ARIMA model Analysis (Time Series)

Ajinkya Ghodekar

Required Libraries

```
#install.packages("dplyr")
#install.packages("forecast")

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(forecast)

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

Including CSV file

```
rm(list=ls(all=TRUE))

tranData = read.csv(file = "F:\\OneDrive_1_21-06-2019\\transaction_data.csv")
```

Required data from CSV

```
quantity_zero = tranData[tranData$QUANTITY == 0 & tranData$SALES_VALUE > 0,]

summary(quantity_zero)

##  household_key    BASKET_ID              DAY          PRODUCT_ID
##  Min.   :  53   Min.   :2.713e+10   Min.   : 13.0   Min.   :   57059
##  1st Qu.: 505   1st Qu.:3.196e+10   1st Qu.:336.5   1st Qu.:  905799
##  Median :1041   Median :3.317e+10   Median :419.0   Median :  986912
##  Mean   :1148   Mean   :3.422e+10   Mean   :413.1   Mean   : 2431378
##  3rd Qu.:1796   3rd Qu.:3.579e+10   3rd Qu.:534.5   3rd Qu.: 1088048
##  Max.   :2491   Max.   :4.168e+10   Max.   :665.0   Max.   :13945244
##     QUANTITY  SALES_VALUE        STORE_ID        RETAIL_DISC
##  Min.   :0   Min.   :0.0000   Min.   :  289.0   Min.   :0.00000
##  1st Qu.:0   1st Qu.:0.0000   1st Qu.:  323.0   1st Qu.:0.00000
##  Median :0   Median :0.0100   Median :  368.0   Median :0.00000
##  Mean   :0   Mean   :0.2967   Mean   : 2435.4   Mean   :0.05841
##  3rd Qu.:0   3rd Qu.:0.0750   3rd Qu.:  422.5   3rd Qu.:0.00000
##  Max.   :0   Max.   :5.8200   Max.   :31782.0   Max.   :2.09000
##    TRANS_TIME      WEEK_NO       COUPON_DISC COUPON_MATCH_DISC
##  Min.   :  41   Min.   : 3.00   Min.   :0    Min.   :0
##  1st Qu.:1328   1st Qu.:48.50   1st Qu.:0    1st Qu.:0
##  Median :1544   Median :61.00   Median :0    Median :0
##  Mean   :1555   Mean   :59.76   Mean   :0    Mean   :0
##  3rd Qu.:1850   3rd Qu.:77.00   3rd Qu.:0    3rd Qu.:0
##  Max.   :2253   Max.   :96.00   Max.   :0    Max.   :0

quantity_gt_zero = tranData[!(tranData$QUANTITY == 0),]

summary(quantity_gt_zero)

##  household_key    BASKET_ID              DAY          PRODUCT_ID
##  Min.   :   1   Min.   :2.698e+10   Min.   :  1.0   Min.   :   25671
##  1st Qu.: 655   1st Qu.:3.014e+10   1st Qu.:220.0   1st Qu.:  916767
##  Median :1271   Median :3.254e+10   Median :373.0   Median : 1027102
##  Mean   :1271   Mean   :3.356e+10   Mean   :371.1   Mean   : 2833218
##  3rd Qu.:1914   3rd Qu.:3.560e+10   3rd Qu.:526.0   3rd Qu.: 1131438
##  Max.   :2500   Max.   :4.185e+10   Max.   :677.0   Max.   :18120301
##     QUANTITY      SALES_VALUE         STORE_ID      RETAIL_DISC
##  Min.   :    1   Min.   :  0.000   Min.   :    1   Min.   :-180.0000
##  1st Qu.:    1   1st Qu.:  1.290   1st Qu.:  330   1st Qu.:  -0.6900
##  Median :    1   Median :  2.000   Median :  370   Median :  -0.0500
##  Mean   :  101   Mean   :  3.117   Mean   : 3071   Mean   :  -0.5436
##  3rd Qu.:    1   3rd Qu.:  3.490   3rd Qu.:  422   3rd Qu.:   0.0000
##  Max.   :89638   Max.   :840.000   Max.   :34280   Max.   :   3.9900
##    TRANS_TIME      WEEK_NO      COUPON_DISC        COUPON_MATCH_DISC
##  Min.   :   0   Min.   : 1.0   Min.   :-55.93000   Min.   :-7.700000
##  1st Qu.:1307   1st Qu.:32.0   1st Qu.:  0.00000   1st Qu.: 0.000000
##  Median :1614   Median :54.0   Median :  0.00000   Median : 0.000000
##  Mean   :1562   Mean   :53.7   Mean   : -0.01253   Mean   :-0.002919
```

```
##  3rd Qu.:1844   3rd Qu.:76.0   3rd Qu.:  0.00000   3rd Qu.: 0.000000
##  Max.   :2359   Max.   :97.0   Max.   :  0.00000   Max.   : 0.000000
```

```
t = select(quantity_gt_zero,WEEK_NO,SALES_VALUE,QUANTITY,RETAIL_DISC,COUPON_D
ISC,COUPON_MATCH_DISC,RETAIL_DISC)
```

```
summary(t)
```

```
##     WEEK_NO      SALES_VALUE         QUANTITY      RETAIL_DISC
##  Min.   : 1.0   Min.   :  0.000   Min.   :    1   Min.   :-180.0000
##  1st Qu.:32.0   1st Qu.:  1.290   1st Qu.:    1   1st Qu.:  -0.6900
##  Median :54.0   Median :  2.000   Median :    1   Median :  -0.0500
##  Mean   :53.7   Mean   :  3.117   Mean   :  101   Mean   :  -0.5436
##  3rd Qu.:76.0   3rd Qu.:  3.490   3rd Qu.:    1   3rd Qu.:   0.0000
##  Max.   :97.0   Max.   :840.000   Max.   :89638   Max.   :   3.9900
##   COUPON_DISC        COUPON_MATCH_DISC
##  Min.   :-55.93000   Min.   :-7.700000
##  1st Qu.:  0.00000   1st Qu.: 0.000000
##  Median :  0.00000   Median : 0.000000
##  Mean   : -0.01253   Mean   :-0.002919
##  3rd Qu.:  0.00000   3rd Qu.: 0.000000
##  Max.   :  0.00000   Max.   : 0.000000
```

Formula

```
t$Actual_price = (t$SALES_VALUE - (t$RETAIL_DISC + t$COUPON_MATCH_DISC)/t$QUA
NTITY)
```

Time Series Data

```
ti = select(t,WEEK_NO,Actual_price)

u = aggregate( Actual_price ~ WEEK_NO , ti, sum)

sum(is.na(u))
```

```
## [1] 0
```

Explore the data

```
tsdata<- ts(u$Actual_price, frequency = 4)

start(tsdata)
```

```
## [1] 1 1
```

```
end(tsdata)
```

```
## [1] 25  1
```

```r
class(tsdata)
```

```
## [1] "ts"
```

```r
frequency(tsdata)
```

```
## [1] 4
```

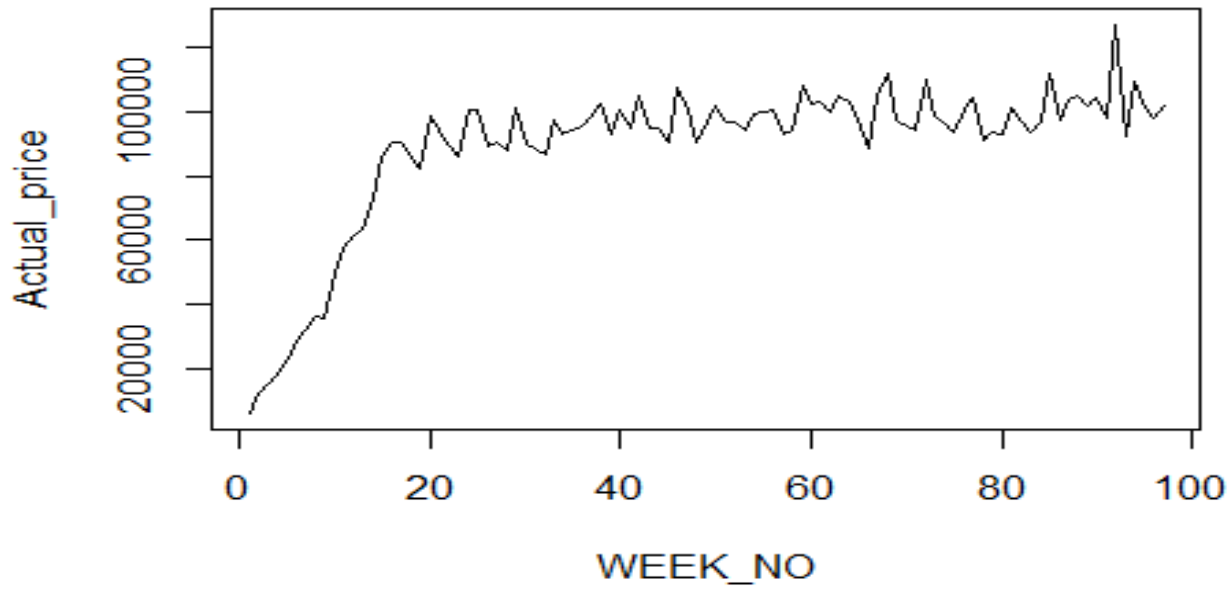
```r
sum(is.na(tsdata))
```

```
## [1] 0
```

```r
summary(tsdata)
```

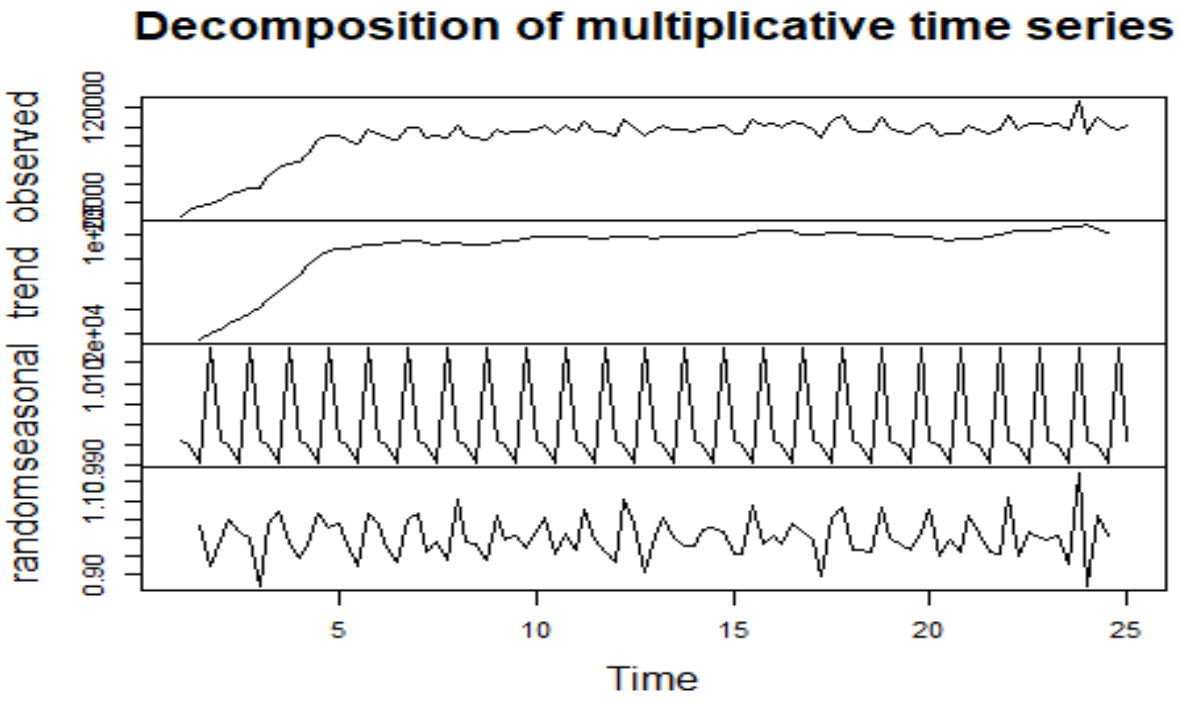
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6065   90181   95349   88791  100855  127147
```

```r
plot(u,type = "l")
```

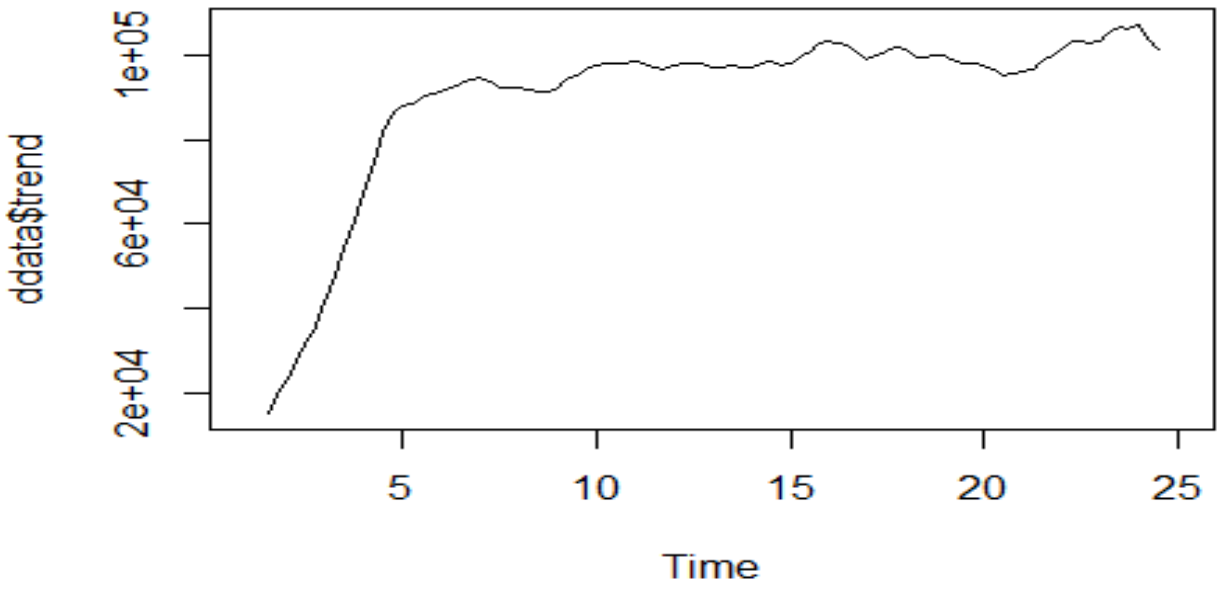


Multiplicative time series

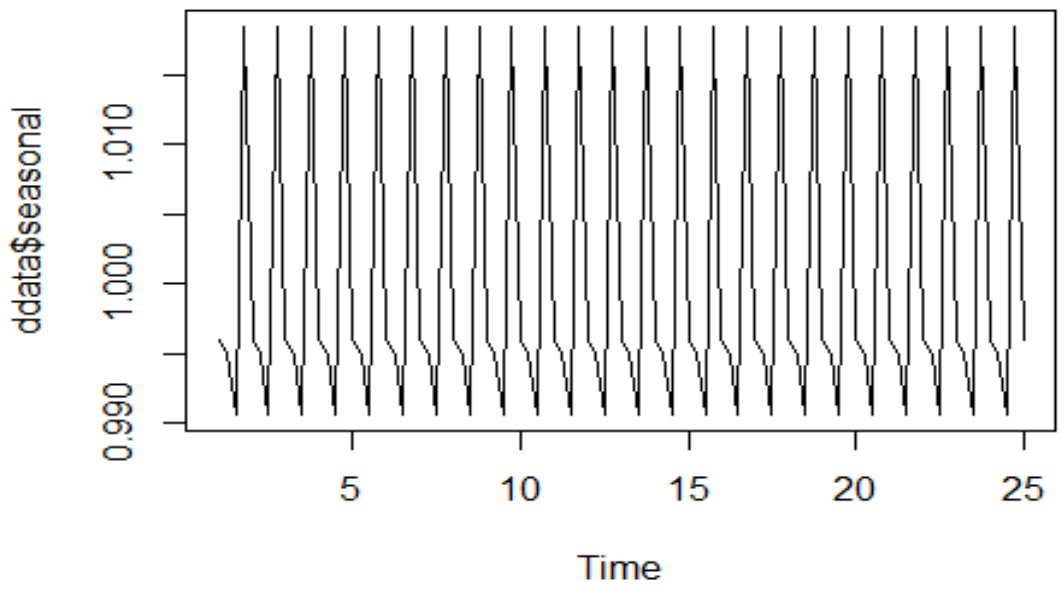
```r
ddata<- decompose(tsdata, "multiplicative")

plot(ddata)
```



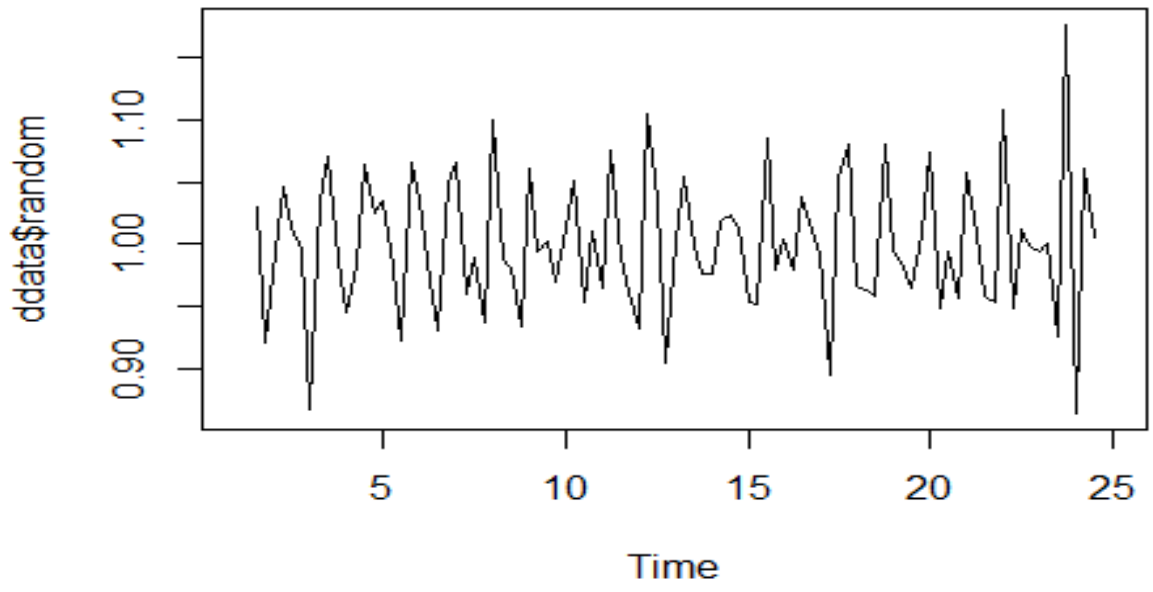

```
plot(ddata$trend)
```



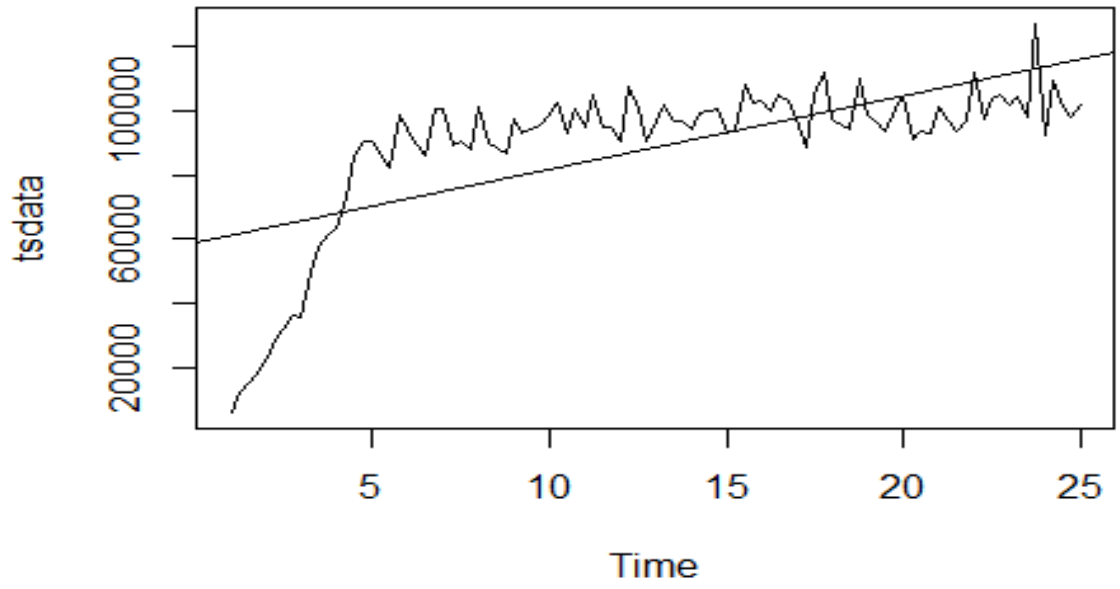

```
plot(ddata$seasonal)
```




```r
plot(ddata$random)
```



```r
plot(tsdata)

abline(reg = lm(tsdata ~ time(tsdata)))
```




```r
cycle(tsdata)
```

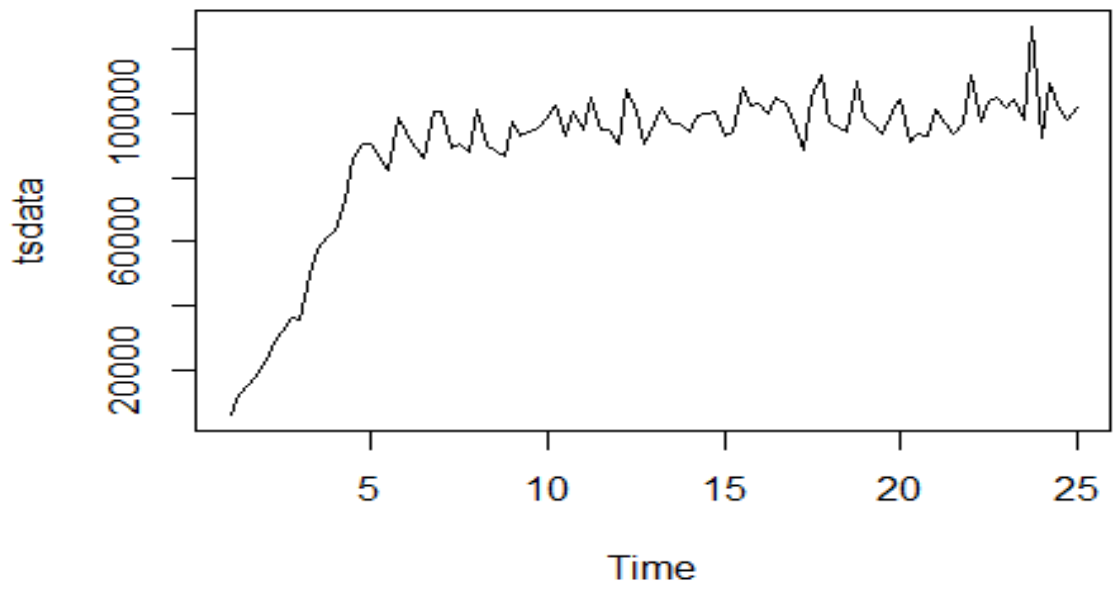
```
##    Qtr1 Qtr2 Qtr3 Qtr4
## 1     1    2    3    4
## 2     1    2    3    4
## 3     1    2    3    4
## 4     1    2    3    4
## 5     1    2    3    4
## 6     1    2    3    4
## 7     1    2    3    4
## 8     1    2    3    4
## 9     1    2    3    4
## 10    1    2    3    4
## 11    1    2    3    4
## 12    1    2    3    4
## 13    1    2    3    4
## 14    1    2    3    4
## 15    1    2    3    4
## 16    1    2    3    4
## 17    1    2    3    4
## 18    1    2    3    4
## 19    1    2    3    4
## 20    1    2    3    4
```

```
## 21    1    2    3    4
## 22    1    2    3    4
## 23    1    2    3    4
## 24    1    2    3    4
## 25    1
```

Box plot of a cycle

```
boxplot(tsdata ~ cycle(tsdata) , xlab = "Quater" , ylab= "Sales Value", main
= "Quaterly sales value")
```


```
plot(tsdata)
```



Creating ARIMA Model

```
mymodel <- auto.arima(ts(u$Actual_price[0:95]))

mymodel

## Series: ts(u$Actual_price[0:95])
## ARIMA(0,2,2)
##
## Coefficients:
##           ma1     ma2
##       -1.6089  0.7573
## s.e.   0.0750  0.0910
##
## sigma^2 estimated as 55637305:  log likelihood=-962.01
## AIC=1930.02   AICc=1930.29   BIC=1937.62

auto.arima(ts(u$Actual_price[0:95]) , ic = 'aic' , trace = TRUE )

##
##  ARIMA(2,2,2)                    : 1932.161
##  ARIMA(0,2,0)                    : 2054.973
##  ARIMA(1,2,0)                    : 2002.355
##  ARIMA(0,2,1)                    : 1964.374
##  ARIMA(1,2,2)                    : 1931.972
##  ARIMA(0,2,2)                    : 1930.022
```

```
##  ARIMA(0,2,3)                    : 1931.928
##  ARIMA(1,2,1)                    : 1946.12
##  ARIMA(1,2,3)                    : 1933.854
##
##  Best model: ARIMA(0,2,2)
```

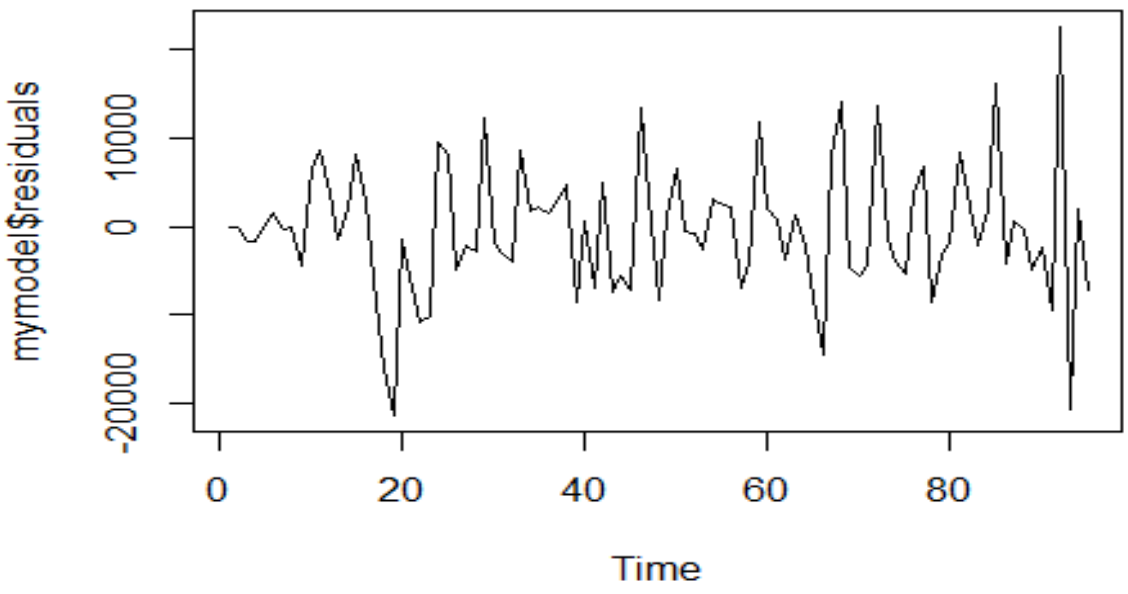
```
## Series: ts(u$Actual_price[0:95])
## ARIMA(0,2,2)
##
## Coefficients:
##           ma1     ma2
##       -1.6089  0.7573
## s.e.   0.0750  0.0910
##
## sigma^2 estimated as 55637305:  log likelihood=-962.01
## AIC=1930.02   AICc=1930.29   BIC=1937.62
```

Required Libraries

```
#install.packages("tseries")

library(tseries)
```

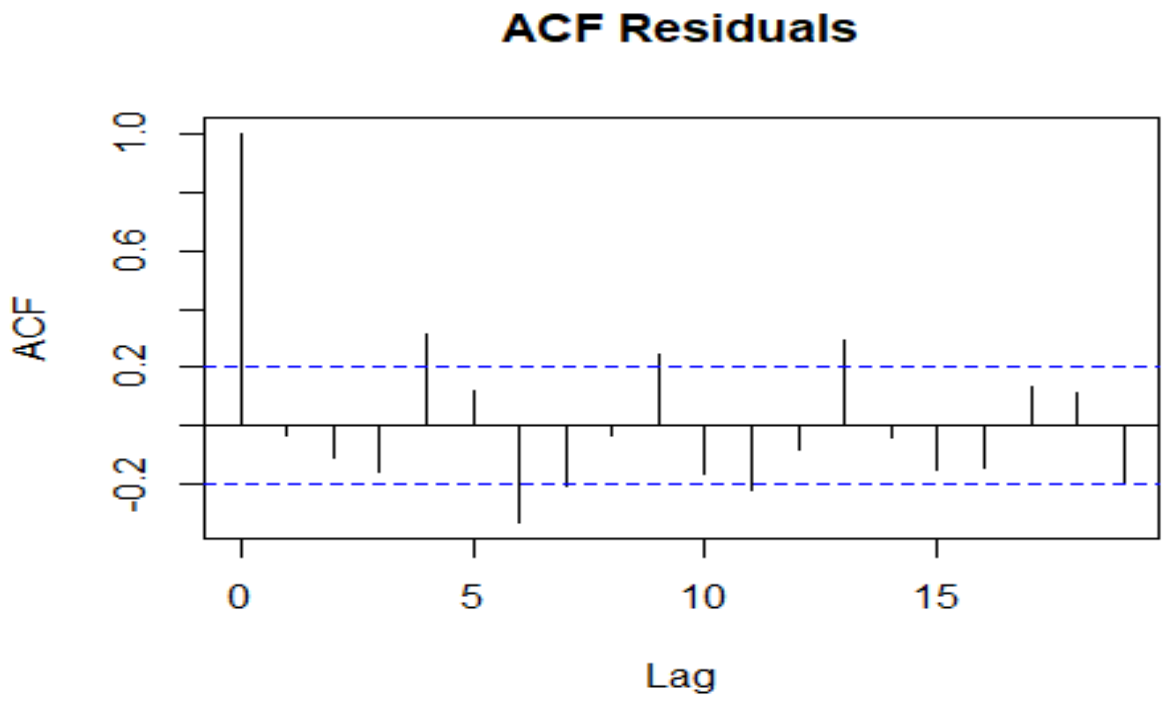
Plot the Residuals

```
plot(mymodel$residuals)
```



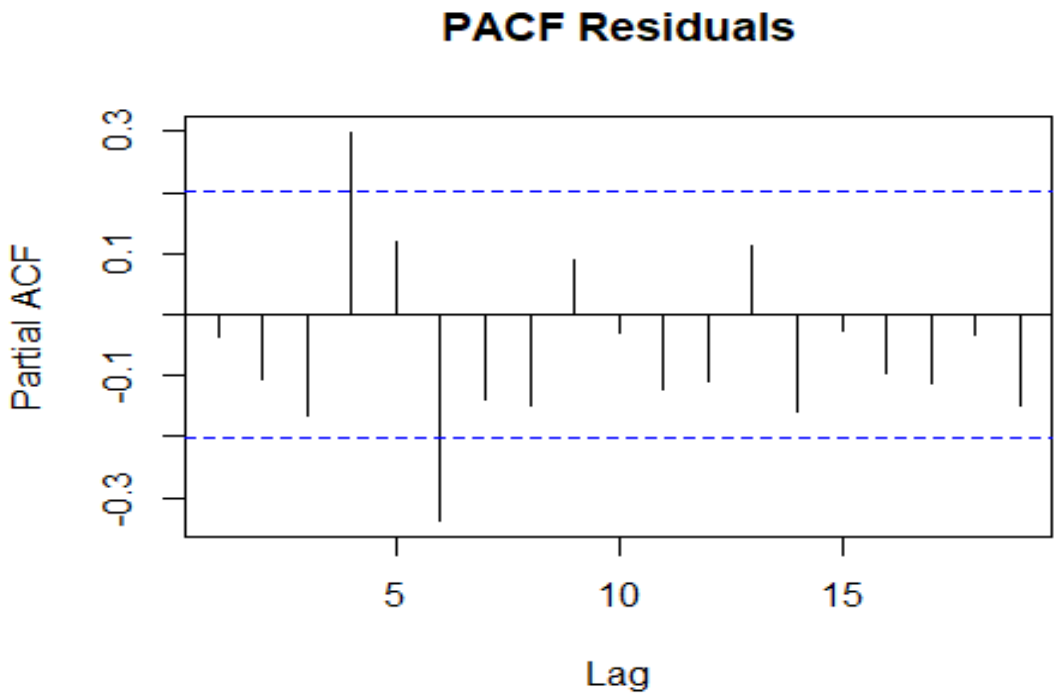
Autocorrelation Function

```
acf(ts(mymodel$residuals), main = "ACF Residuals")
```



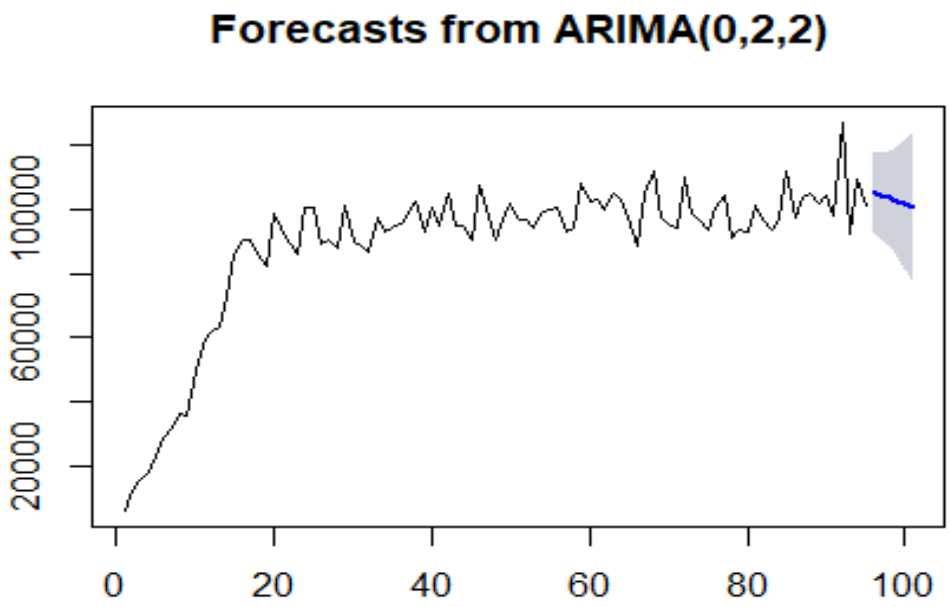
Partial Autocorrelation Function

```
pacf(ts(mymodel$residuals), main = "PACF Residuals")
```



Forecast prediction for Week _No 98

```
myforecast <- forecast(mymodel,level = c(90),h=6)

plot(myforecast)
```




```
myforecast
```

```
##     Point Forecast    Lo 90    Hi 90
##  96       105531.6 93262.62 117800.7
##  97       104565.4 91391.32 117739.5
##  98       103599.1 88855.69 118342.6
##  99       102632.9 85644.76 119621.0
## 100       101666.6 81820.57 121512.7
## 101       100700.4 77466.64 123934.1
```

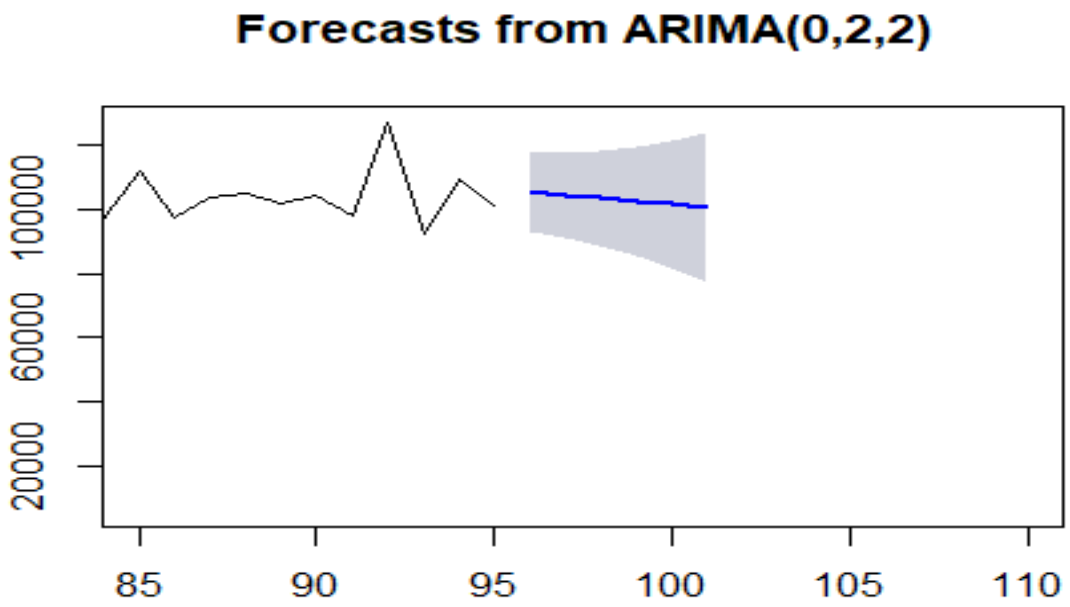
See the Forecasted values

```
myforecast$mean
```

```
## Time Series:
## Start = 96
## End = 101
## Frequency = 1
## [1] 105531.6 104565.4 103599.1 102632.9 101666.6 100700.4
```

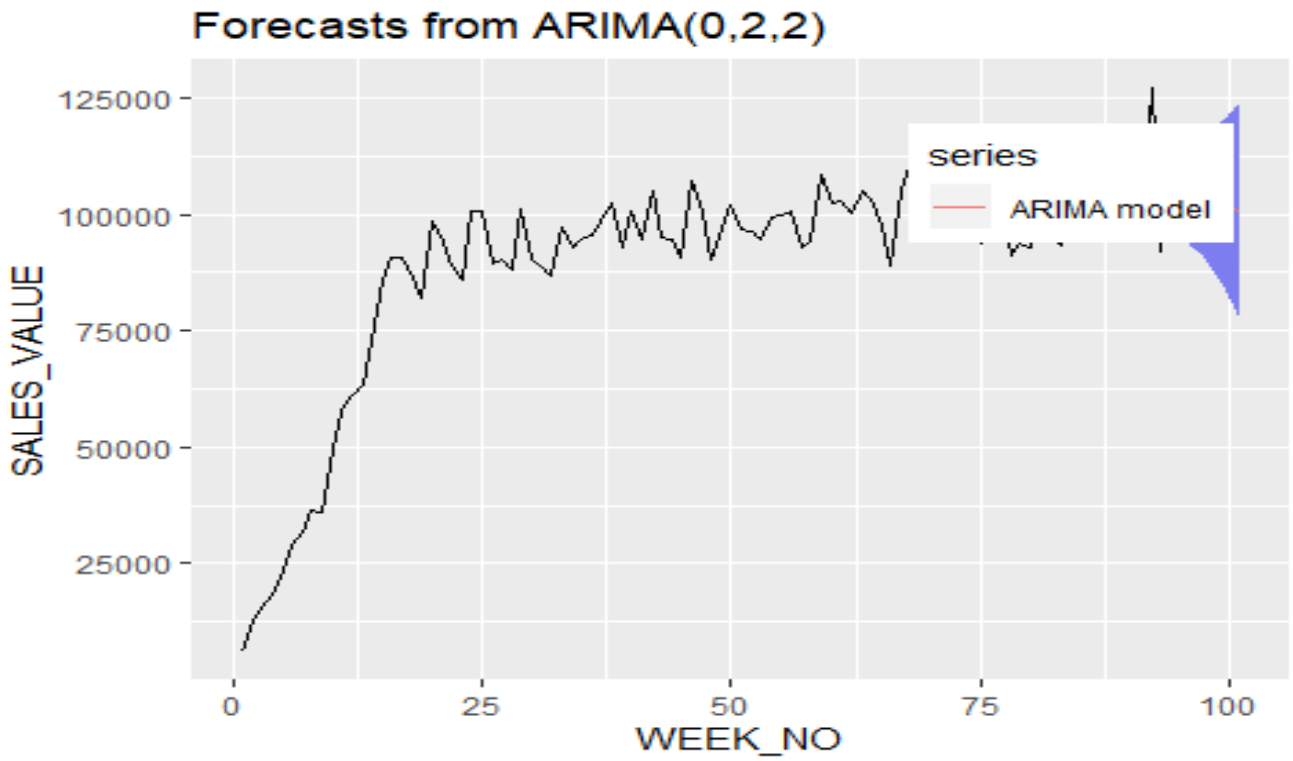
Plot last observations and the forecast

```
plot(myforecast, xlim = c(85, 110))
```



Comparision plot for 2 models

```
library(ggplot2)
autoplot(myforecast)+
  forecast::autolayer(myforecast$mean,
                      series = 'ARIMA model') +
  xlab('WEEK_NO') +
  ylab('SALES_VALUE') +
  guides(
    colour = guide_legend(
      Title = 'Forecast Method')) +
  theme(
    legend.position = c(0.8, 0.8))
```



Box test

```
Box.test(mymodel$residuals,lag = 4, type = "Ljung-Box")

##
##  Box-Ljung test
##
## data:  mymodel$residuals
## X-squared = 13.648, df = 4, p-value = 0.008507

Box.test(mymodel$residuals,lag = 1, type = "Ljung-Box")

##
##  Box-Ljung test
##
## data:  mymodel$residuals
## X-squared = 0.12203, df = 1, p-value = 0.7268

Box.test(mymodel$residuals,lag = 2, type = "Ljung-Box")

##
##  Box-Ljung test
##
## data:  mymodel$residuals
## X-squared = 1.231, df = 2, p-value = 0.5404
```