**Dataset:** Reliance.com NSE Stock Price

**Data Description**

The data consists

**Time series plot of data**

The data is stored in the excel file Raw Data Reliance.csv

**>data= read.csv("~/Raw Data Reliance NSE.csv")**

This data is converted into a time series. It is named as ts2.

>ts2=ts(data$Closing.Stock..Price,frequency = 5)

Here, the time series is made from week 1 to 63 with 5 days in weeks 1 to 62 and 2 days in week 63.

Hence,

> ts2

Time Series:

Start = c(1, 1)

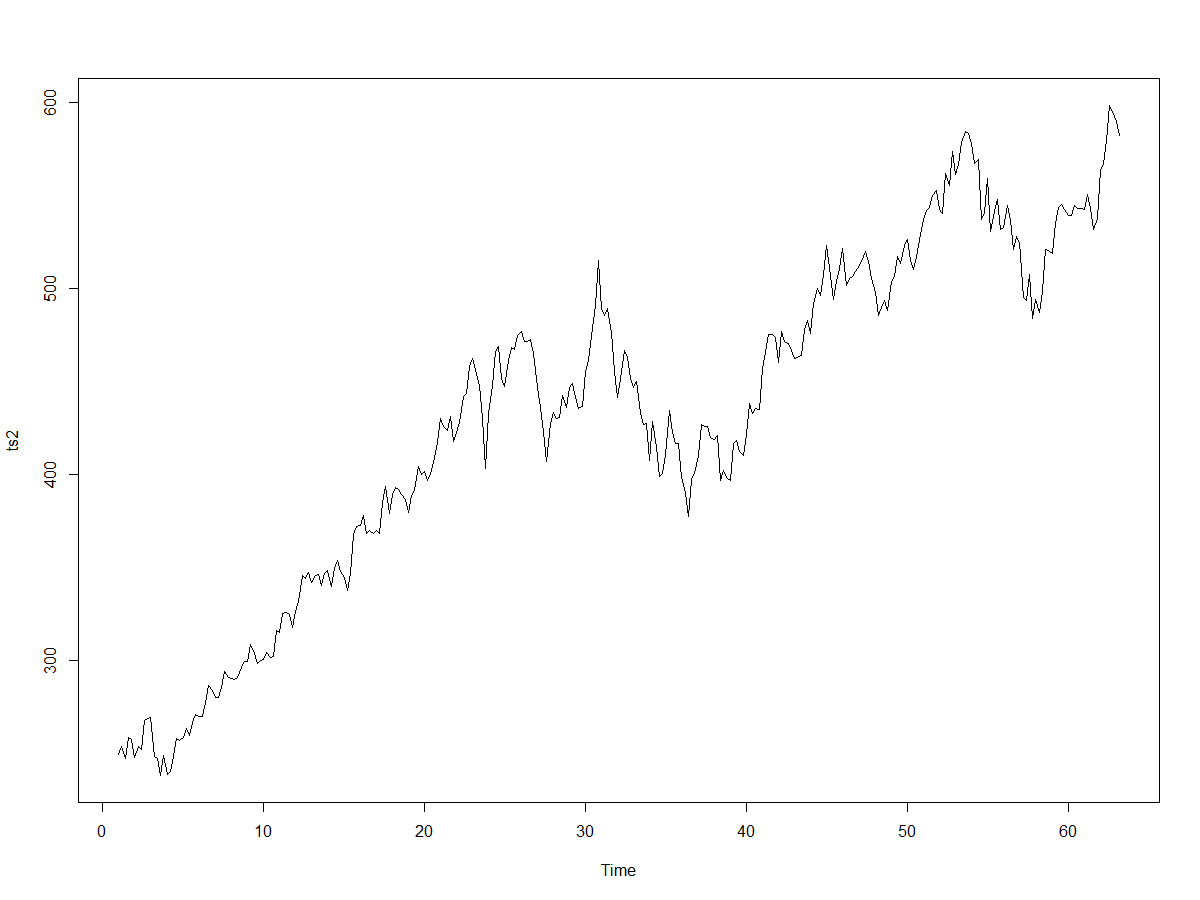
End = c(63, 2)

Frequency = 5

Since the data consists of stock prices for 5 weekdays from Monday to Friday, the frequency of time series is taken as 5.

**Plot of complete data time series**

>ts.plot(ts2)



From the time series plot, it is visible that the data has a trend(increasing with increase in time).

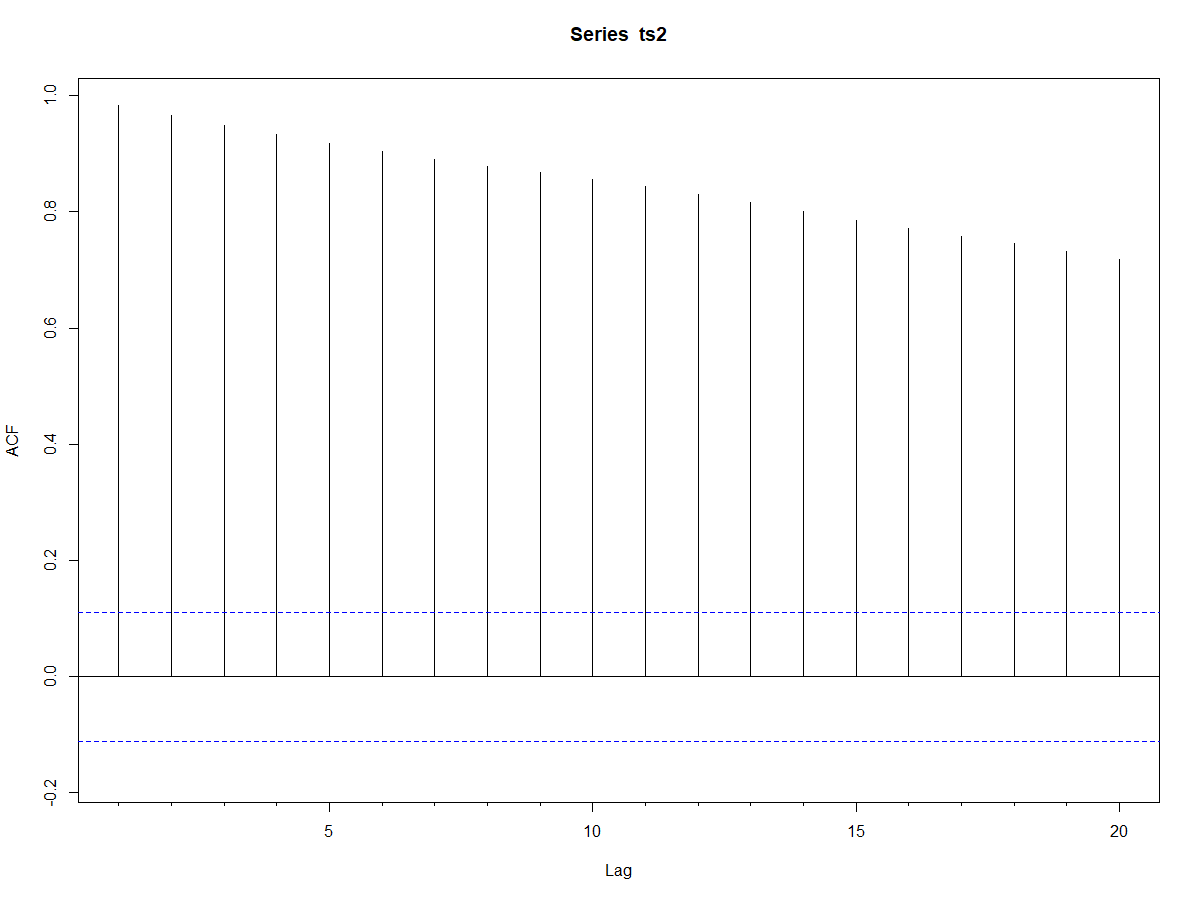
**White noise test for time series plot**

**Test for stationarity:**

**ACF Plot**

library(forecast)

Acf(ts2,lag.max = 20)



From the above ACF correlogram, it can be seen that for the first 20 lags, there is significant autocorrelation between values of Yt. Hence, the time series displays a trend.

**Ljung-Box test**

Ho: r1=r2=…=r20=0

Ha: One of them is non-zero

Box.test(ts2,lag=20,"Ljung-Box")

#Output: Box-Ljung test

data: ts2

X-squared = 4688.3, df = 20, p-value < 2.2e-16

At 5% level of significance, since p< alpha, we reject Ho. Atleast one of r1,r2…r20 is non-zero. Hence, there is autocorrelation for first 20 lags of Yt.

Hence, the series displays significant trend.

**Q.1 Create the time series, corresponding to the model period as well as**

**validation period.**

**Model data period:** 30 June 2006 to 14 August 2007

>modeldatatimeseries=ts(data$Closing.Stock..Price,start=c(1,1),end=c(57,1),frequency = 5)

>modeldatatimeseries

Time Series:

Start = c(1, 1)

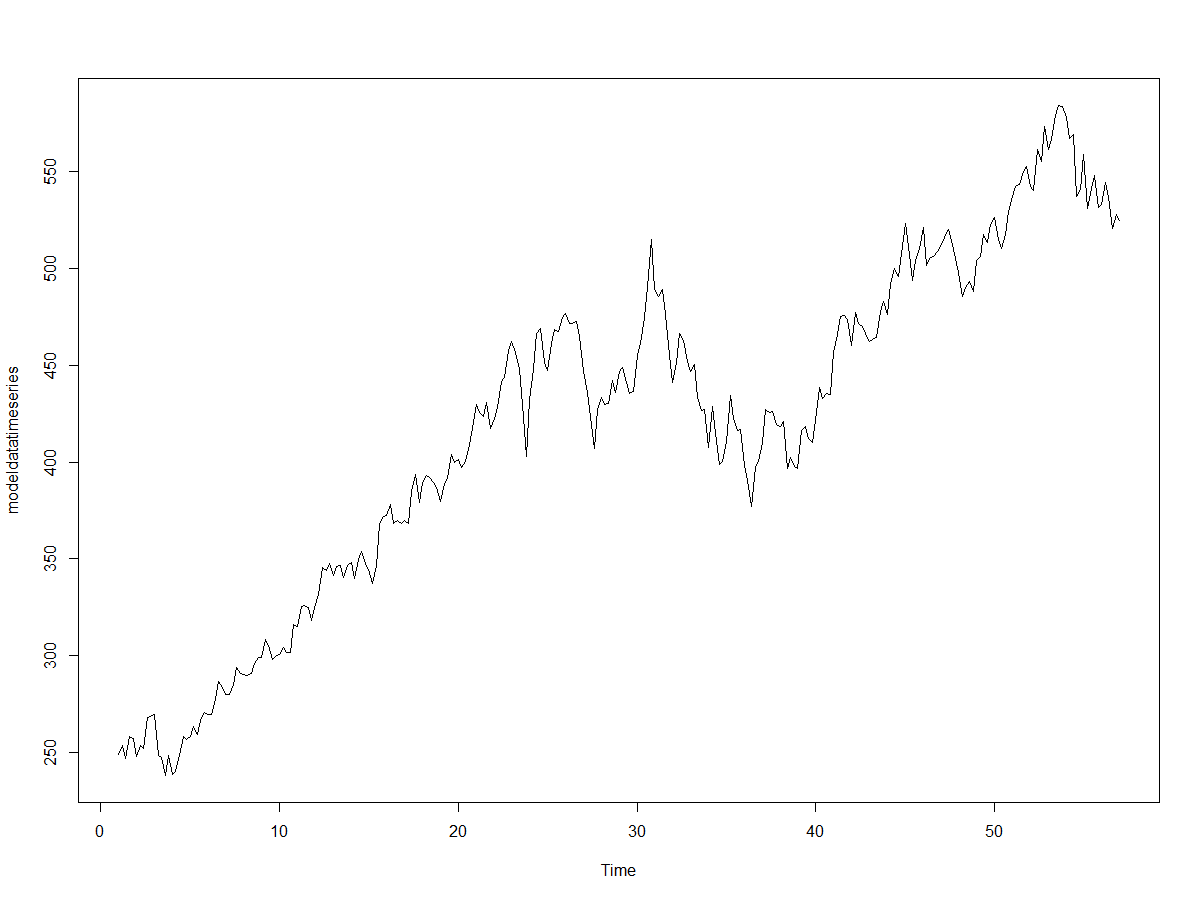
End = c(57, 1)

Frequency = 5

14 August 2007 is 281st datapoint. Hence, time series ends at 1st day of 57th week

for 5 days a week data.

**Time series plot of model data**



The time series has increasing trend and increasing variance with increase in time.

**Summary statistics**

>summary(modeldatatimeseries)

Min. 1st Qu. Median Mean 3rd Qu. Max.

238.1 348.2 428.7 418.0 482.9 584.2

|  |  |
| --- | --- |
| |  | | --- | |  | |

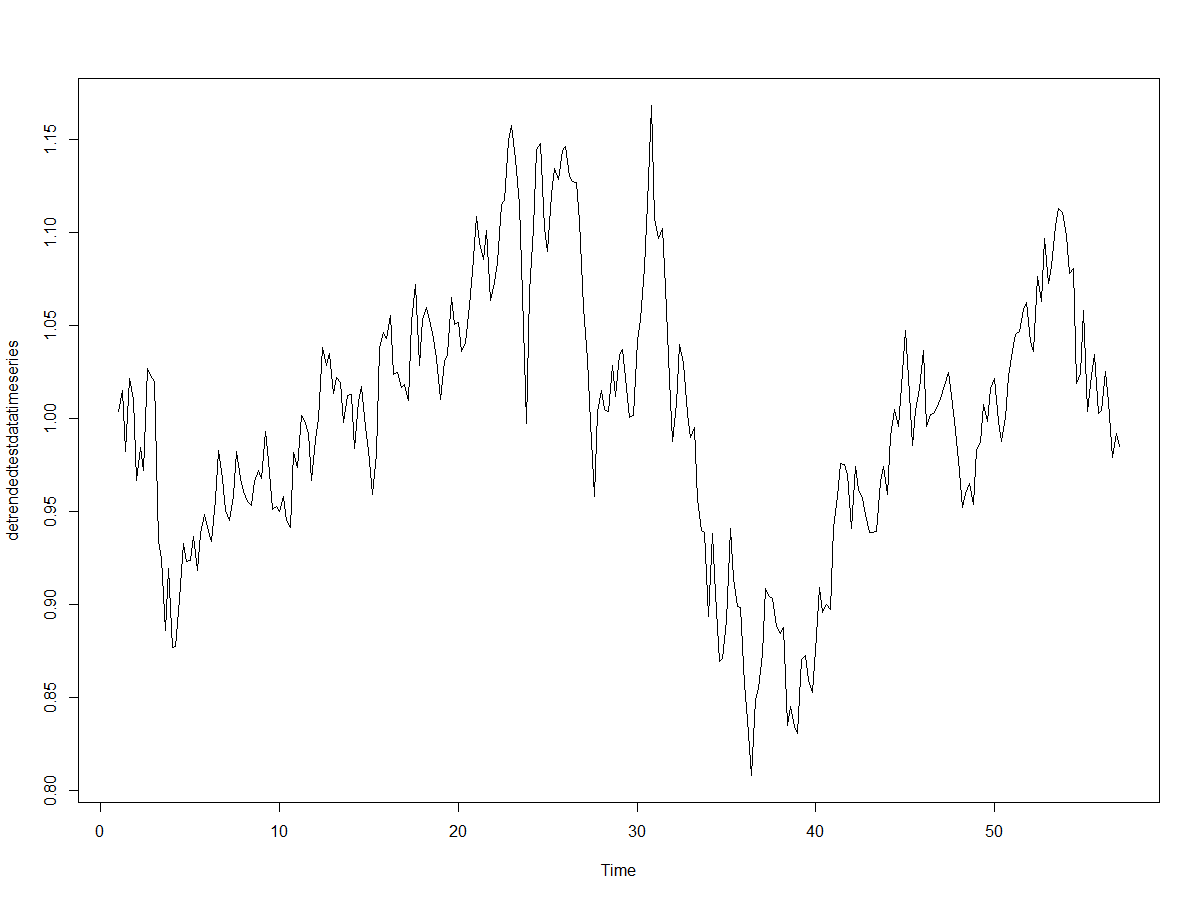
**Validation data**

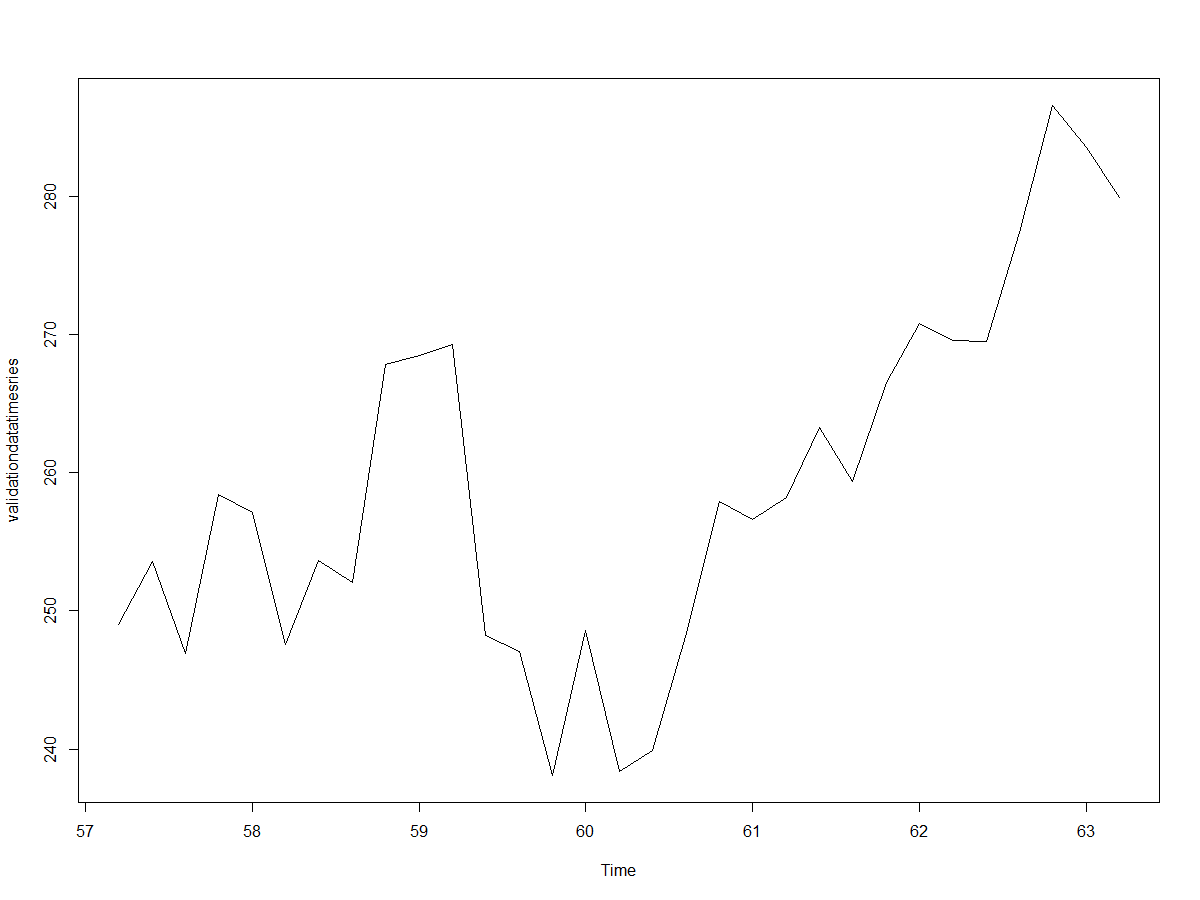
**Period:** 16 August 2007 to 27 September 2007

>validationdatatimesries=ts(data$Closing.Stock..Price,start = c(57,2),end = c(63,2),frequency = 5)

**Time series plot of validation data**

>ts.plot(validationdatatimesries)





**Summary statistics**

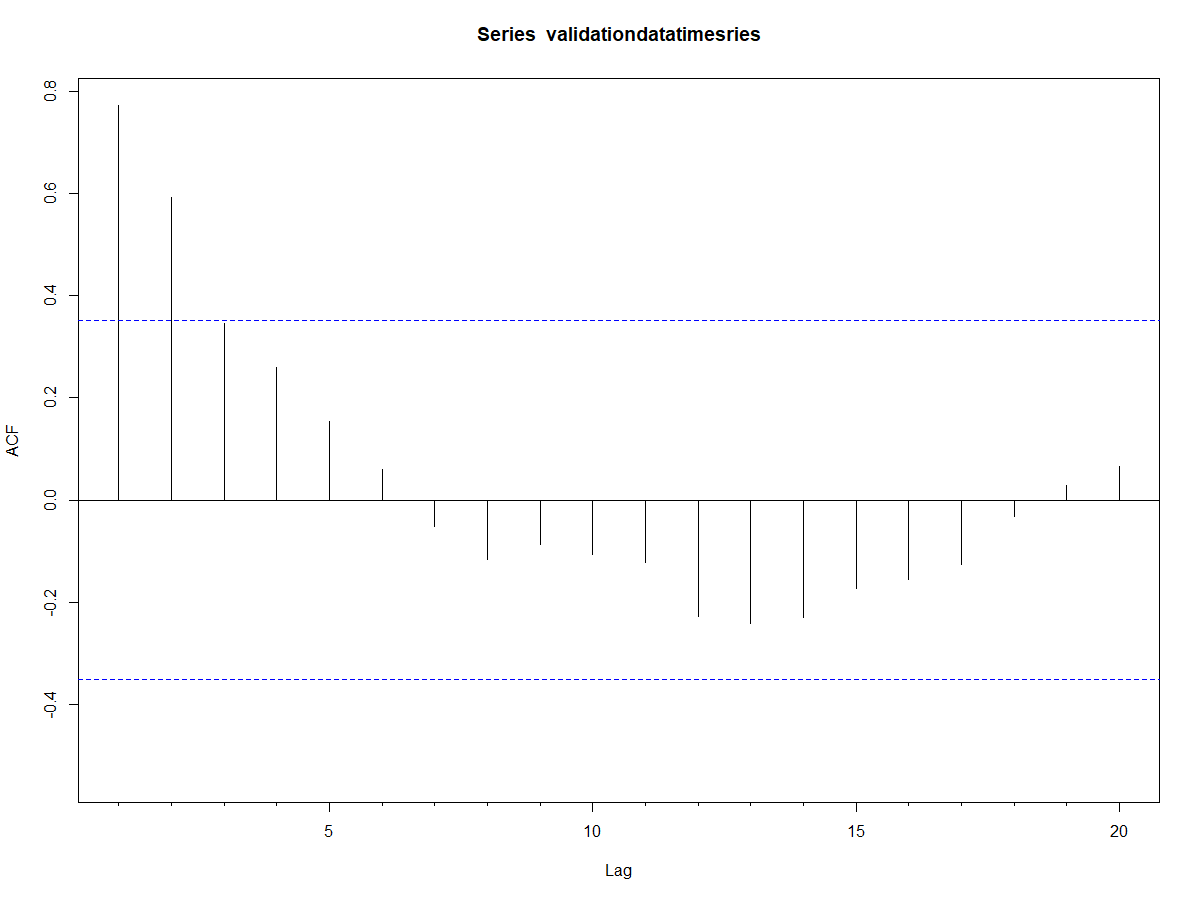
>summary(validationdatatimesries)

Min. 1st Qu. Median Mean 3rd Qu. Max.

238.1 248.4 257.9 259.1 268.9 286.6

**ACF Plot of validation data**

>Acf(validationdatatimesries,lag.max = 20)



**Ljung-Box test**

>Box.test(validationdatatimesries,lag=20,"Ljung-Box")

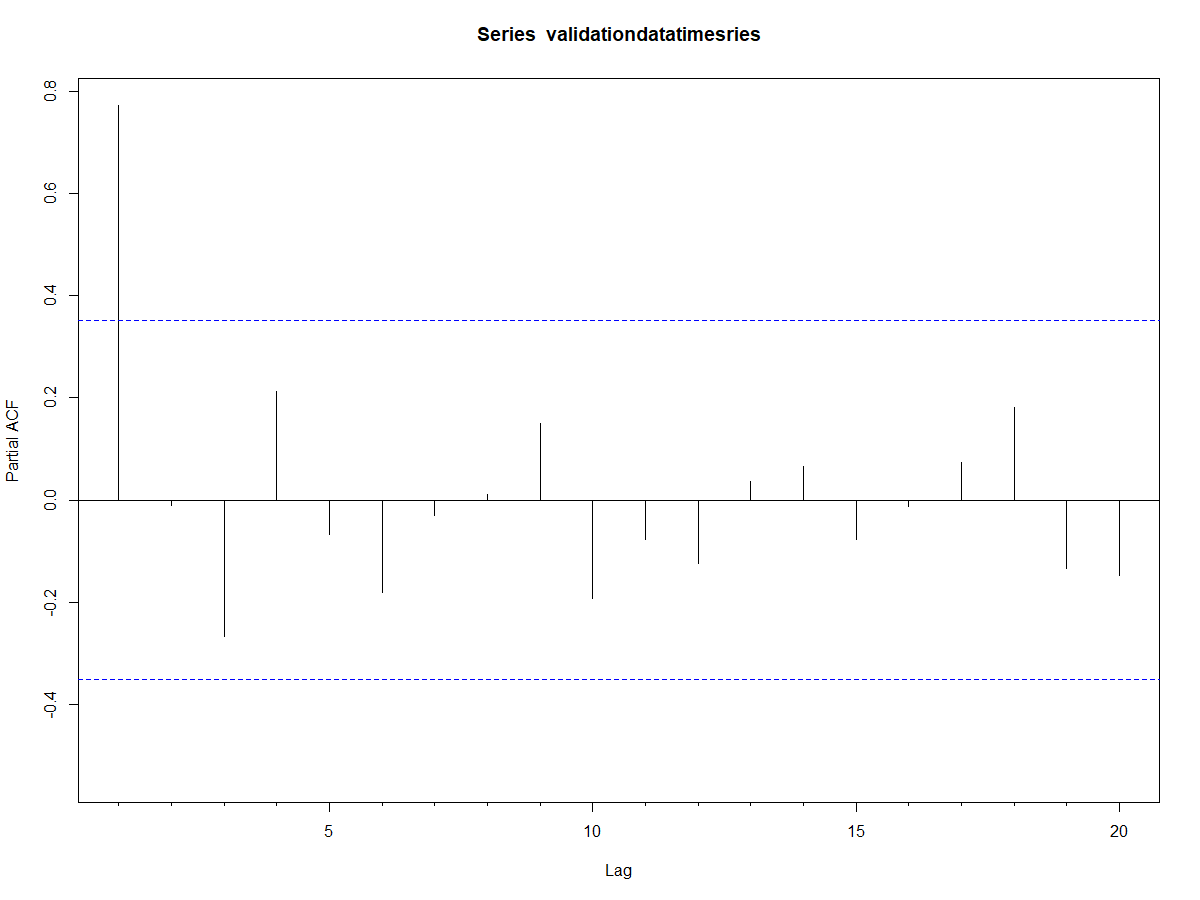
**#Output:** Box-Ljung test

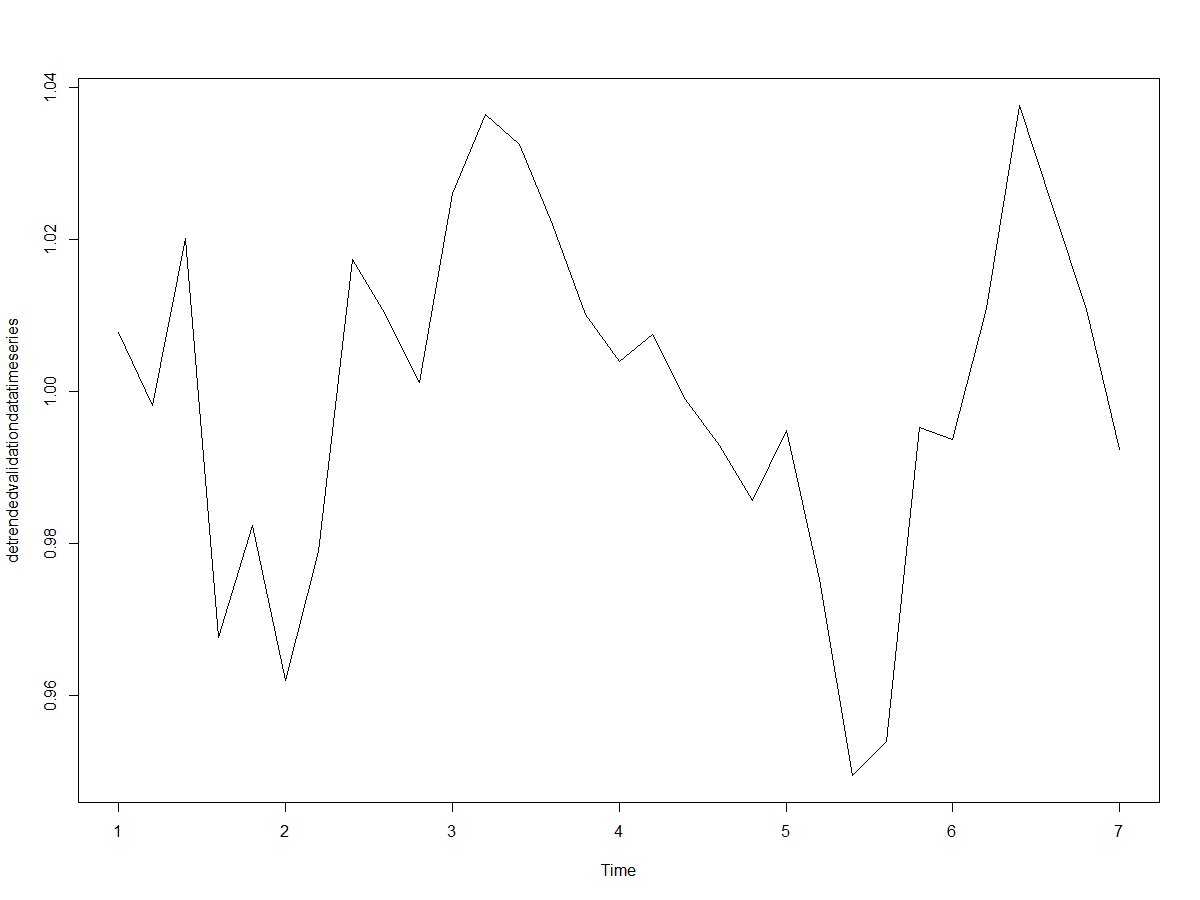
data: validationdatatimesries

X-squared = 57.653, df = 20, **p-value = 1.633e-05**

**PACF plot**

>Pacf(validationdatatimesries,lag.max = 20)





**Q.2**

**Q.3 Split the time series (entire) into two roughly equal part and compute/plot summary stats ACF/PACF of the two parts and reflect.**

**The time series has 312 points. It is split into two parts of 156 datapoints each.**

**Part 1: Closing stock prices from date 30/06/2006 to 13/02/2007**

ts\_part1=ts(data$Closing.Stock..Price,start=c(1,1),end=c(32,1),frequency = 5)

**Summary statistics**

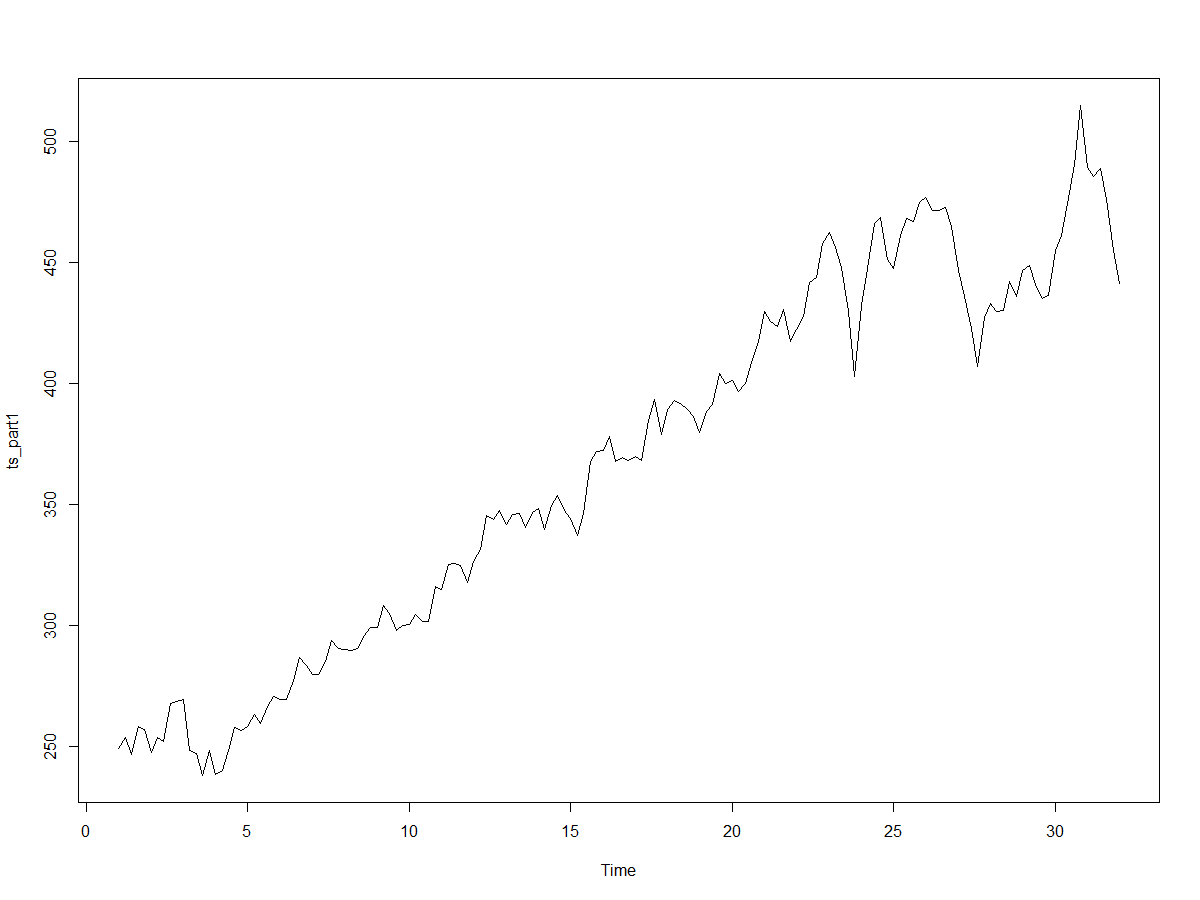
summary(ts\_part1)

Min. 1st Qu. Median Mean 3rd Qu. Max.

238.1 297.6 369.8 366.2 435.4 515.0

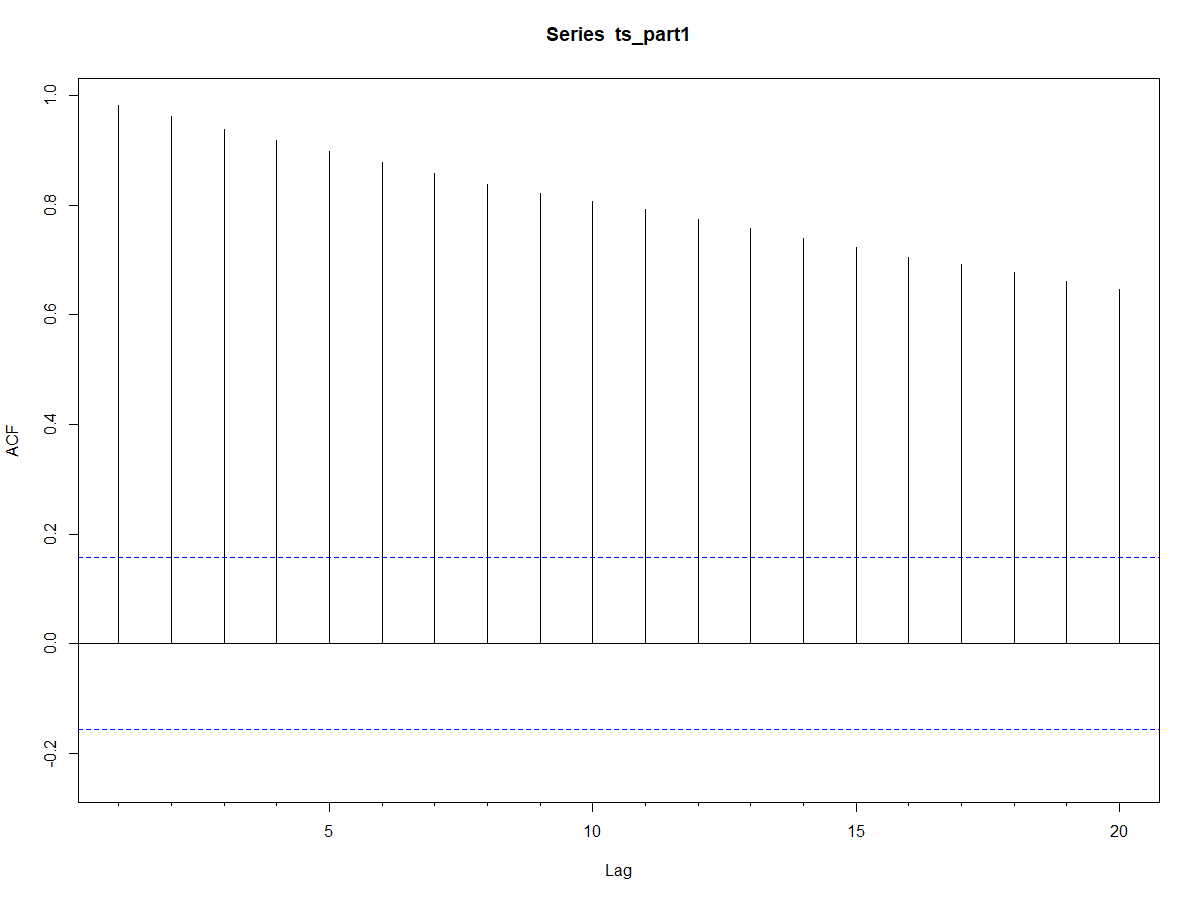
**Plot of time series part 1**

ts.plot(ts\_part1)

From the plot, it can be seen that the time series has **increasing trend** & increasing variance with increasing time. **Hence, it is multiplicative.**

**ACF plot**

Acf(ts\_part1,lag.max = 20)



**Ljung-Box test**

Box.test(ts\_part1,lag=20,"Ljung-Box")

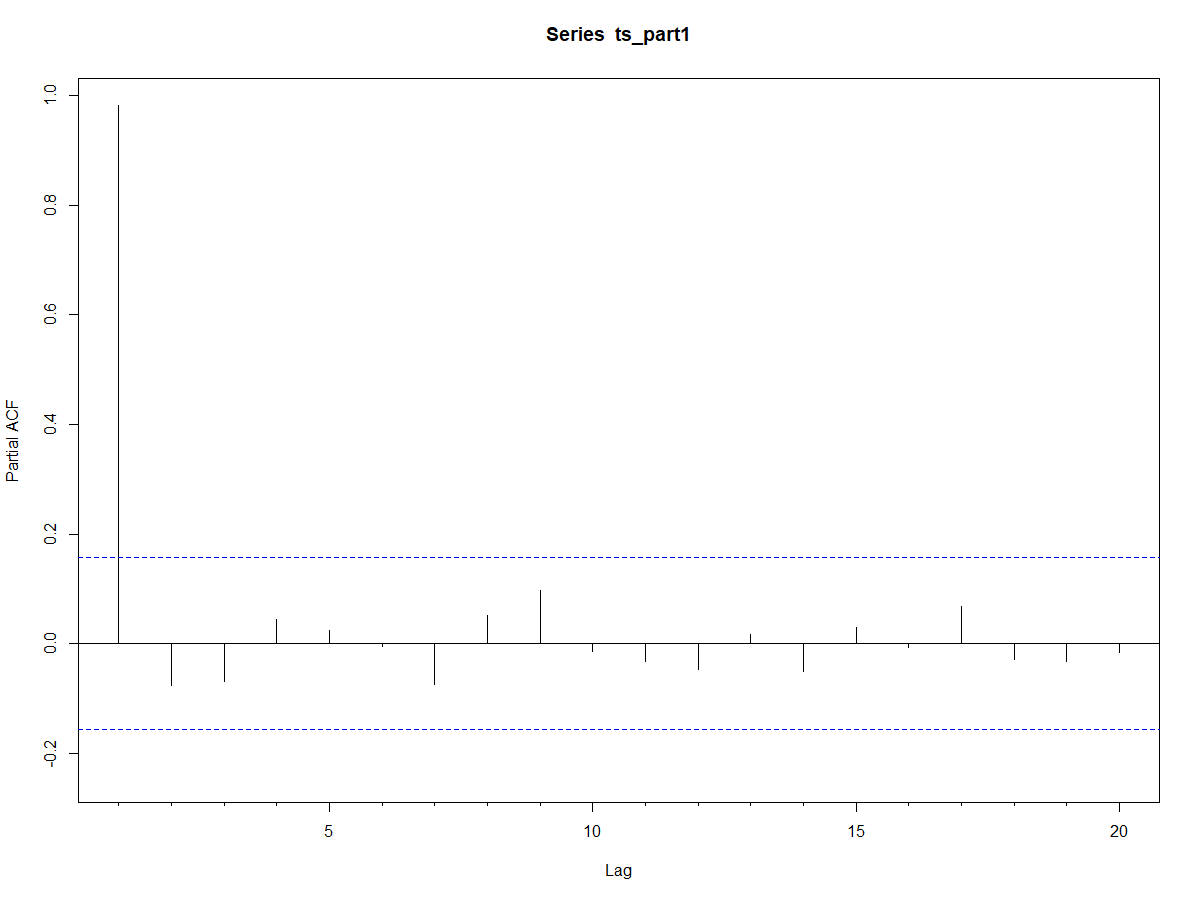
Box-Ljung test

data: ts\_part1

X-squared = 2201, df = 20, p-value < 2.2e-16

**PACF plot**

Pacf(ts\_part1,lag.max = 20)



**Part 2 : Closing stock prices from date 14/02/2007 to 27/09/2007**

|  |  |  |
| --- | --- | --- |
| ts\_part2=ts(data$Closing.Stock..Price,start=c(32,2),end=c(63,2),frequency = 5)  **Plot of time series part 2**  ts.plot(ts\_part2)    It can be visualised from the plot that the time series has an increasing trend.  **Summary statistics**  summary(ts\_part2)  Min. 1st Qu. Median Mean 3rd Qu. Max.  238.1 297.6 369.8 366.2 435.4 515.0  **ACF plot**  Acf(ts\_part2,lag.max = 20)    From the ACF correlogram, it can be seen that there is significant autocorrelation  for the first twenty lags.  **Ljung-Box test**  Box.test(ts\_part2,lag=20,"Ljung-Box")  #Output: Box-Ljung test  data: ts\_part2  X-squared = 2201, df = 20, p-value < 2.2e-16  Hence, at 5% level of significance, there is significant autocorrelation of Yt for first twenty lags.  So, the data is non-stationary.  **PACF plot**    **Q.4. Now implement a variation of the decomposition method on the original data as**  **well as on the Box-Cox transformed data (and reverse transformation on the fitted/**  **forecasted values).**  **1. Model data – Original model**  **ACF plot**  >Acf(modeldatatimeseries,lag.max = 20)    **Ljung-Box test**  >Box.test(modeldatatimeseries,lag=20,"Ljung-Box")  #Output: Box-Ljung test  data: modeldatatimeseries  X-squared = 4232.5, df = 20, p-value < 2.2e-16  **PACF plot**  Pacf(modeldatatimeseries,lag.max = 20)    **Trend estimation**  **1.Smoothing the series using MA**  Since the data is 5-day weekly data, we use MA-5 to smoothen the data.  >MA5\_modeldata= ma(test\_data,5)  Now, we regress the MA data with the corresponding time stamps of 3,8,11 & further  To create the time stamp:  k3=seq(from=3,to=281,by=5)  Selecting the values in the smoothened model data corresponding to these time  stamps:  >MA5\_modeldata[k3]  **Regression for Quadratic trend:**  >k3sq=k3^2  >modeldata\_MA5.qm = lm(MAmatrix[,2]~k3 + k3sq)  >summary(modeldata\_MA5.qm)  Call:  lm(formula = MAmatrix[, 2] ~ k3 + k3sq)  Residuals:  Min 1Q Median 3Q Max  -73.940 -16.500 2.239 15.396 53.291  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 2.467e+02 1.239e+01 19.905 < 2e-16 \*\*\*  k3 1.609e+00 2.037e-01 7.901 1.61e-10 \*\*\*  k3sq -2.100e-03 7.021e-04 -2.991 0.00421 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 30.68 on 53 degrees of freedom  **Multiple R-squared: 0.8862**, Adjusted R-squared: 0.8819  F-statistic: 206.4 on 2 and 53 DF, **p-value: < 2.2e-16**  **Fitted values for the entire model data duration for the quadratic trend**  >coeffs\_modeldata\_MA5.qm=coefficients(modeldata\_MA5.qm)  >modeldatadays=1:281  >modeldataquadratictrendfittedvalues=coeffs\_modeldata\_MA5.qm[1]+(coeffs\_modeldata\_MA5.qm[2]\*modeldatadays)+(coeffs\_modeldata\_MA5.qm[3]\*(modeldatadays^2))  >modeldataquadratictrendfittedvalues  **De-trended model data**  > detrendedmodeldata=modeldata/modeldataquadratictrendfittedvalues  **Time series of detrended model data**  > detrendedmodeldatatimeseries=ts(detrendedmodeldata,frequency = 5)  **Time series plot of detrended test data**  > ts.plot(detrendedmodeldatatimeseries)    **ACF plot of detrended test data**  > Acf(detrendedmodeldata,lag.max = 17) (Since model data has 281 datapoints, max lag has been taken upto = sqrt(281) = approx.17)    From the above correlogram, it can be seen that even upto 17 lags, there is  significant autocorrelation for detrended test data. This shows that after detrending,  significant seasonality exists.  **Method of estimation of seasonality index: Ratio to trend method**  Trend = testdata.qm$fitted.values  detrendedtestdata=test\_data/ testdata.qm$fitted.values  In ratio to trend method, we calculate the average value of detrended data for each  day to determine its seasonality.  For example, for **Friday** detrended data, Seasonality is estimated by  >k1=seq(from=1,to=281,by=5)  >FridayClosingPrices=detrendedmodeldata[k1]  >mean(FridayClosingPrices)  [1] 0.9972005  **Monday**  >k2=seq(from=2,to=281,by=5)  >MondayClosingPrices=detrendedmodeldata[k2]  >mean(MondayClosingPrices)  [1] 0.9996593  **Tuesday**  >k3=seq(from=3,to=281,by=5)  >TuesdayClosingPrices=detrendedmodeldata[k3]  >mean(TuesdayClosingPrices)  [1] 0.9996737  **Wednesday**  >k4=seq(from=4,to=281,by=5)  >WednesdayClosingPrices=detrendedmodeldata[k4]  >mean(WednesdayClosingPrices)  [1] 1.001133  **Thursday**  >k5=seq(from=5,to=281,by=5)  >ThursdayClosingPrices=detrendedmodeldata[k5]  >mean(ThursdayClosingPrices)  [1] 0.9988785  >Seasonalityindex=c(mean(FridayClosingPrices),mean(MondayClosingPrices),mean(TuesdayClosingPrices),mean(WednesdayClosingPrices),mean(ThursdayClosingPrices))  >Seasonalityindex  [1] 0.9972005 0.9996593 0.9996737 1.0011329 0.9988785  The mean of seasonalityindex for multiplicative model should be 1  >mean(Seasonalityindex)  [1] 0.999309  > SI\_modeldata=Seasonalityindex/ mean(Seasonalityindex)   |  | | --- | | >SI\_modeldata  [1] 0.9978901 1.0003506 1.0003650 1.0018252 0.9995692  >mean(SI\_modeldata)  [1] 1 | | Seasonality index for all 312 data points  >seasonality=c(rep(SI\_modeldata,times=62),SI\_modeldata[1:2])  **Forecasting for model data**  >modeldata\_forecastedval=modeldataquadratictrendfittedvalues\*seasonality[1:281]  **Fitting of quadratic trend for validation data points**  >validationdatadays=282:312  > validationdataquadraticfittedvalues=coeffs\_modeldata\_MA5.qm[1]+  (coeffs\_modeldata\_MA5.qm[2]\*validationdatadays)+(coeffs\_modeldata\_MA5.qm[3]  \*(validationdatadays^2))  **2. Validation data – Original model**  **Forecasting of values for validation data**  >validationdata\_forecastedval=validationdataquadraticfittedvalues\*  seasonality[282:312] |   **3. Box-cox transformed data**  >boxcoxdata=BoxCox(data$Closing.Stock..Price,2)  >boxcoxdatats=ts(boxcoxdata,frequency = 5)  Dividing the box-cox transformed data into 2 parts: model data and validation data.  >modeldata\_boxcox=boxcoxdatats[1:281]  >validationdata\_boxcox=boxcoxdatats[282:312]  **3.1 Model data: Box-cox transformed model**  Making time series of model data  >modeldata\_boxcoxts=ts(modeldata\_boxcox,frequency =5)  **Time Series Plot of transformed model data**  >ts.plot(modeldata\_boxcoxts)    From the above plot, it can be seen that the data has **increasing trend** and variance  is increasing with increase in time. Hence, **multiplicative model** is suitable for the  data.  **Test of stationarity for model data**  **ACF Plot**  >Acf(modeldata\_boxcoxts)    **Ljung-Box test**  >Box.test(modeldata\_boxcoxts,lag = 10,"Ljung-Box")  **#Output:** Box-Ljung test  data: modeldata\_boxcoxts  X-squared = 2439.4, df = 10, **p-value < 2.2e-16**  Reject Ho at 5% level of significance.  **PACF test**  >Pacf(modeldata\_boxcoxts)    **Fitting quadratic trend in box-cox transformed model data**  **1.Smoothing the boxcox transformed series using MA**  >MA5\_modeldata\_boxcox= ma(modeldata\_boxcox,5)  Now, we regress the MA data with the corresponding time stamps of 3,8,11 & further  k3=seq(from=3,to=281,by=5)  Selecting the values in the smoothened model data corresponding to these time  stamps:  >MA5\_modeldata\_boxcox[k3]  **Regression for Quadratic trend:**  >k3sq=k3^2  >MAmatrix\_boxcox=cbind(k3matrix,matrix(MA5\_modeldata\_boxcox[k3],56,1))  >modeldata\_MA5\_boxcox.qm = lm(MAmatrix\_boxcox[,2]~k3 + k3sq)  >summary(modeldata\_MA5\_boxcox.qm)  Call:  lm(formula = MAmatrix\_boxcox[, 2] ~ k3 + k3sq)  Residuals:  Min 1Q Median 3Q Max  -30586.2 -6019.3 163.1 5080.4 25134.0  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 30335.4539 5341.8228 5.679 5.84e-07 \*\*\*  k3 470.3141 87.7986 5.357 1.87e-06 \*\*\*  k3sq -0.1997 0.3026 -0.660 0.512  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 13220 on 53 degrees of freedom  Multiple R-squared: 0.8715, Adjusted R-squared: 0.8666  F-statistic: 179.7 on 2 and 53 DF, p-value: < 2.2e-16  Since the coefficient of k3 square term is insignificant, it is dropped from the  equation.  >modeldata\_MA5\_boxcox.lm=lm(MAmatrix\_boxcox[,2]~k3)  >summary(modeldata\_MA5\_boxcox.lm)  Call:  lm(formula = MAmatrix\_boxcox[, 2] ~ k3)  Residuals:  Min 1Q Median 3Q Max  -29562.7 -6498.2 -632.6 5909.1 25210.1  **Coefficients:**  Estimate Std. Error t value Pr(>|t|)  (Intercept) 32973.32 3525.14 9.354 6.89e-13 \*\*\*  k3 414.20 21.75 19.045 < 2e-16 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 13150 on 54 degrees of freedom  Multiple R-squared: 0.8704, Adjusted R-squared: 0.868  F-statistic: 362.7 on 1 and 54 DF, p-value: < 2.2e-16  **Quadratic trend fitted values for model data**  >coeffs\_modeldata\_boxcox=coefficients(modeldata\_MA5\_boxcox.lm)  >coeffs\_modeldata\_boxcox  (Intercept) k3  32973.3225 414.1977  >boxcoxmodeldatatrendfitval=coeffs\_modeldata\_boxcox[1]+(coeffs\_modeldata\_boxcox[2]\*  modeldatadays)  **De-trended box-cox transformed model data**  >detrendedmodeldata\_boxcox=modeldata\_boxcox/boxcoxmodeldatatrendfitval  **Time series of detrended model data**  >detrendedmodeldata\_boxcoxts=ts(detrendedmodeldata\_boxcox,frequency = 5)  **Time series plot of detrended test data**  ts.plot(detrendedmodeldata\_boxcoxts)  **Time series plot of detrended model data**    **ACF plot of detrended box-cox transformed model data**  >Acf(detrendedmodeldata\_boxcox,lag.max=17)    **Ljung-Box test**  > Box.test(detrendedmodeldata\_boxcox,lag=17,"Ljung-Box")  Box-Ljung test  data: detrendedmodeldata\_boxcox  X-squared = 2391.1, df = 17, **p-value < 2.2e-16**  Hence, the data is non-stationary even after detrending.  **Seasonality Index**  **1. Friday**    >Fri\_boxcox=detrendedmodeldata\_boxcox[k1]  >FriSI=mean(Fri\_boxcox)  > FriSI  [1] 0.9891189  **2. Monday**    >Mon\_boxcox=detrendedmodeldata\_boxcox[k2]  >MonSI=mean(Mon\_boxcox)  >MonSI  [1] 0.9943059  **3. Tuesday**    >Tue\_boxcox=detrendedmodeldata\_boxcox[k3]  >TueSI=mean(Tue\_boxcox)  >TueSI  [1] 0.9953985  **4.Wednesday**    >Wed\_boxcox=detrendedmodeldata\_boxcox[k4]  >WedSI=mean( Wed\_boxcox)  >WedSI  [1] 0.997997  **5. Thursday**  >Thu\_boxcox=detrendedmodeldata\_boxcox[k5]  >ThuSI=mean( Thu\_boxcox)  >ThuSI  [1] 0.9933987  >SI=c(FriSI,MonSI,TueSI,WedSI,ThuSI)  > mean(SI)  [1] 0.9940438  For multiplicative model, the mean of SI should be equal to 1.  Hence, adjust SI so that mean=1.  >SI\_boxcox=SI/mean(SI)  >SI\_boxcox  [1] 0.9950456 1.0002637 1.0013628 1.0039769 0.9993510  SI for all 312 days is given by  >SI\_boxcox\_total=c(rep(SI\_boxcox,times=62),SI\_boxcox[1:2])  **Forecasting for model data**  >modeldata\_boxcox\_forecast= boxcoxmodeldatatrendfitval\*SI\_boxcox\_total[1:281]  **Reverse transformation for model data forecasted values**  >modeldataforecast\_invboxcox=InvBoxCox(modeldata\_boxcox\_forecast,2)  >modeldataforecast\_invboxcox  **3.2** **Validation data: Box- Cox transformed model**  Time series of validation data  >validationdata\_boxcoxts=ts(validationdata\_boxcox,frequency = 5)  Time series plot of transformed validation data  >ts.plot(validationdata\_boxcoxts)    The above time series plot shows **increasing trend**.  **ACF plot**  >Acf(validationdata\_boxcoxts,lag.max = 20)    **Ljung-Box test**  >Box.test(validationdata\_boxcoxts,lag=20,"Ljung-Box")  **#Output:** Box-Ljung test  data: validationdata\_boxcoxts  X-squared = 82.141, df = 20, **p-value = 1.695e-09**  **Reject Ho at 5% level of significance.**  **PACF plot**  >Pacf(validationdata\_boxcoxts,lag.max = 20)    **Fitting the trend values for validation days**  >validationdatadays  >coeffs\_modeldata\_boxcox  > boxcoxvalidationdatatrendfitval=coeffs\_modeldata\_boxcox[1]+(coeffs\_modeldata\_boxcox  + [2]\* validationdatadays)  **Forecast for validation datapoints**  >validationdataboxcoxforecast=boxcoxvalidationdatatrendfitval\*SI\_boxcox\_total[282:312]  **Inverse transformation of forecast for validation data**    >validationdataforecast\_invboxcox=InvBoxCox(validationdataboxcoxforecast,2)  >validationdataforecast\_invboxcox  [1] 547.3885 548.4459 549.9180 549.4036 548.9701 551.1599 552.2142 553.6861 553.1579  552.7112 554.9057 555.9570 557.4287 556.8868 556.4271 558.6263 559.6747  [18] 561.1463 560.5909 560.1184 562.3224 563.3679 564.8395 564.2708 563.7855  565.9943 567.0371 568.5087 567.9267 567.4289 569.6425  **Forecast error for validation data**  >validationdataboxcoxforecasterror=validation\_data-validationdataforecast\_invboxcox  **Q.5. Compute MAPE/RMSE/MAE of the two methods (original vis-a-vis reversed BC**  **transformed) in the model period as well as validation (hold-out period).**  An error function Acc\_3( ) is created for estimating RMSE, MAE and MAPE values.  Acc\_3 <- function(Y, Yhat) {  error <- Y - Yhat  err <- error[!is.na(error)] #remove NA  Y <- Y[!is.na(error)]  RMSE <- round(sqrt(mean(err^2)), 2)  MAD <- round(mean(abs(err)), 2)  MAPE <- round(mean(abs(err/Y)) \* 100, 3)  ErrVec <- cbind(RMSE, MAD, paste(toString(MAPE),"%"))  colnames(ErrVec) <- c("RMSE","MAD","MAPE")  return(ErrVec)  }  **Error terms for model data for original model**  > Acc\_3(modeldata,modeldata\_forecastedval)  RMSE MAD MAPE  [1,] "31.11" "23.57" "5.567 %"  **Error terms for validation data for original model**  >validation\_data=data$Closing.Stock..Price[282:312]  >Acc\_3(validation\_data,validationdata\_forecastedval)  RMSE MAD MAPE  [1,] "28.14" "21.39" "4.006 %"  **Error terms for model data for box-cox transformed model**  >Acc\_3(modeldata,modeldataforecast\_invboxcox)  RMSE MAD MAPE  [1,] "31.04" "24.38" "6.037 %"  **Error terms for validation data for box-cox transformed model**  >Acc\_3(validation\_data,validationdataforecast\_invboxcox)  RMSE MAD MAPE  [1,] "32.41" "27.3" "5.242 %"  **Q.6. Perform the white noise test on residuals (fitted values) on either method**  **1. Residuals for model data for original model**  **Error values for forecast of validation data**  >modeldataforecasterror=modeldata-modeldata\_forecastedval  **ACF plot of model data forecast error**  >Acf(modeldataforecasterror,lag.max=17)    **Ljung-Box test**  >Box.test(modeldataforecasterror,lag = 17,"Ljung-Box")  Box-Ljung test  data: modeldataforecasterror  X-squared = 2341.2, df = 17, **p-value < 2.2e-16**  p<alpha at 5% level of significance, **Reject Ho**  There is significant autocorrelation among error terms. **Fails white noise test.**  **2. Residuals for validation data for original model**  **Error values for forecast of validation data**  >validationforecasterror=validation\_data-validationdata\_forecastedval  **ACF plot of validation data forecast error**  >Acf(validationforecasterror)    **White noise test for validation data forecast error**  >Box.test(validationforecasterror,lag=10,"Ljung-Box")  Box-Ljung test  data: validationforecasterror  X-squared = 67.007, df = 10, **p-value = 1.671e-10**  p<alpha at 5% level of significance, reject Ho.  There is significant autocorrelation among error terms. Fails white noise test.  **3. Residuals for model data for box-cox transformed model**  **Error values for box cox model data forecast**  >Boxcoxmodeldataforecasterror= modeldata-modeldataforecast\_invboxcox  **Time series of forecast error**  >Boxcoxmodeldataforecasterrorts=ts(Boxcoxmodeldataforecasterror,frequency=5)  **Time series plot of forecast error**    **Acf plot of box cox model data forecast error**  >Acf(Boxcoxmodeldataforecasterror,lag.max=17)    **Ljung-Box test**  >Box.test(Boxcoxmodeldataforecasterror,lag = 17,"Ljung-Box")  Box-Ljung test  data: Boxcoxmodeldataforecasterror  X-squared = 2310, df = 17, **p-value < 2.2e-16**  p<alpha at 5% level of significance, reject Ho.  There is significant autocorrelation among error terms. **Fails white noise test.**  **4. Residuals for validation data for box-cox transformed model**  **Forecast error for validation data**  >validationdataboxcoxforecasterror=validation\_data-validationdataforecast\_invboxcox  **Time series of validation data forecasted errors**  >validationdataboxcoxforecasterrorts=ts(validationdataboxcoxforecasterror,frequency=5)  **Time series plot of validation data forecasted errors**  >ts.plot(validationdataboxcoxforecasterrorts)    **ACF plot of error terms for validation data forecast**  >Acf(validationdataboxcoxforecasterror,lag.max=10)    **Ljung-Box test**  Box.test(validationdataboxcoxforecasterror,lag=10,"Ljung-Box")  Box-Ljung test  data: validationdataboxcoxforecasterror  X-squared = 61.308, df = 10, **p-value = 2.048e-09**  p<alpha at 5% level of significance, **Reject Ho.**  There is significant autocorrelation among error terms. **Fails white noise test.**  **Revised code of 28 July**  library(forecast)  data= read.csv("~/Raw Data Reliance NSE.csv")  ts2=ts(data$Closing.Stock..Price,frequency = 5)  ts2  ts.plot(ts2)  Acf(ts2,lag.max = 20)  Box.test(ts2,lag=20,"Ljung-Box")  modeldata=data$Closing.Stock..Price[1:281]  modeldatatimeseries=ts(data$Closing.Stock..Price,start=c(1,1),end=c(57,1),frequency = 5)  summary(modeldatatimeseries)  Acf(modeldatatimeseries,lag.max = 20)  Box.test(modeldatatimeseries,lag=20,"Ljung-Box")  Pacf(modeldatatimeseries,lag.max = 20)  test\_data=data$Closing.Stock..Price[1:281]  MA5\_modeldata= ma(test\_data,5)  k3=seq(from=3,to=281,by=5)  MA5\_modeldata[k3]  k3sq=k3^2  modeldata\_MA5.qm = lm(MAmatrix[,2]~k3 + k3sq)  summary(modeldata\_MA5.qm)  coeffs\_modeldata\_MA5.qm=coefficients(modeldata\_MA5.qm)  modeldatadays=1:281  modeldataquadratictrendfittedvalues=coeffs\_modeldata\_MA5.qm[1]+(coeffs\_modeldata\_MA5.qm[2]\*modeldatadays)+(coeffs\_modeldata\_MA5.qm[3]\*(modeldatadays^2))  modeldataquadratictrendfittedvalues  detrendedmodeldata=modeldata/modeldataquadratictrendfittedvalues  detrendedmodeldatatimeseries=ts(detrendedmodeldata,frequency = 5)  ts.plot(detrendedmodeldatatimeseries)  Acf(detrendedmodeldata,lag.max = 17)  Trend = modeldataquadratictrendfittedvalues  k1=seq(from=1,to=281,by=5)  FridayClosingPrices=detrendedmodeldata[k1]  mean(FridayClosingPrices)  k2=seq(from=2,to=281,by=5)  MondayClosingPrices=detrendedmodeldata[k2]  mean(MondayClosingPrices)  k3=seq(from=3,to=281,by=5)  TuesdayClosingPrices=detrendedmodeldata[k3]  mean(TuesdayClosingPrices)  k4=seq(from=4,to=281,by=5)  WednesdayClosingPrices=detrendedmodeldata[k4]  mean(WednesdayClosingPrices)  k5=seq(from=5,to=281,by=5)  ThursdayClosingPrices=detrendedmodeldata[k5]  mean(ThursdayClosingPrices)  Seasonalityindex=c(mean(FridayClosingPrices),mean(MondayClosingPrices),mean(TuesdayClosingPrices),mean(WednesdayClosingPrices),mean(ThursdayClosingPrices))  Seasonalityindex  mean(Seasonalityindex)  SI\_modeldata=Seasonalityindex/ mean(Seasonalityindex)  SI\_modeldata  mean(SI\_modeldata)  SI\_modeldata\_all=c(rep(SI\_modeldata,times=56),SI\_modeldata[1])  modeldata\_forecastedval=modeldataquadratictrendfittedvalues\*SI\_modeldata\_all  modeldata\_forecastedval  modeldataforecasterror=modeldata-modeldata\_forecastedval  Acf(modeldataforecasterror,lag.max=17)  Box.test(modeldataforecasterror,lag = 17,"Ljung-Box")  **Error codes given by sir**  **1. MAE <- function(true, est){**  **error <- true - est**  **return(mean(abs(error), na.rm=TRUE))**  **}**  **MSE <- function(true, est){**  **error <- true - est**  **return(mean(error^2, na.rm=TRUE))**  **}**  **MAPE <- function(true, est){**  **error <- true - est**  **pe <- error \* 100 / true**  **return(mean(abs(pe), na.rm=TRUE))**  **}**  **RMSE <- function(true, est){**  **error <- true - est**  **MSE <- mean(error^2, na.rm=TRUE)**  **return(MSE^0.5)**  **}**  **2. Acc <- function(Yhat,Y) {**  **error <- Y - Yhat**  **err <- error[!is.na(error)] #remove NA**  **Y <- Y[!is.na(error)]**  **MSE <- round(mean(err^2), 2)**  **MAD <- round(mean(abs(err)), 2)**  **MAPE <- round(mean(abs(err/Y)) \* 100, 3)**  **ErrVec <- cbind(MSE, MAD, paste(toString(MAPE),"%"))**  **colnames(ErrVec) <- c("MSE","MAD","MAPE")**  **return(ErrVec)**  **}**  **Acc(air\_fit,air)**  **error <- air - air\_fit**  **plot(error)**  3. Acc <- function(Yhat,Y) {  error <- Y - Yhat  err <- error[!is.na(error)] #remove NA  Y <- Y[!is.na(error)]  MSE <- round(mean(err^2), 2)  MAD <- round(mean(abs(err)), 2)  MAPE <- round(mean(abs(err/Y)) \* 100, 3)  ErrVec <- cbind(MSE, MAD, paste(toString(MAPE),"%"))  colnames(ErrVec) <- c("MSE","MAD","MAPE")  return(ErrVec)  } |
|  |
| |  | | --- | |  | |

p<alpha at 5% level of significance, reject Ho.

There is significant autocorrelation among error terms. Fails white noise test.