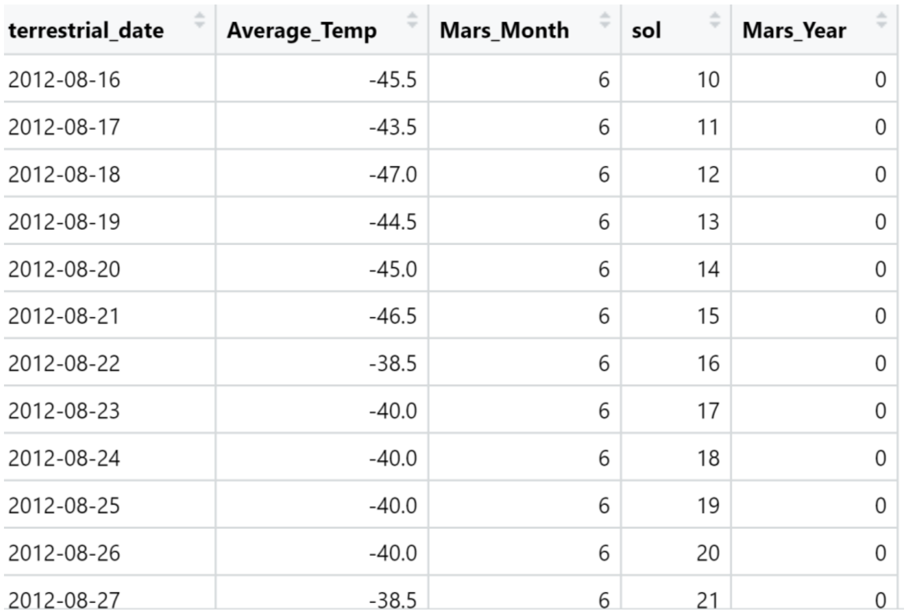
The challenge here is to convert Martian days, months, and years into understandable time series data in terms of Earth months to determine their seasonality. Sol counts the number of Martian days since the rover landed on mars. For the purpose of analysis, based on real seasons on Mars, we convert the

Martian days, months and years into Earth scale and treat them like time period on Earth. This can be easily done due to the 12-month cycle that Mars follows. This helps in clear understanding of seasonal patterns, apply the already present time series tools/plots and thus provide accurate forecasts even for Mars weather.

The datasets contain NA/null values which are replaced with the previous values to maintain continuity and provide the best results. The missing values in the beginning of the dataset are dropped.

The min_temp and max_temp are averaged into Average_temp variable (in °C) and saved in the dataset's Dataframe. The given month is renamed to Mars_Month and the martian year is calculated using the data present in the Mars_Month and the Sol field, 4 martian years are observed. The martian years are numbered as 0,1,2 and 3.

Since the days of the month are quite irregular ranging from 46 to 67 sols/Martian days, we aggregate the data into monthly data.

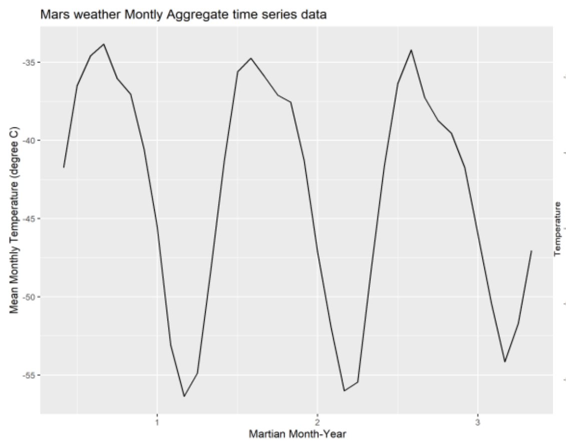


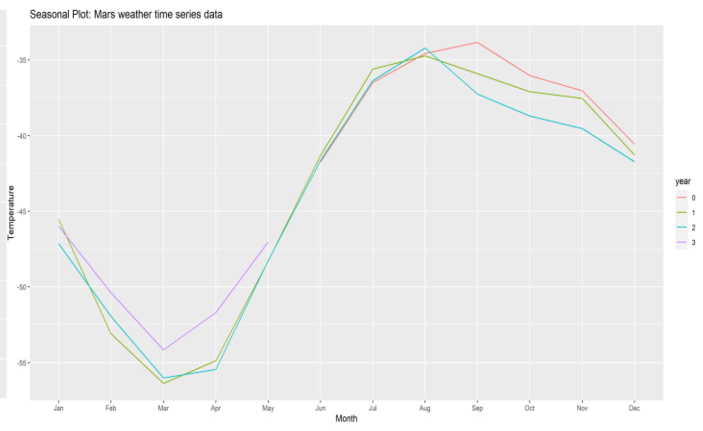
terrestrial_date	Average_Temp	Mars_Month	sol	Mars_Year
2012-08-16	-45.5	6	10	0
2012-08-17	-43.5	6	11	0
2012-08-18	-47.0	6	12	0
2012-08-19	-44.5	6	13	0
2012-08-20	-45.0	6	14	0
2012-08-21	-46.5	6	15	0
2012-08-22	-38.5	6	16	0
2012-08-23	-40.0	6	17	0
2012-08-24	-40.0	6	18	0
2012-08-25	-40.0	6	19	0
2012-08-26	-40.0	6	20	0
2012-08-27	-38.5	6	21	0

2.3 Box-Jenkins model

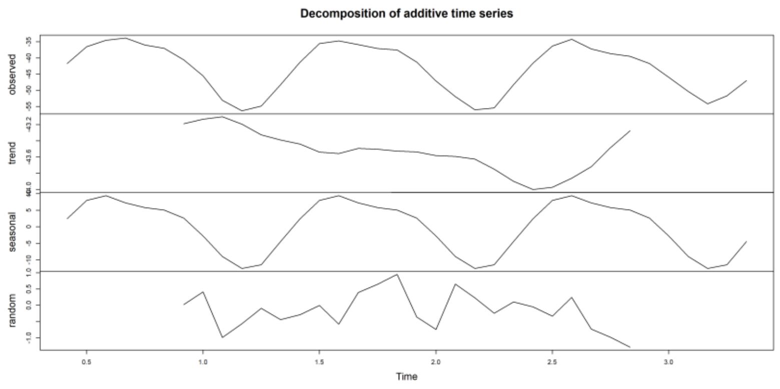
We follow the Box Jenkins model to explore the various methods to identify models used to forecast the data.

Plot of Monthly Aggregate Temperature time series data. We observe a high seasonality and barely noticeable trend. Seasonal Plot shows clear seasonality for the 4 years – 0,1,2,3



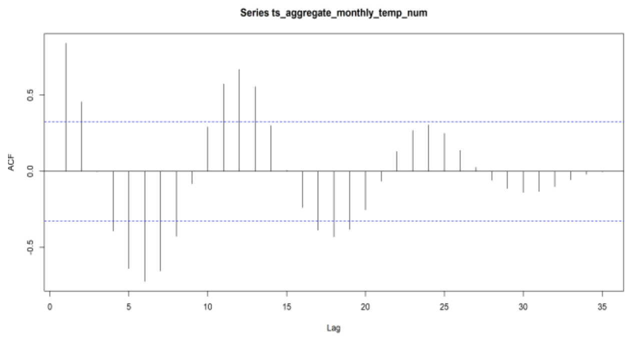


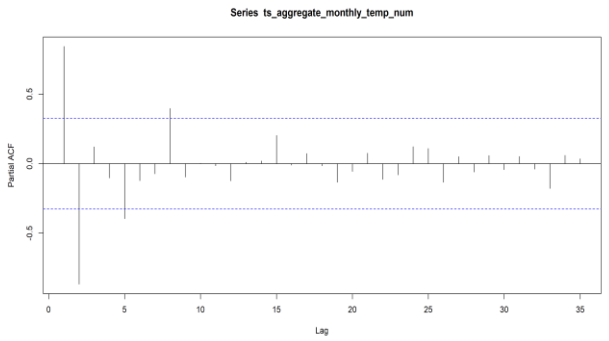
Plot decomposotion is as follows:



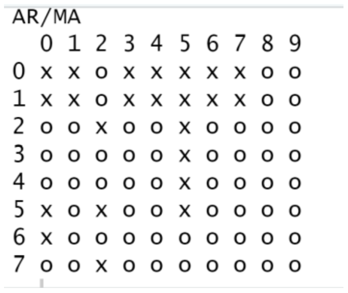
CADF, Augmented Dickey-Fuller Test (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test all indicate that the data is stationary, however, we can clearly observe the seasonal pattern in the data which shows that the data is not fully stationary and mean keeps changing for each seasonal cycle.

ACF and PACF plots are as follows:





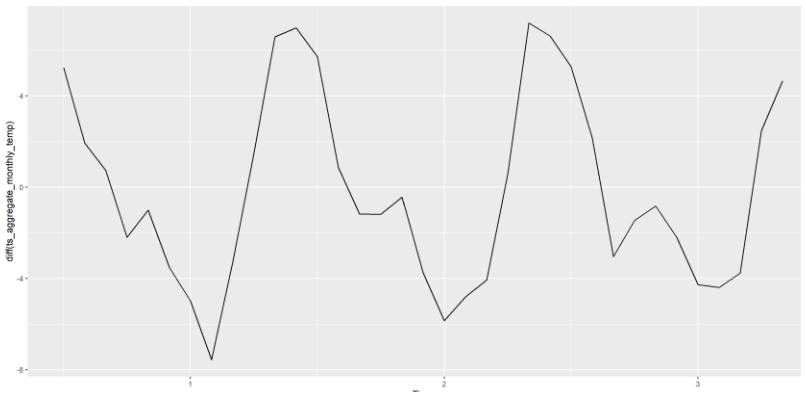
EACF



```
AR/MA
  0 1 2 3 4 5 6 7 8 9
0 x x o x x x x x o o
1 x x o x x x x x o o
2 o o x o o x o o o o
3 o o o o o x o o o o
4 o o o o o x o o o o
5 x o x o o x o o o o
6 x o o o o o o o o o
7 o o x o o o o o o o
```

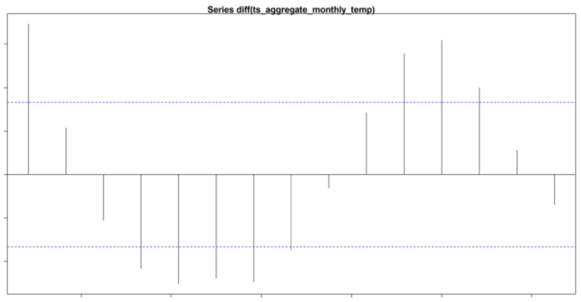
We observe a High variance of 54.353980

Making the data stationary: We make the data stationary by taking the first difference of the data. We still see a pattern of seasonality in the data, it has variance of 17.2 which is still high.

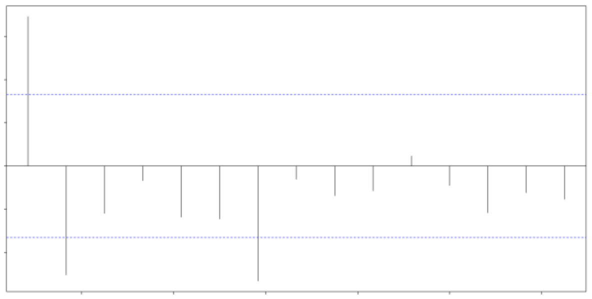


ADF test gives a p-value of 0.1964 which is less than 0.05, thus the data is still not stationary.

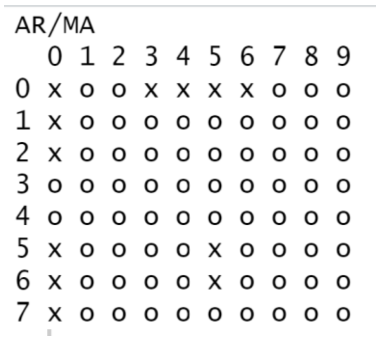
ACF plot



PACF Plot



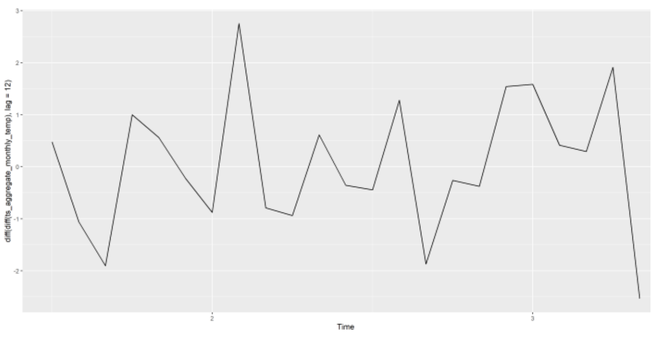
EACF



```
AR/MA
  0 1 2 3 4 5 6 7 8 9
0 x o o x x x x o o o
1 x o o o o o o o o o
2 x o o o o o o o o o
3 o o o o o o o o o o
4 o o o o o o o o o o
5 x o o o o x o o o o
6 x o o o o x o o o o
7 x o o o o o o o o o
```

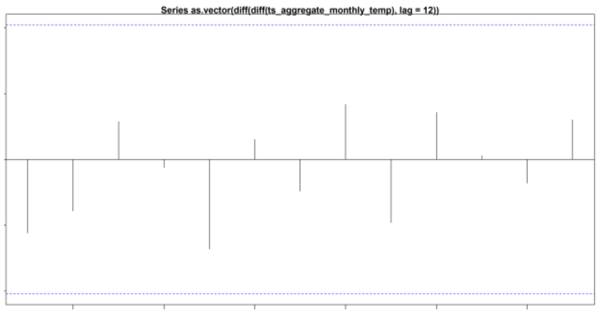
We observe a SARMA(2,1)(1,3)[12] model or a SARIMA(3,1)(1,3)[12] model from the plots above.

We further remove seasonality by differencing it at lag 12, we get an almost 0.03 mean and 1.7 variance. Plot is as follows:

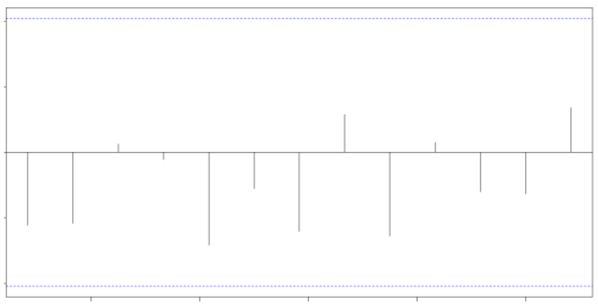


CADF test and KPSS test display stationarity in the data.

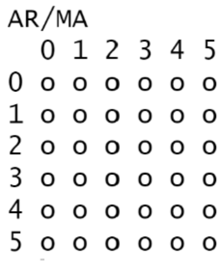
ACF plot



PACF plot



EACF



```
AR/MA
  0 1 2 3 4 5
0 o o o o o o
1 o o o o o o
2 o o o o o o
3 o o o o o o
4 o o o o o o
5 o o o o o o
```

We notice that ACF, PACF and EACF suggest that it could be a SARIMA(0,1,0)(0,1,0)[12] model.

We also explore SARIMA(2,1,1)(1,1,3)[12] and SARIMA(3,1,1)(1,1,3)[12] models that we observed earlier.

2.4 STATISTICAL CONCLUSIONS

Using ML method for SARIMA model, we get AIC=32.32 and log likelihood = -15.16 for SARIMA(0,1,0)(0,1,0)[12]

We get AIC=41.24 for SARIMA(2,1,1)(1,1,3)[12] using ML method

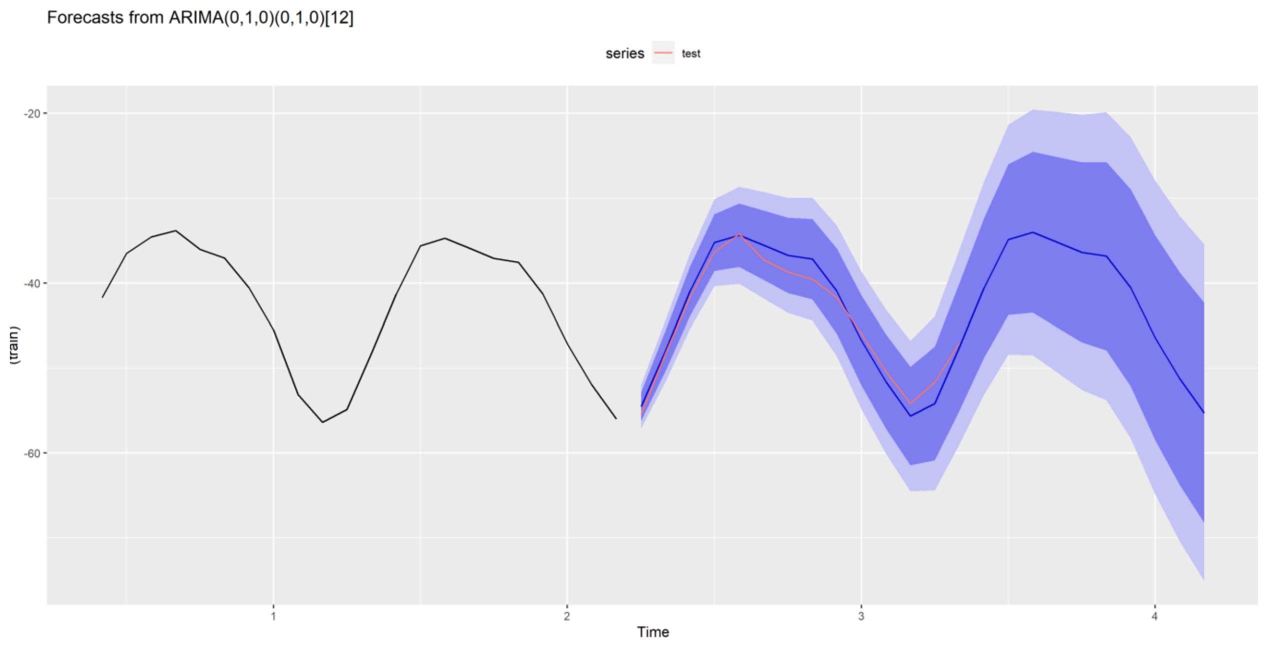
We get AIC=41.63 for SARIMA(3,1,1)(1,1,3)[12] using ML method

Taking into consideration the model with the least AIC score, SARIMA(0,1,0)(0,1,0)[12] model is chosen. The ML method and the CSS method, yield the same results for the Mars Temperature monthly aggregate data.

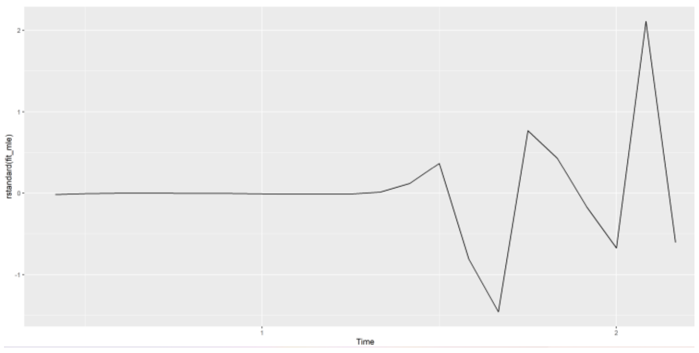
Train and test data is split from the original data, and the model is fitted on the training data.

We forecast the SARIMA(0,1,0)(0,1,0)[12] model using the test data that is set aside.

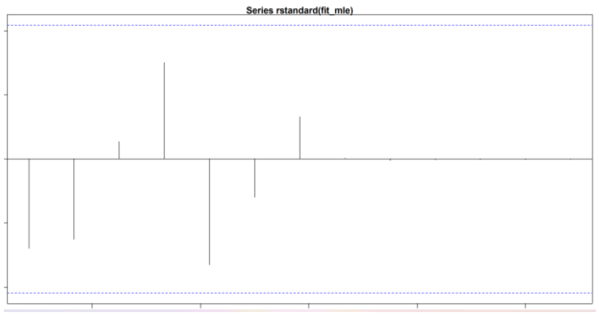
We observe that the accuracy of the forecasted data (blue line) is impeccable with respect to the testing set (red line), plotted along with training data (black line). The confidence interval is the purple area surrounding the dark blue line which follows an accurate seasonal pattern.

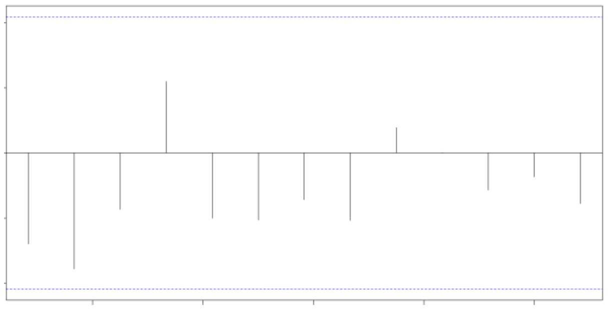


Residual Analysis: Plot of the residuals is as follows



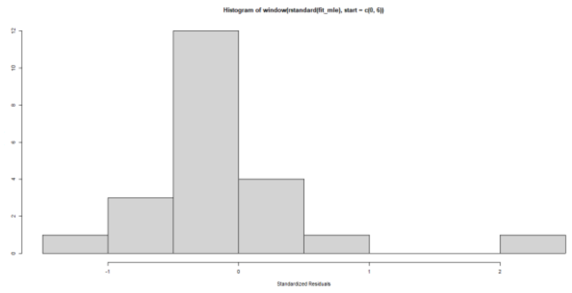
ACF plot PACF plot

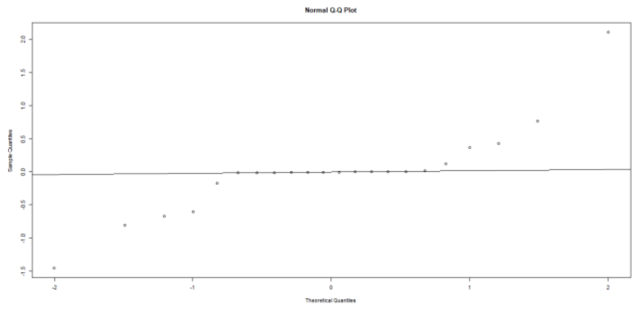




The residuals show no correlation among them.

Histogram and QQ-plot of residuals are as follows which follows a distribution:





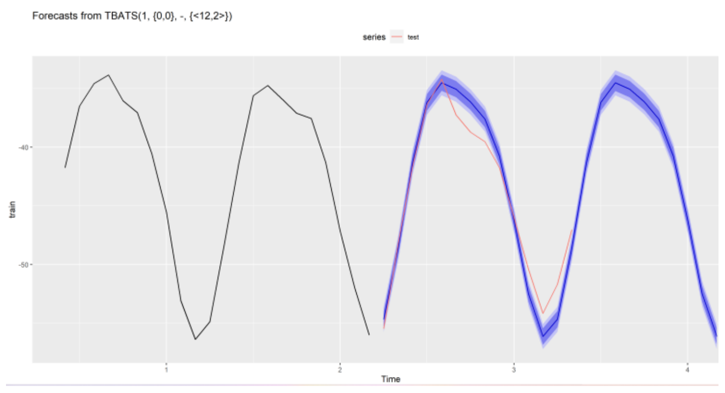
Box-Ljung test p-value 0.95 is greater than 0.05, thus the residuals are uncorrelated.

Shapiro-Wilk test for normality gives a p-value of 0.0005, that is less than 0.05, thus the residuals are normal.

Thus, we can conclude that the model is a good fit for the given seasonal mars data.

To cross check using another method we use the library tbats:

The output from this method yields the exact same result as the previous method as displayed in the below forecast of the values:



This confirms SARIMA(0,1,0)(0,1,0)[12] model and the accuracy of the forecast done using this model.

2.5 Conclusions in the context of the problem:

Weather data of any planet usually follows a seasonal pattern, however, the challenge we faced in this paper was to correctly interpret Mars weather data in terms of Earth times and use the appropriate time series analysis techniques to plot, fit and forecast this data.

This model can be used to produce accurate forecast in the future during various explorations in order to understand the expected whether prior to any future launches to Mars. The techniques used here can be generally applied to other data sources pertaining to other fields such as forest fires, sales, heat and electricity consumption. This paper provides useful insights on the identification and use of SARIMA models.

3 Bibliography:

https://pudding.cool/2018/01/mars-weather/

https://www.kaggle.com/datasets/imkrkannan/mars-weather-data

https://darksky.net/dev

https://mars.nasa.gov/msl/weather/

https://finance.yahoo.com/quote/%5EIXIC/history?p=%5EIXIC

The Use of ARCH/GARCH Models in Applied Econometrics- by RobertEngle

Time Series Analysis with ARIMA – ARCH/GARCH model in R - L---Stern Group, Ly Pham

Pattern Recognition-Product Sales Analysis Using SARIMA Model in TS Forecasting-R. Kulkarni, M. Rane

Study of effectiveness of time series modeling Arima in forecasting stock prices – Mondal, Shit, Goswami

4 Appendix

4.1 Code for NASDAQ Composite Non-Seasonal Data

```
install.packages('quantmod')

install.packages('aTSA')

library(quantmod)

library(xts)

library(TSA)

library(CADFtest)

library(forecast)

library(stats)

library(FinTS)

library(rugarch)

library(fBasics)

library(PerformanceAnalytics)


startDate = as.Date("2012-01-01") #Specify period of time we are interested in

endDate = as.Date("2022-01-01")


getSymbols("^IXIC", from = startDate, to = endDate, warnings = FALSE, auto.assign = TRUE)


head(IXIC)

str(IXIC)

nasdaq.AdjClose <- IXIC$IXIC.Adjusted

week <- format(as.Date(index(IXIC)), "%U")

year <- format(as.Date(index(IXIC)), "%Y")


any(is.na(nasdaq.AdjClose))


#Aggregating data on a Weekly basis for each year

aggregate_weekly<-aggregate( IXIC$IXIC.Adjusted ~ week + year , IXIC$IXIC.Adjusted , mean )
```

```
View(aggregate_weekly)

nasdaq.AdjClose <- aggregate_weekly$IXIC.Adjusted


ts_nasdaq.AdjClose <- ts(nasdaq.AdjClose, start = c(2012,1), frequency=52)


#Plotting the original Stock Adjusted Close Price

plot(ts_nasdaq.AdjClose,type='l',main='NASDAQ Composite Stock Price Time series')


adf.test(ts_nasdaq.AdjClose) #p-value greater than 0.05-> accept null-> data is Not stationary

CADFtest(ts_nasdaq.AdjClose)


#Computes the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for the null hypothesis that x is level or trend stationary

kpss.test(ts_nasdaq.AdjClose, null = c("Level", "Trend"), lshort = TRUE) #p-value<0.05, reject null, Not Stationary


ArchTest(ts_nasdaq.AdjClose) #reject null, Arch effects seen


## Differencing the series

diff.nasdaq=diff(ts_nasdaq.AdjClose)


any(is.na(diff.nasdaq))


#Plot differences of original series

plot(diff.nasdaq,type='l',main='Difference NASDAQ Composite')


adf.test(diff.nasdaq) #p-value less than 0.05-> reject null-> diff data is stationary


#Take log of original series and plot the log price

log.nasdaq=log(ts_nasdaq.AdjClose)

any(is.na(log.nasdaq))


plot(log.nasdaq,type='l',main='Log NASDAQ Composite')


adf.test(log.nasdaq) #p-value greater than 0.05-> accept null-> log data is Not stationary
```

```
#Differencing log price and plot

difflog.nasdaq=diff(log.nasdaq)

plot(difflog.nasdaq,type='l',main='Difference Log NASDAQ Composite')

any(is.na(difflog.nasdaq))


adf.test(difflog.nasdaq) #p-value less than 0.05-> reject null-> diff log data is stationary

CADFtest(difflog.nasdaq)

kpss.test(difflog.nasdaq, null = c("Level", "Trend"), lshort = TRUE) #p-value>0.05, accept null, It is Stationary


###

#Histogram

hist(difflog.nasdaq, xlab="Weekly Diff Log of stock prices", prob=TRUE, main="Histogram for Weekly Diff Log of stock prices")

xfit<-seq(min(difflog.nasdaq),max(difflog.nasdaq),length=40)

yfit<-dnorm(xfit,mean=mean(difflog.nasdaq),sd=sd(difflog.nasdaq))

lines(xfit, yfit, col="blue", lwd=2)


#QQ-plot

qqnorm(difflog.nasdaq)

qqline(difflog.nasdaq, col = 2)


## Model Identification


# ACF and PACF of Log NASDAQ Composite

acf.nasdaq=acf(ts_nasdaq.AdjClose,main='ACF NASDAQ Composite',lag.max=100) #Decreases but does not die, needs
differencing

pacf.nasdaq=pacf(ts_nasdaq.AdjClose,main='PACF NASDAQ Composite',lag.max=100,ylim=c(-0.5,1)) #Cuts off after lag 1


# ACF and PACF of Log NASDAQ Composite

acf.nasdaq=acf(log.nasdaq,main='ACF NASDAQ Composite',lag.max=100) #Decreases but does not die, needs differencing

pacf.nasdaq=pacf(log.nasdaq,main='PACF NASDAQ Composite',lag.max=100,ylim=c(-0.5,1)) #Cuts off after lag 1


#ACF and PACF of Differenced log NASDAQ Composite
```

```
acf.nasdaq=acf(difflog.nasdaq,main='ACF Difference Log NASDAQ Composite',lag.max=100) # 1

pacf.nasdaq=pacf(difflog.nasdaq,main='PACF Difference Log NASDAQ Composite',lag.max=100,ylim=c(-0.5,1)) #3 or 2 or 1

eacf(difflog.nasdaq) #2,1 and 2,2 and 6,0 and 5,1 and 4,1 and 4,3 and 5,4


acf.nasdaq=acf(difflog.nasdaq^2,main='ACF of Square of Difference Log NASDAQ Composite',lag.max=100) # 1 or 8 or 6

acf.nasdaq=acf(abs(difflog.nasdaq),main='ACF of Absolute of Difference Log NASDAQ Composite',lag.max=100) # 1 or 6 or 7 or 8

#MA observed in ACF of Squared and abs diff log data


pacf.nasdaq=pacf(nasdaq.AdjClose^2,main='PACF Square of NASDAQ Composite',lag.max=100,ylim=c(-0.5,1)) #Cuts off after lag 1

pacf.nasdaq=pacf(abs(nasdaq.AdjClose),main='PACF Abs of NASDAQ Composite',lag.max=100,ylim=c(-0.5,1)) #Cuts off after lag 1



#arima_x = auto.arima(difflog.nasdaq)

#arima_x #1,0,2 zero mean


#arima_n = auto.arima(nasdaq.AdjClose)

#arima_n #orig=0,2,1 #log=1,1,2 #diff=0,1,1


#Ljung Box test on squared values of the stock price returns

#H0: null hypothesis is there exists no autocorrelation

Box.test(difflog.nasdaq^2, type="Ljung") #reject null, there is autocorrelation


basicStats(difflog.nasdaq)


## ARIMA models

arima010=stats::arima(log.nasdaq,order=c(0,1,0),method='ML')

summary(arima010)


arima111=stats::arima(log.nasdaq,order=c(1,1,1),method='ML')

summary(arima111)
```

```
arima112=stats::arima(log.nasdaq,order=c(1,1,2),method='ML')
summary(arima112)

arima314=stats::arima(log.nasdaq,order=c(3,1,4),method='ML')
summary(arima314) #best

arima211=stats::arima(log.nasdaq,order=c(2,1,1),method='ML')
summary(arima211)

arima212=stats::arima(log.nasdaq,order=c(2,1,2),method='ML')
summary(arima212)



arima010$aic
arima111$aic
arima112$aic
arima314$aic
arima211$aic
arima212$aic

Box.test(rstandard(arima314), lag=20, type = 'Ljung-Box') #p-value=0.98>0.05, accept null, thus model is just fine, fits well
plot(residuals(arima314),type='h',ylab='Standardized Residuals')
qqnorm(residuals(arima314)) ;qqline(residuals(arima314), col='red')

res.arima314=arima314$res
squared.res.arima314=res.arima314^2

acf.squared314=acf(squared.res.arima314,main='ACF Squared Residuals',lag.max=100) #1 or 4
pacf.squared314=pacf(squared.res.arima314,main='PACF Squared Residuals',lag.max=100,ylim=c(-0.5,1)) #AR(1) or AR(2)

acf.squared314=acf(abs(res.arima314),main='ACF abs Residuals',lag.max=100) #1 or 6
pacf.squared314=pacf(abs(res.arima314),main='PACF abs Residuals',lag.max=100,ylim=c(-0.5,1)) #AR(1) or AR(2) or AR(3)
```

```
ArchTest(res.arima314) #p-value<0.05->reject null-> has ARCH effect and therefore GARCH model can be fitted



###GARCH modelling


#Univariate eGARCH model with sstd distribution GARCH(4,1)


ug_spec41 <- ugarchspec(mean.model=list(armaOrder=c(0,0)),
                   variance.model = list(model = 'eGARCH',garchOrder = c(4,1)),
                   distribution.model = 'sstd')


#Model estimation
ugfit41 = ugarchfit(spec = ug_spec41, data = res.arima314, out.sample = 125)
ugfit41


#Univariate eGARCH model with sstd distribution GARCH(5,1)


ug_spec51 <- ugarchspec(mean.model=list(armaOrder=c(0,0)),
                   variance.model = list(model = 'eGARCH',garchOrder = c(5,1)),
                   distribution.model = 'sstd')


#Model estimation
ugfit51 = ugarchfit(spec = ug_spec51, data = res.arima314, out.sample = 125)
ugfit51



#Univariate eGARCH model with sstd distribution GARCH(2,1)


ug_spec21 <- ugarchspec(mean.model=list(armaOrder=c(0,0)),
                  variance.model = list(model = 'eGARCH',garchOrder = c(2,1)),
                  distribution.model = 'sstd')


#Model estimation
```

```
ugfit21 = ugarchfit(spec = ug_spec21, data = res.arima314, out.sample = 125)

ugfit21


#Univariate eGARCH model with sstd distribution GARCH(2,2)


ug_spec22 <- ugarchspec(mean.model=list(armaOrder=c(0,0)),

                    variance.model = list(model = 'eGARCH',garchOrder = c(2,2)),

                    distribution.model = 'sstd')


#Model estimation

ugfit22 = ugarchfit(spec = ug_spec22, data = res.arima314, out.sample = 125)

ugfit22

infocriteria(ugfit22)


par(mar=c(1,1,1,1))

plot(ugfit22, which='all')


#Univariate eGARCH model with sstd distribution GARCH(2,3)


ug_spec23 <- ugarchspec(mean.model=list(armaOrder=c(0,0)),

                    variance.model = list(model = 'eGARCH',garchOrder = c(2,3)),

                    distribution.model = 'sstd')


#Model estimation

ugfit23 = ugarchfit(spec = ug_spec23, data = res.arima314, out.sample = 125)

ugfit23


#Univariate gjrGARCH model with sstd distribution GARCH(2,2)


ug_spec21_gjr <- ugarchspec(mean.model=list(armaOrder=c(0,0)),

                    variance.model = list(model = 'gjrGARCH',garchOrder = c(2,1)),

                    distribution.model = 'sstd')
```

```
#Model estimation

ugfit21_gjr = ugarchfit(spec = ug_spec21_gjr, data = res.arima314, out.sample = 125)

ugfit21_gjr



#Univariate gjrGARCH model with sstd distribution GARCH(2,3)


ug_spec23_gjr <- ugarchspec(mean.model=list(armaOrder=c(0,0)),

                    variance.model = list(model = 'gjrGARCH',garchOrder = c(2,3)),

                    distribution.model = 'sstd')


#Model estimation

ugfit23_gjr = ugarchfit(spec = ug_spec23_gjr, data = res.arima314, out.sample = 125)

ugfit23_gjr


###




######--------------------------------

#We choose the best model


infocriteria(ugfit41) #BIC= -5.525844

infocriteria(ugfit51) #BIC= -5.505368

infocriteria(ugfit21) #BIC= -5.537102

infocriteria(ugfit22) #BIC= -5.523385

infocriteria(ugfit23) #BIC= -5.540685


#Best 2

infocriteria(ugfit21_gjr) #BIC= -5.476419

infocriteria(ugfit23_gjr) #BIC= -5.468274
```

```
#Choosing the model with the least BIC

ugfit_best = ugfit21_gjr

plot(ugfit_best, which='all')



ugfore_best <- ugarchforecast(ugfit_best, n.ahead = 20, n.roll=100)
ugfore_best



plot(ugfore_best, which='all')


# Estimated standardized returns
stdret <- residuals(ugfit_best, standardize = TRUE)
chart.Histogram(stdret, methods = c("add.normal", "add.density"),
        colorset=c("gray","red","blue"))

chart.QQPlot(stdret)
# Goodness of fit for the mean prediction
e <- residuals(ugfit_best)
mean(e^2)

# Goodness of fit for the variance prediction
d <- e^2 - sigma(ugfit_best)^2
mean(d^2)

# Compute the likelihood
likelihood(ugfit_best)

acf(stdret) #No significant correlations observed
Box.test(stdret, type = "Ljung-Box") #pvalue >0.05, good fit
shapiro.test(coredata(stdret)) #p-value 3.022e-07<0.05, residuals are normal
```

```
######Estimate all models
variance.models <- c("sGARCH", "gjrGARCH", "eGARCH")
distribution.models <- c("norm", "std", "sstd")
c <- 1
for (variance.model in variance.models) {
  for (distribution.model in distribution.models) {
    garchspec <- ugarchspec(mean.model = list(armaOrder = c(0, 0)),
                        variance.model = list(model = variance.model),
                        distribution.model = distribution.model)
    garchfit <- ugarchfit(data = ts_nasdaq.AdjClose, spec = garchspec)
    if (c==1) {
      msigma <- sigma(garchfit)
    } else {
      msigma <- merge(msigma, sigma(garchfit))
    }
    c <- c + 1
  }
}

plot(msigma)
plot(ts_nasdaq.AdjClose)

avesigma <- xts(rowMeans(msigma), order.by = time(msigma))
plot(avesigma)
```

4.2 Code for Mars Seasonal Weather Data

```
## Clear all variables in R
rm(list=ls())
## Clear all graphs
dev.off()
```

```
# Install required packages

install.packages('fpp2')

install.packages('TSstudio')

install.packages('Jmisc')


#Import required libraries

library(fpp2) #forecasting package

library(TSstudio)

library(tidyr)

library(CADFtest)

library(fBasics)

library(TSA)

library(forecast)

library(Jmisc)


# Retrieve the dataset

data <- read.csv("C:/Users/Chrisia/OneDrive - stevens.edu/Documents/Stevens 2nd Semester/MA 641 Time Series Analysis
1/Project/Seasonal_MarsWeather/R code Mars Weather/mars-weather.csv",header=T)


#View the data and attributes

str(data)

summary(data)

data <- data[order(data$sol),]

View(data)


# calculate average temperature using mean of high and low temperatures

avg_temp = list((data$min_temp+data$max_temp)/2)


#Add the average temperature column to the original dataset

new_data <- cbind(data, avg_temp = avg_temp)

colnames(new_data)[11] <- "Average_Temp"
```

```
View(new_data)


#Create a new dataset with only the columns required for time series analysis: terrestrial_date and Average_Temp

new_data1 = new_data[c("terrestrial_date","Average_Temp","month","sol")]

colnames(new_data1)[3] <- "Mars_Month"


#Filling missing NA time series data with previous observations

new_data1 <- na.locf(new_data1)

View(new_data1)

# Data Preprocessing: Add a Martian year column which is missing in the data

# New years start from sol 1019 and 1688

# Set Year 0 = Sol 10 to 350, Year 1 = Sol 351 to 1018; Year 2 = Sol 1019 to 1687; Year 3 = Sol 1688 to 1977

new_data1$Mars_Year[new_data1$sol<351] <- 0

new_data1$Mars_Year[new_data1$sol>=351  & new_data1$sol<1019] <- 1

new_data1$Mars_Year[new_data1$sol>=1019 & new_data1$sol<1688] <- 2

new_data1$Mars_Year[new_data1$sol>=1688] <- 3


#Use only the numeric month values

new_data1$Mars_Month <- as.numeric(gsub("\\D", "", new_data1$Mars_Month))


View(new_data1)


#Save the data as time series entry

ts_temp <- ts(new_data1$Average_Temp, start = c(1,6), frequency=668) #657 days in a year in mars


#Performing Preliminary Analysis


#Time series plot

autoplot(ts_temp) + ggtitle("Mars weather time series data") + ylab("Temperature (degree C)") + xlab("Martian Days (sol)")


#  Aggregate 'Average_Temp' on months and year and get mean

aggregate_monthly_temp<-aggregate( Average_Temp ~ Mars_Month + Mars_Year , new_data1 , mean )

View(aggregate_monthly_temp)
```

```
ts_aggregate_monthly_temp <- ts(aggregate_monthly_temp$Average_Temp, start = c(0,6), frequency=12)


autoplot(ts_aggregate_monthly_temp) + ggtitle("Mars weather Montly Aggregate time series data") + ylab("Mean Monthly Temperature (degree C)") + xlab("Martian Month-Year")


ggseasonplot(ts_aggregate_monthly_temp) + ggtitle("Seasonal Plot: Mars weather time series data") + ylab("Temperature")

#Observation: Clear seasonality observed for Mar temperature values


#Converting ts into list of numbers

ts_aggregate_monthly_temp_num <- coredata(ts_aggregate_monthly_temp)

ts_temp_num <- coredata(ts_temp)


decompose_ts <- decompose(ts_aggregate_monthly_temp)

plot(decompose_ts$figure, type='b', las=2, xlab='Month')

plot(decompose_ts)


#Dickey Fuller Tests

#If the p-value is less than or equal to the specified significance level, the null hypothesis is rejected; otherwise, the null hypothesis is not rejected.


#Augmented Dickey-Fuller Test

CADFtest(ts_aggregate_monthly_temp) #p-value is 1.673e-05<0.05, null hypothesis is rejected, consider alternative hypothesis

adf.test(ts_aggregate_monthly_temp) #p-value is 0.04909,

#Result shows that it is stationary but we do further testing


#Kwiatkowski-Phillips-Schmidt-Shin ( KPSS Test)

kpss.test(ts_aggregate_monthly_temp, null= "Level") #level stationary, pvalue is 0.1

kpss.test(ts_aggregate_monthly_temp, null = "Trend") #trend stationary, pvalue is 0.1



#Autocorrelation, Partial Autocorrelation and Extended Autocorrelation Function
```

```
acf(ts_aggregate_monthly_temp_num, lag.max = 36) #3, clear seasonality pattern seen

pacf(ts_aggregate_monthly_temp_num, lag.max = 36) #1

eacf(ts_aggregate_monthly_temp, ar.max = 7, ma.max = 9)

basicStats(ts_aggregate_monthly_temp) #High variance of 54.353980



#Taking FIRST DIFFERENCE -------------------------------------------


autoplot(diff(ts_aggregate_monthly_temp), type='l') #No visible upward trend observed, seasonality still visible


basicStats(diff(ts_aggregate_monthly_temp)) #Variance 17.2


#Dickey Fuller Tests


#Augmented Dickey-Fuller Test

CADFtest(diff(ts_aggregate_monthly_temp), type = "trend") #p-value is 0.02597<0.05, null hypothesis is rejected, consider alternative hypothesis

adf.test(diff(ts_aggregate_monthly_temp)) #p-value is 0.1964>0.05, we cannot reject null hypothesis, thus data is NOT stationary


#Kwiatkowski-Phillips-Schmidt-Shin ( KPSS Test)

kpss.test(diff(ts_aggregate_monthly_temp), null= "Level") #level stationary, pvalue is 0.1

kpss.test(diff(ts_aggregate_monthly_temp), null = "Trend") #trend stationary, pvalue is 0.1


#Autocorrelation, Partial Autocorrelation and Extended Autocorrelation Function

acf(diff(ts_aggregate_monthly_temp)) # 1 and 3s

pacf(diff(ts_aggregate_monthly_temp)) #2 and 1s

eacf(diff(ts_aggregate_monthly_temp), ar.max = 7, ma.max = 9) #0,1

#MA2 2

#Observation: The general upward trend has now disappeared but the strong seasonality is still seen.


#Nonseasonal=MA(3) Seasonal=MA(5), MA(6)

#NS = MA(2) MA(3), s=MA(3) MA(6)
```

```
#FIRST DIFFRENCE and SEASONAL DIFFERENCE ------------------------

autoplot(diff(diff(ts_aggregate_monthly_temp), lag=12), type='l')

basicStats(diff(diff(ts_aggregate_monthly_temp), lag=12)) #Variance reduced to 1.7

#Dickey Fuller Tests

#Augmented Dickey-Fuller Test

CADFtest(diff(diff(ts_aggregate_monthly_temp), lag=12), type = "trend") #p-value is 0.01<0.05, null hypothesis is rejected, consider alternative hypothesis

#adf.test(ts_aggregate_monthly_temp) #p-value is 0.08649>0.05 thus cannot reject null hypothesis, thus data is not stationary

#adf.test(diff(diff(ts_aggregate_monthly_temp), lag=12)) #p-value is 0.2722>0.05, we cannot reject null hypothesis, thus data is NOT stationary

#Kwiatkowski-Phillips-Schmidt-Shin ( KPSS Test)

kpss.test(diff(diff(ts_aggregate_monthly_temp), lag=12), null= "Level") #level stationary, pvalue is 0.1

kpss.test(diff(diff(ts_aggregate_monthly_temp), lag=12), null = "Trend") #trend stationary, pvalue is 0.1

#Autocorrelation, Partial Autocorrelation and Extended Autocorrelation Function

acf(as.vector(diff(diff(ts_aggregate_monthly_temp), lag=12))) #2 , lag.max=36,ci.type='ma'

pacf(diff(diff(ts_aggregate_monthly_temp), lag=12))

eacf(diff(diff(ts_aggregate_monthly_temp_num), lag=12), ar.max = 5, ma.max = 5)

####--------------------

#Splitting test and train data

train <- head(ts_aggregate_monthly_temp, round(length(ts_aggregate_monthly_temp) * 0.6))

h <- length(ts_aggregate_monthly_temp) - length(train)

test <- tail(ts_aggregate_monthly_temp, h)

#Forecasting methods

#SARIMA(0,1,0)(0,1,0)[12] model
```

```
fit_mle00 <- Arima((train), order = c(0,1,0), seasonal = c(0,1,0), method='ML')

fit_mle00 #AIC=32.32


#SARIMA(2,1,1)(1,1,3)[12] model

fit_mle21 <- Arima((train), order = c(2,1,1), seasonal = c(1,1,3), method='ML')

fit_mle21 #AIC=41.24


#SARIMA(3,1,1)(1,1,3)[12] model

fit_mle31 <- Arima((train), order = c(3,1,1), seasonal = c(1,1,3), method='ML')

fit_mle31 #AIC=41.63


#SARIMA(0,1,0)(0,1,0)[12] model

fit_css00 <- Arima((train), order = c(0,1,0), seasonal = c(0,1,0), method='CSS')

fit_css00 #Log-Likelihood=-15.16



best_fit_mle = fit_mle00


#Plotting the forecasted model

forecast_daily <- forecast(best_fit_mle, h=24)

autoplot(forecast_daily)+ autolayer(test) + theme(legend.position = 'top')


#plot(forecast_daily,n1=c(0,6),n.ahead=24,xlab='Year',type='b',ylab='Mars Temperature')


#Residual Analysis - Plot, acf, qqplot, hist, shapiro wilk, Ljung-Box, forecast

#Plot  residuals

autoplot(rstandard(fit_mle))


#ACF and PACF of residuals

acf(rstandard(fit_mle))

pacf(rstandard(fit_mle))


#Histogram of residuals
```

```
win.graph(width=3, height=3,pointsize=8)

hist(window(rstandard(fit_mle),start=c(0,6)),xlab='Standardized Residuals')


# qqplot of residuals

win.graph(width=2.5,height=2.5,pointsize=8)

qqnorm(window(rstandard(fit_mle),start=c(0,6)))

qqline(window(rstandard(fit_mle),start=c(0,6)))


# Ljung-Box test for correlation

Box.test(rstandard(fit_mle), lag=20, type = 'Ljung-Box') #p-value 0.95>0.05, thus the residuals are uncorrelated


#Shapiro-Wilk test for normality

shapiro.test(rstandard(fit_mle)) #p-value 0.0005<0.05, residuals are normal



# Forecasting the model using tbats for cross checking

fit <- tbats(train)

fit #ARMA(0,0) errors, Box-Cox transformation of 1, 2 Fourier pairs with period 12 months

seasonal <- !is.null(fit$seasonal)

paste("The data is seasonal = ",seasonal)

forecast_daily <- forecast(fit, h=24)

autoplot(forecast_daily)+ autolayer(test) + theme(legend.position = 'top')
```