**Forecastiong Exam 2019 Exercise # 3**

For this exercise I have chosen seasonal time series data of U.S Census Bureau Monthly Wholesale Sale of Beer, Wine and Distilled Alcohlic Beverages from January 2000 to December 2018. Below is the link for data used.

<https://www.census.gov/econ/currentdata/dbsearch?program=MWTS&startYear=2000&endYear=2018&categories%5B%5D=4248&dataType=SM&geoLevel=US&notAdjusted=1&submit=GET+DATA&releaseScheduleId=>

First of all, I will explore the data set in R.

**Data Reading and Exploration**

We split the data into train and test set for training I used data till December 2015 and for testing from Jan 2016 till December 2018.

#Set Working Directory

setwd("C:/Users/mmajid1/Desktop/Forecasting")

data\_Beverages\_Sale<-read\_excel("SeriesReport-201904241559.xls")

Beverages\_Sale<- ts(data\_Beverages\_Sale[,2], frequency = 12, start = c(2000,1))

# Split the data in training and test set

Beverages\_Sale1 <- window(Beverages\_Sale, end=c(2015,12))

Beverages\_Sale2 <- window(Beverages\_Sale, start=c(2016,1))

# Retrieve the length of the test set

h <- length(Beverages\_Sale2)

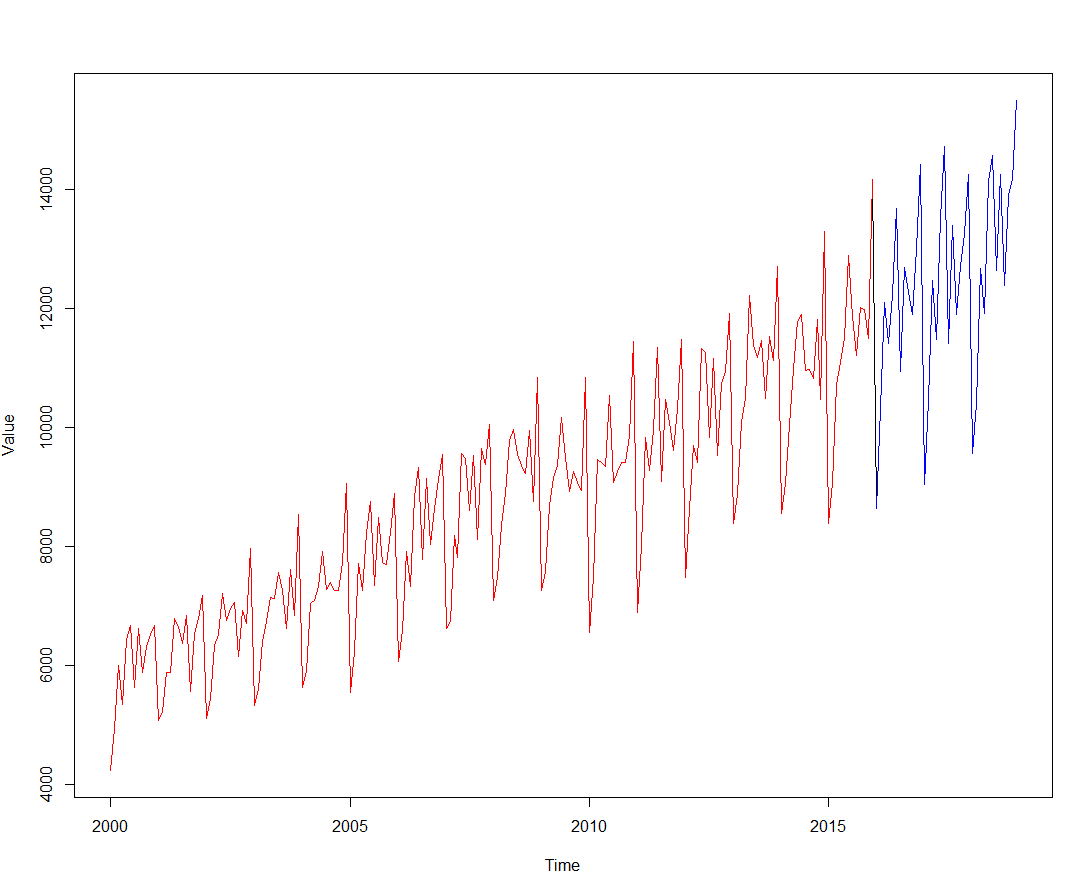
First we investigate the data set. We have seasonal (monthly) data, so we also explore the seasonal properties of the time series.

# Plot the data

plot(Beverages\_Sale)

lines(Beverages\_Sale1, col="red")

lines(Beverages\_Sale2, col="blue")



We note an increasing trend in the time series, and high seasonality. We investigate the seasonality by means of a seasonplot and a monthplot.

par(mfrow=c(1,2))

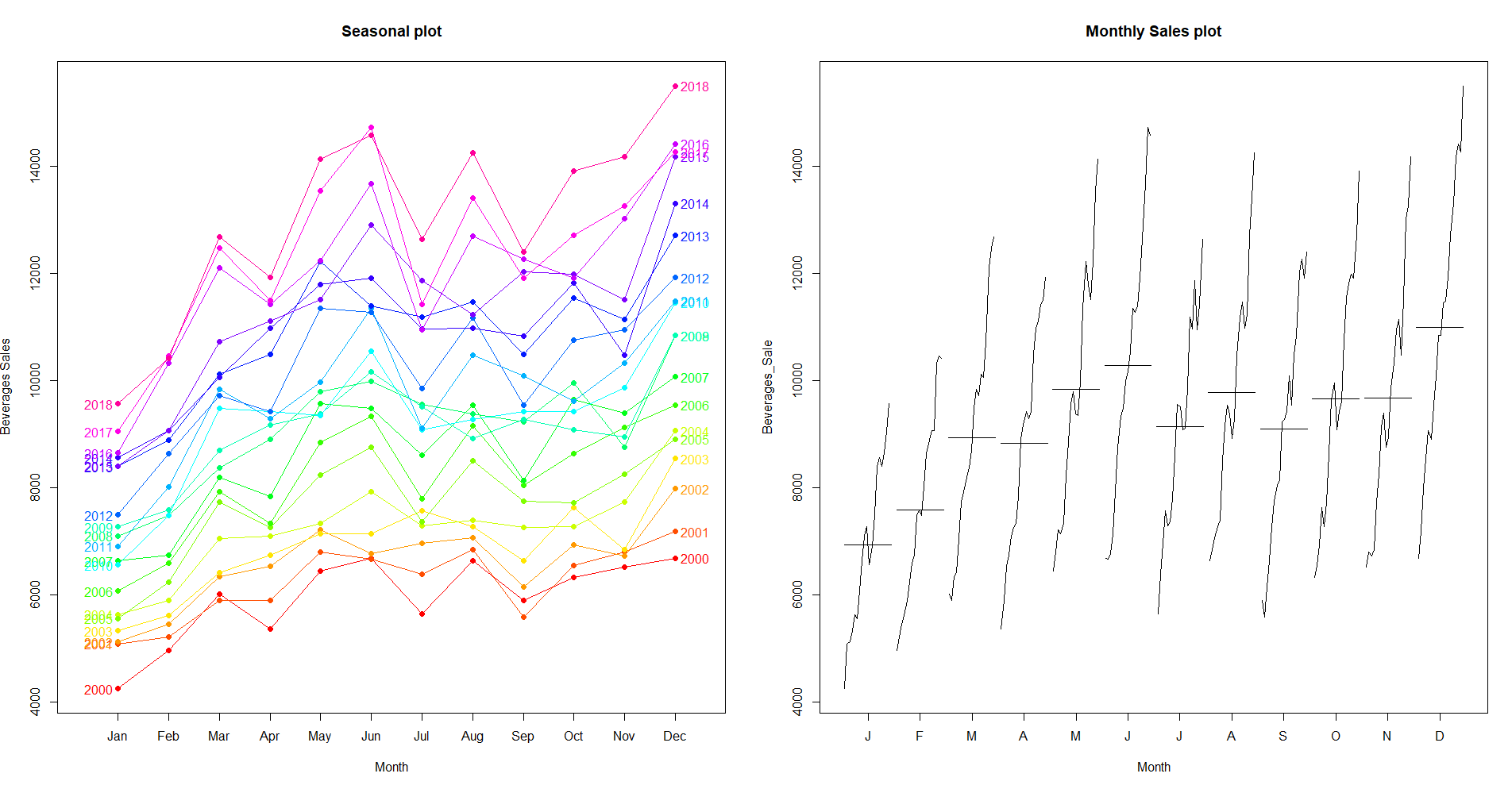
seasonplot(Beverages\_Sale, year.labels=TRUE, year.labels.left=TRUE,

main="Seasonal plot",

ylab="Beverages Sales",col=rainbow(20), pch=19)

monthplot(Beverages\_Sale, main="Monthly Sales plot", ylab = "Beverages\_Sale",

xlab="Month", type="l")



We can see that beverages sales are low at the start of the year and it increases till June and then it becomes stable and we can see increasing trend in the recent years too.

As there is high seasonality and trend so we will first do forecasting using seasonal naïve method.

**Seasonal Naïve Method**

n <- snaive(Beverages\_Sale1, h=h) # seasonal naive

a\_n <- accuracy(n,Beverages\_Sale2)[,c(2,3,5,6)]

a\_train\_n <- a\_n[1,]

a\_train\_n

RMSE MAE MAPE MASE

552.705321 456.105556 5.206858 1.000000

a\_test\_n <- a\_n[2,]

a\_test\_n

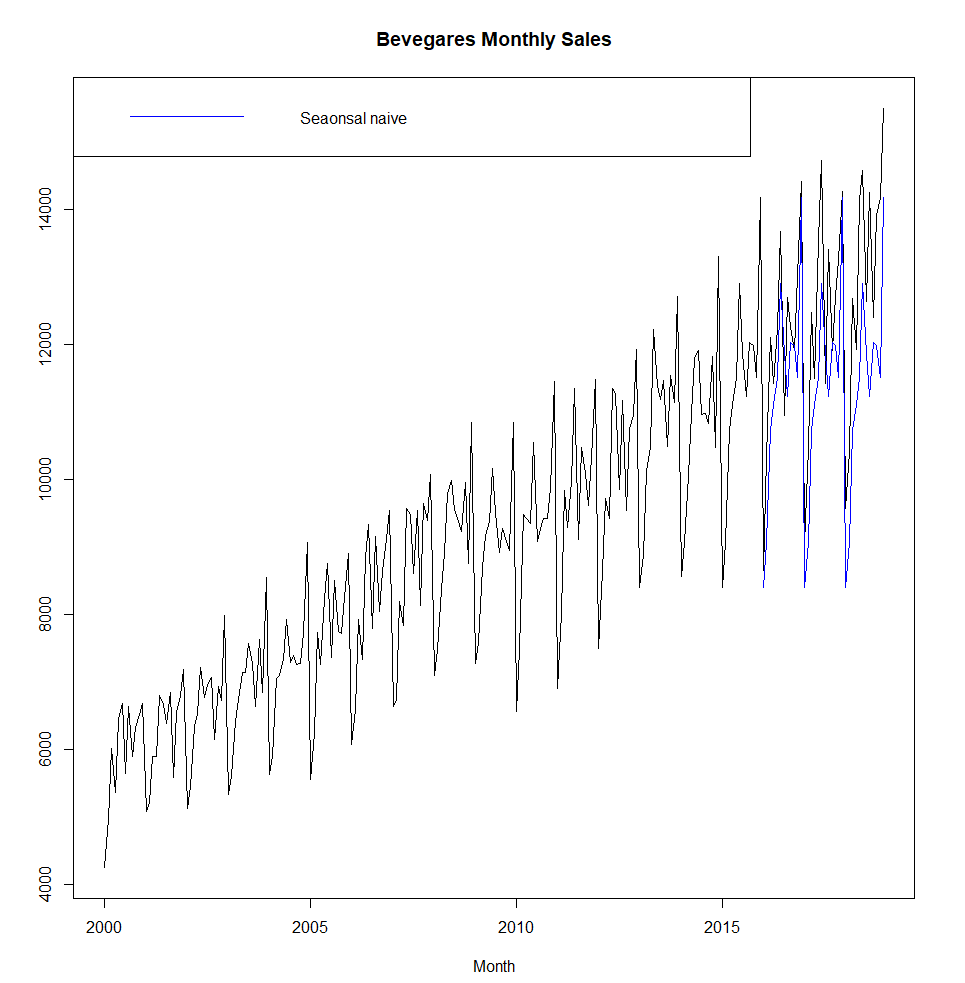
RMSE MAE MAPE MASE

1412.535132 1173.055556 9.222635 2.571895

plot(Beverages\_Sale,main="Bevegares Monthly Sales", ylab="",xlab="Month")

lines(n$mean,col=4)

legend("topleft",lty=1,col=c(4),legend=c("Seaonsal naive"))



By visual inspection, it looks like seasonal naïve method shows acceptable result.

Now we check the quality of residuals.

res <- residuals(n)

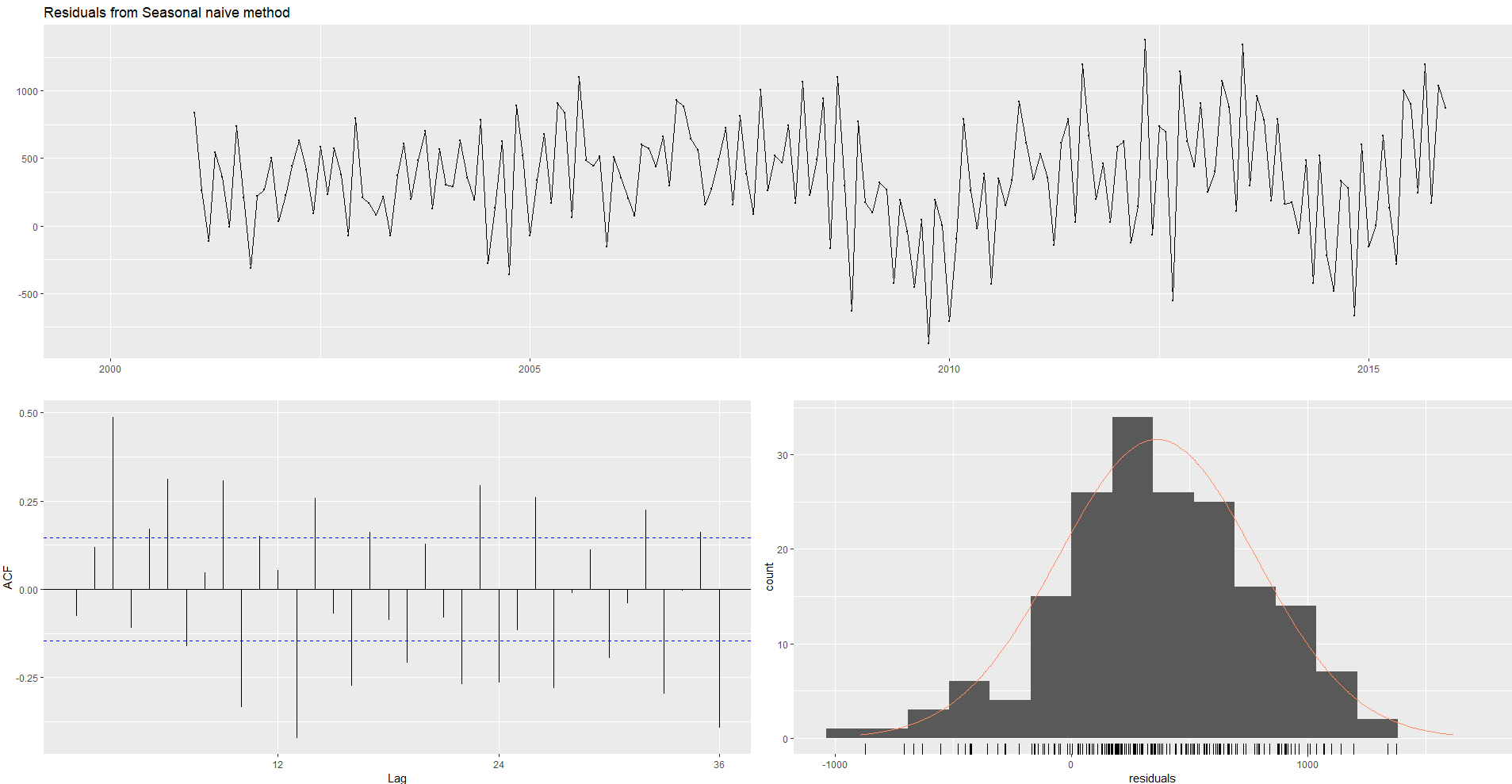
checkresiduals(n)

Ljung-Box test

data: Residuals from Seasonal naive method

Q\* = 256.23, df = 24, p-value < 2.2e-16

Model df: 0. Total lags used: 24



res <- na.omit(res)

LjungBox(res, lags=seq(1,24,4), order=0)

> res <- na.omit(res)

> LjungBox(res, lags=seq(1,24,4), order=0)

lags statistic df p-value

1 1.056043 1 3.041195e-01

5 55.458121 5 1.050662e-10

9 97.379702 9 0.000000e+00

13 158.903888 13 0.000000e+00

17 193.227701 17 0.000000e+00

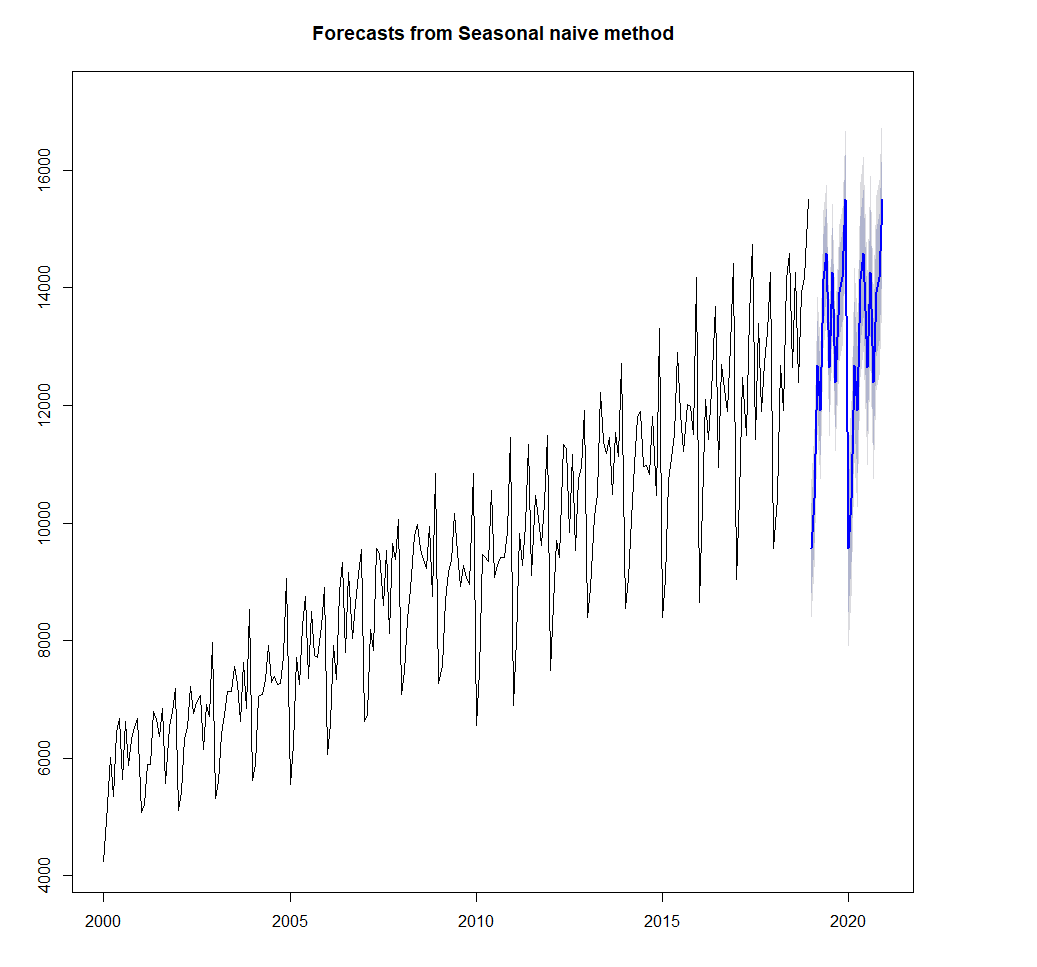
21 208.437135 21 0.000000e+00

The residual diagnostics shows that the residuals of this seasonal naïve method are not white noise.

A complete forecast on this dataset looks as follows.

n\_final <- snaive(Beverages\_Sale, h=24)

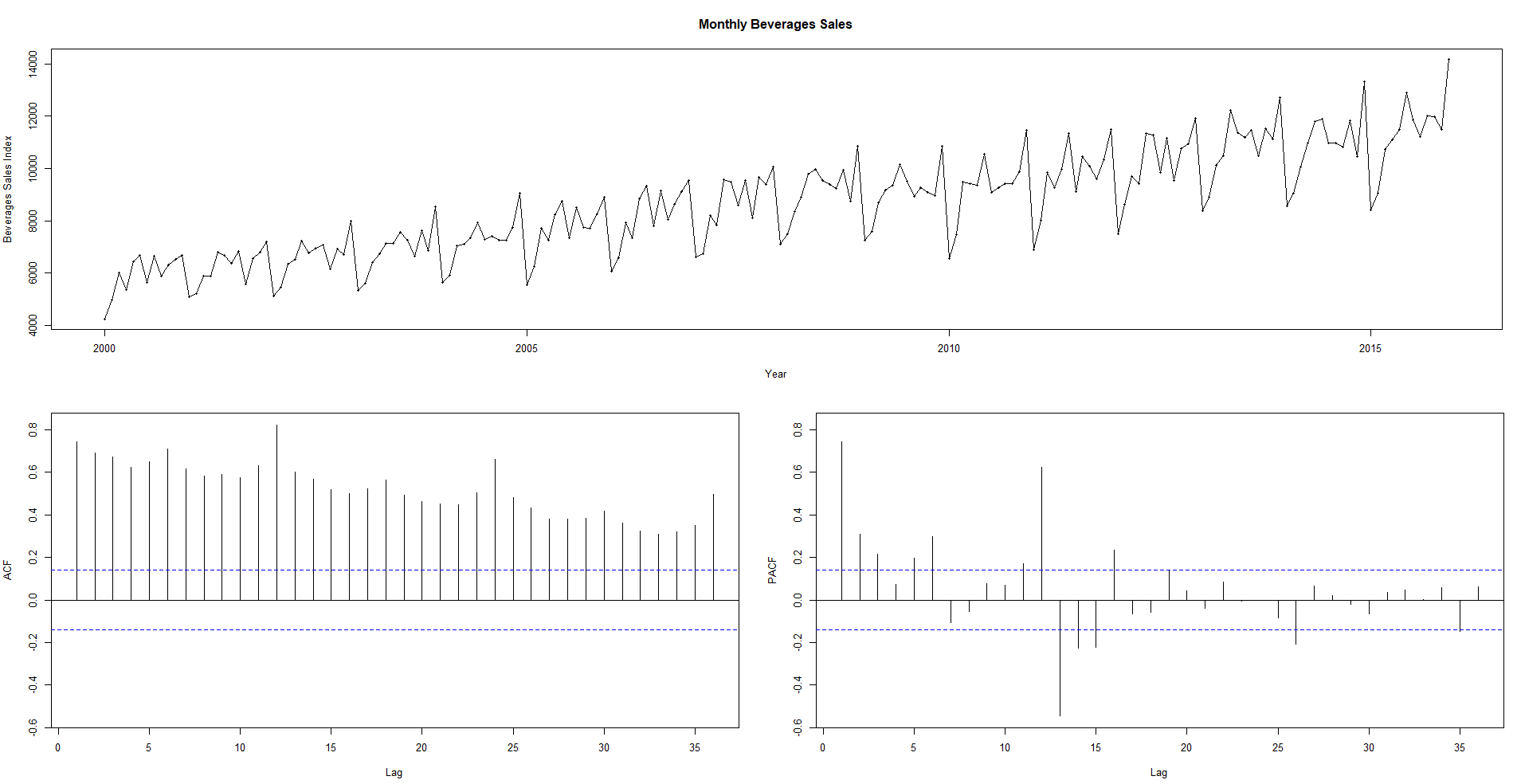
plot(n\_final)



**ARIMA MODELS**

Now I will forecast this time series using ARIMA Models. We further investigate the characteristics of time series.

tsdisplay(Beverages\_Sale1, main="Monthly Beverages Sales", ylab="Beverages Sales Index", xlab="Year")



The ACF shows that non stationarity is caused by both seasonality and trend. We start by differencing the data ndiffs suggest one difference and nsdiffs suggest one difference too.

> ndiffs(Beverages\_Sale1)

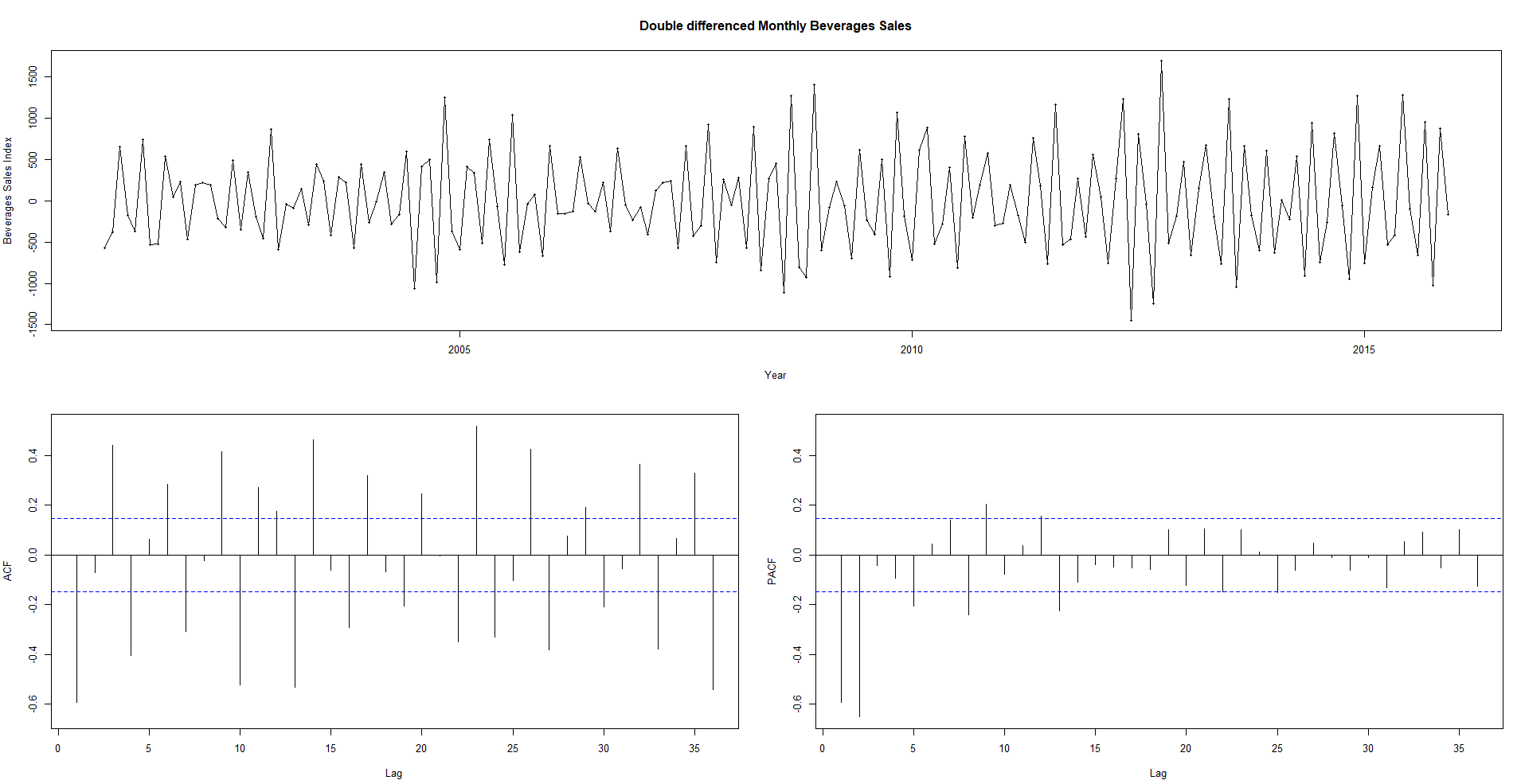
[1] 1

> nsdiffs(diff(Beverages\_Sale1))

[1] 1

The characteristics of a double differenced time series are as follows.

tsdisplay(diff(diff(Beverages\_Sale1,12)), main="Double differenced Monthly Beverages Sales", ylab="Beverages Sales Index", xlab="Year")

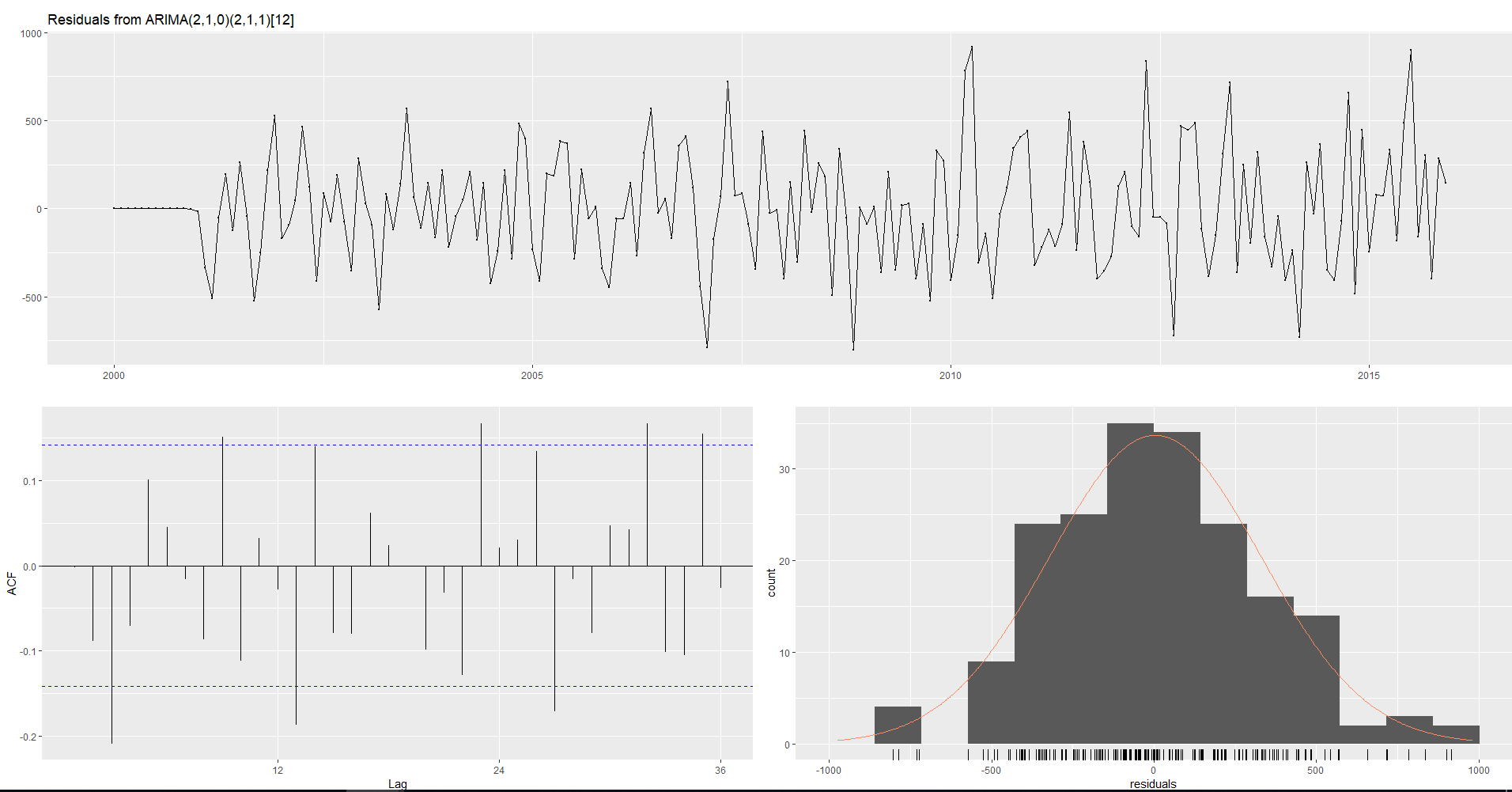


**Model Estimation**

We start with auto arima method to get initial idea of the suitable model. We disable the stepwise and approximate search and ask for first and seasonal differences.

m0 <- auto.arima(Beverages\_Sale1, stepwise = FALSE, approximation = FALSE, d=1, D=1)

checkresiduals(m0)



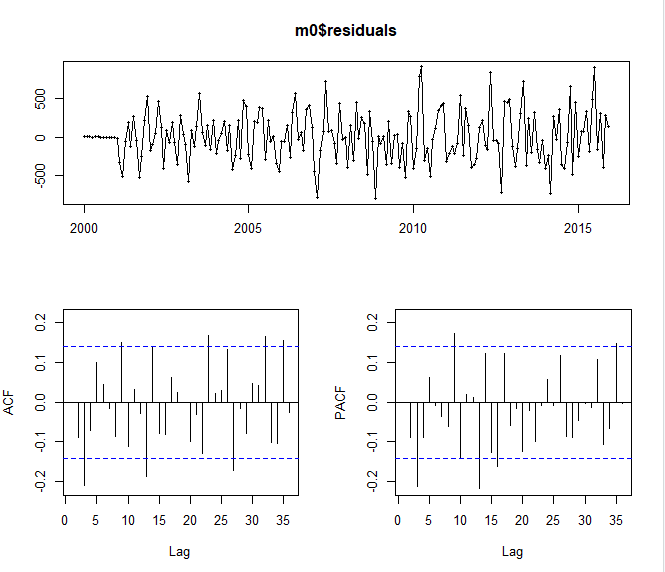
Ljung-Box test

data: Residuals from ARIMA(2,1,0)(2,1,1)[12]

Q\* = 49.919, df = 19, p-value = 0.0001347

Model df: 5. Total lags used: 24

tsdisplay(m0$residuals)



LjungBox(m0$residuals, lags=seq(length(m0$coef),24,4), order=length(m0$coef))

lags statistic df p-value

5 13.15852 0 0.0000000000

9 19.75793 4 0.0005574512

13 29.94818 8 0.0002158836

17 37.53970 12 0.0001825537

21 40.00760 16 0.0007766025

f0 <- forecast(m0, h=h)

accuracy(f0,Beverages\_Sale2)[,c(2,3,5,6)]

RMSE MAE MAPE MASE

Training set 324.9917 253.4818 2.926817 0.5557525

Test set 727.1139 565.0179 4.857477 1.2387876

The auto.arima procedure results in an ARIMA(2,1,0)(2,1,1)[12] model. This model shows satisfactory diagnostics. The residuals behave reasonably well. We will now explore some variations starting from this model and check model fit and forecast accuracy.

getinfo <- function(x,h,...)

{

train.end <- time(x)[length(x)-h]

test.start <- time(x)[length(x)-h+1]

train <- window(x,end=train.end)

test <- window(x,start=test.start)

fit <- Arima(train,...)

fc <- forecast(fit,h=h)

a <- accuracy(fc,test)

result <- matrix(NA, nrow=1, ncol=5)

result[1,1] <- fit$aicc

result[1,2] <- a[1,6]

result[1,3] <- a[2,6]

result[1,4] <- a[1,2]

result[1,5] <- a[2,2]

return(result)

}

mat <- matrix(NA,nrow=54, ncol=5)

modelnames <- vector(mode="character", length=54)

line <- 0

for (i in 2:4){

for (j in 0:2){

for (k in 0:1){

for (l in 0:2){

line <- line+1

mat[line,] <- getinfo(Beverages\_Sale,h=37,order=c(i,1,j),seasonal=c(k,1,l))

modelnames[line] <- paste0("ARIMA(",i,",1,",j,")(",k,",1,",l,")[12]")

}

}

}

}

colnames(mat) <- c("AICc", "MASE\_train", "MASE\_test", "RMSE\_train", "RMSE\_test")

rownames(mat) <- modelnames

# best AICc

mat[mat[,1]==min(mat[,1])]

[1] 2590.9868160 0.5402703 1.2922811 315.8832245 725.1746251

#best MASE\_train

mat[mat[,2]==min(mat[,2])]

[1] 2592.6871710 0.5389859 1.2854320 315.1400483 726.3855263

#best MASE\_test

mat[mat[,3]==min(mat[,3])]

[1] 2597.6066284 0.5463284 1.1986874 320.8525765 667.4803119

#best RMSE\_train

mat[mat[,4]==min(mat[,4])]

2592.6871710 0.5389859 1.2854320 315.1400483 726.3855263

#best RMSE\_test

mat[mat[,5]==min(mat[,5])]

[1] 2602.0189428 0.5753083 1.2249512 334.6797275 653.3097600

Based on the analysis we select 4 most suitable models which can bring us closer to the most suitable results.

1. m0: ARIMA(2,1,0)(2,1,1)is the model selected by auto.arima. It shows acceptable fit, forecasting

accuracy and residual diagnostics.

2. m1: ARIMA (4,1,1) (0,1,2) shows the best RMSE and MASE on the training set.

3. m2: ARIMA(2,1,0)(1,1,1) shows the best RMSE on the test set.

4.m3: ARIMA(4,1,2)(0,1,2) shows the best MASE on test set.

5. m4:ARIMA(3,1,1)(0,1,2) shows the best AIC

We now study these models in more details.

#Lowest RMSE Train

m1 <- Arima(Beverages\_Sale1, order=c(4,1,1), seasonal=c(0,1,2))

LjungBox(m1$residuals, lags=seq(length(m1$coef),24,4), order=length(m1$coef))

lags statistic df p-value

7 3.196755 0 0.000000e+00

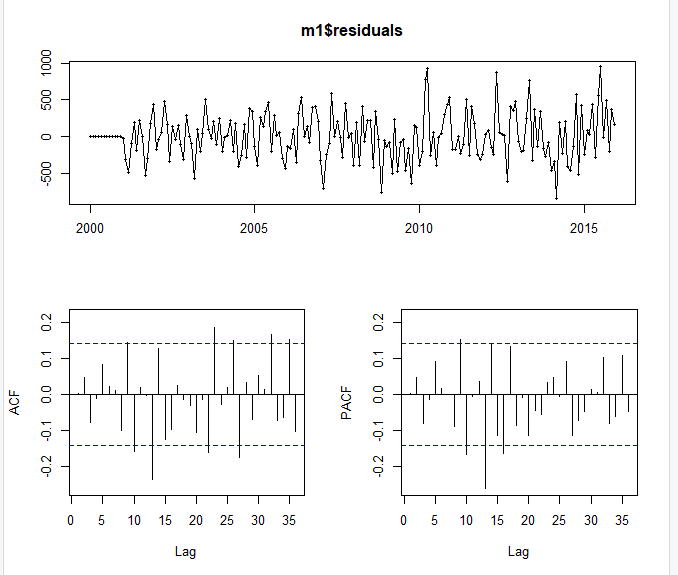
11 14.583189 4 5.648574e-03

15 32.762000 8 6.797518e-05

19 35.131426 12 4.462620e-04

23 50.858991 16 1.671342e-05

tsdisplay(m1$residuals)



f1 <- forecast(m1, h=h)

#LOWEST RMSE TEST

m2 <- Arima(Beverages\_Sale1, order=c(2,1,0), seasonal=c(1,1,1))

LjungBox(m2$residuals, lags=seq(length(m2$coef),24,4), order=length(m2$coef))

lags statistic df p-value

4 10.20421 0 0.000000e+00

8 16.11439 4 2.869447e-03

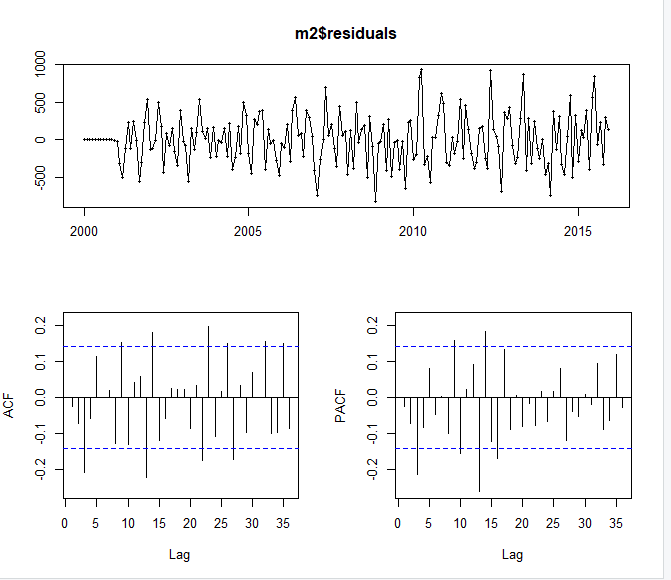
12 25.32071 8 1.371430e-03

16 45.84816 12 7.368840e-06

20 47.79977 16 5.106403e-05

24 65.82267 20 8.624711e-07

tsdisplay(m2$residuals)



f2 <- forecast(m2, h=h)

#Lowest MASE TEST

m3 <- Arima(Beverages\_Sale1, order=c(4,1,2), seasonal=c(0,1,2))

LjungBox(m3$residuals, lags=seq(length(m3$coef),24,4), order=length(m3$coef))

lags statistic df p-value

8 4.348095 0 0.000000e+00

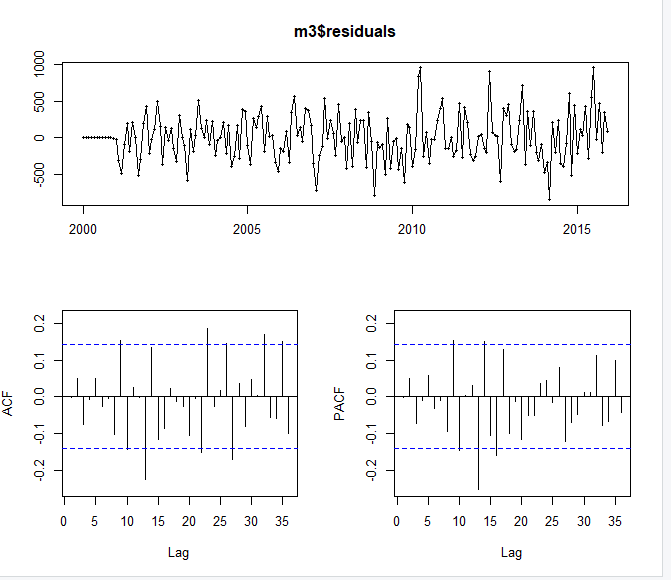
12 13.314827 4 9.835675e-03

16 32.125170 8 8.845383e-05

20 34.854191 12 4.940107e-04

24 47.532839 16 5.622542e-05

tsdisplay(m3$residuals)



f3 <- forecast(m3, h=h)

#Lowest AIC

m4 <- Arima(Beverages\_Sale1, order=c(3,1,1), seasonal=c(0,1,2))

LjungBox(m4$residuals, lags=seq(length(m4$coef),24,4), order=length(m4$coef))

lags statistic df p-value

6 7.040592 0 0.0000000000

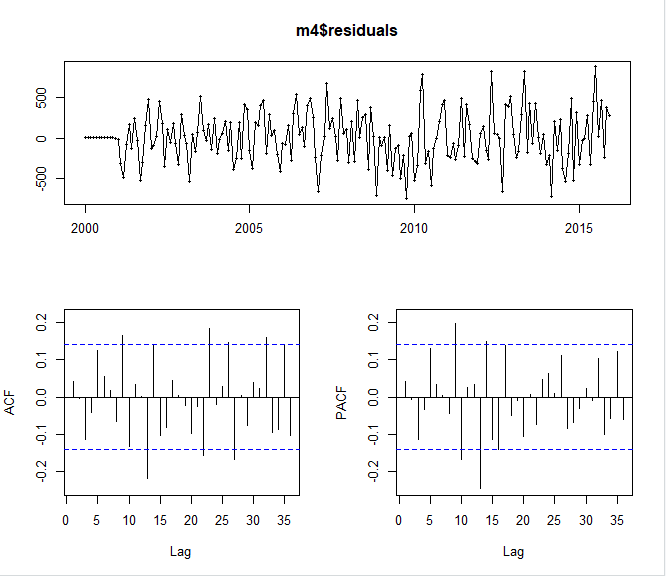
10 17.150684 4 0.0018068851

14 31.442556 8 0.0001171687

18 35.482216 12 0.0003922516

22 43.103038 16 0.0002696382

tsdisplay(m4$residuals)



f4 <- forecast(m4, h=h)

We bring together the relevant accuracy measures in the tables below.

a\_m0 <- accuracy(f0,Beverages\_Sale2)[,c(2,3,5,6)]

a\_m1 <- accuracy(f1,Beverages\_Sale2)[,c(2,3,5,6)]

a\_m2 <- accuracy(f2,Beverages\_Sale2)[,c(2,3,5,6)]

a\_m3 <- accuracy(f3,Beverages\_Sale2)[,c(2,3,5,6)]

a\_m4 <- accuracy(f4,Beverages\_Sale2)[,c(2,3,5,6)]

a\_train\_a <- rbind(a\_m0[1,], a\_m1[1,], a\_m2[1,], a\_m3[1,], a\_m4[1,])

rownames(a\_train\_a) <- c("a\_m0", "a\_m1", "a\_m2", "a\_m3","a\_m4" )

a\_train\_a

RMSE MAE MAPE MASE

a\_m0 324.9917 253.4818 2.926817 0.5557525

a\_m1 321.4218 248.5021 2.853528 0.5448345

a\_m2 333.9986 260.5919 3.003492 0.5713411

a\_m3 320.0965 247.2081 2.844193 0.5419977

a\_m4 315.8709 245.4285 2.818089 0.5380958

a\_test\_a <- rbind(a\_m0[2,], a\_m1[2,], a\_m2[2,], a\_m3[2,], a\_m4[2,])

rownames(a\_test\_a) <- c("a\_m0", "a\_m1", "a\_m2", "a\_m3","a\_m4" )

a\_test\_a

RMSE MAE MAPE MASE

a\_m0 727.1139 565.0179 4.857477 1.238788

a\_m1 654.7347 546.6748 4.587902 1.198571

a\_m2 676.3698 568.4955 4.754321 1.246412

a\_m3 690.7391 560.3196 4.761846 1.228487

a\_m4 699.0933 562.7486 4.400901 1.233812

We can observe that m1 ARIMA(2,1,0)(1,1,1) is the best model in terms of RMSE. While its residuals are not a white noise.

**Final Model And Forecasts**

We select M1 as our final model ARIMA(4,1,1)(0,1,2). We show the estimated parameters and the generated forecasts below. Note that the prediction intervals around the forecasts are quite small, indicating that our forecast has reasonably high precision.

a\_final <- Arima(Beverages\_Sale, order=c(2,1,0), seasonal=c(1,1,1))

summary(a\_final)

ARIMA(4,1,1)(0,1,2)[12]

Coefficients:

ar1 ar2 ar3 ar4 ma1 sma1 sma2

-0.0853 0.1644 0.4768 -0.0877 -0.9626 -0.2610 -0.3009

s.e. 0.0778 0.0741 0.0747 0.0727 0.0393 0.0714 0.0693

sigma^2 estimated as 131503: log likelihood=-1572.18

AIC=3160.35 AICc=3161.05 BIC=3187.32

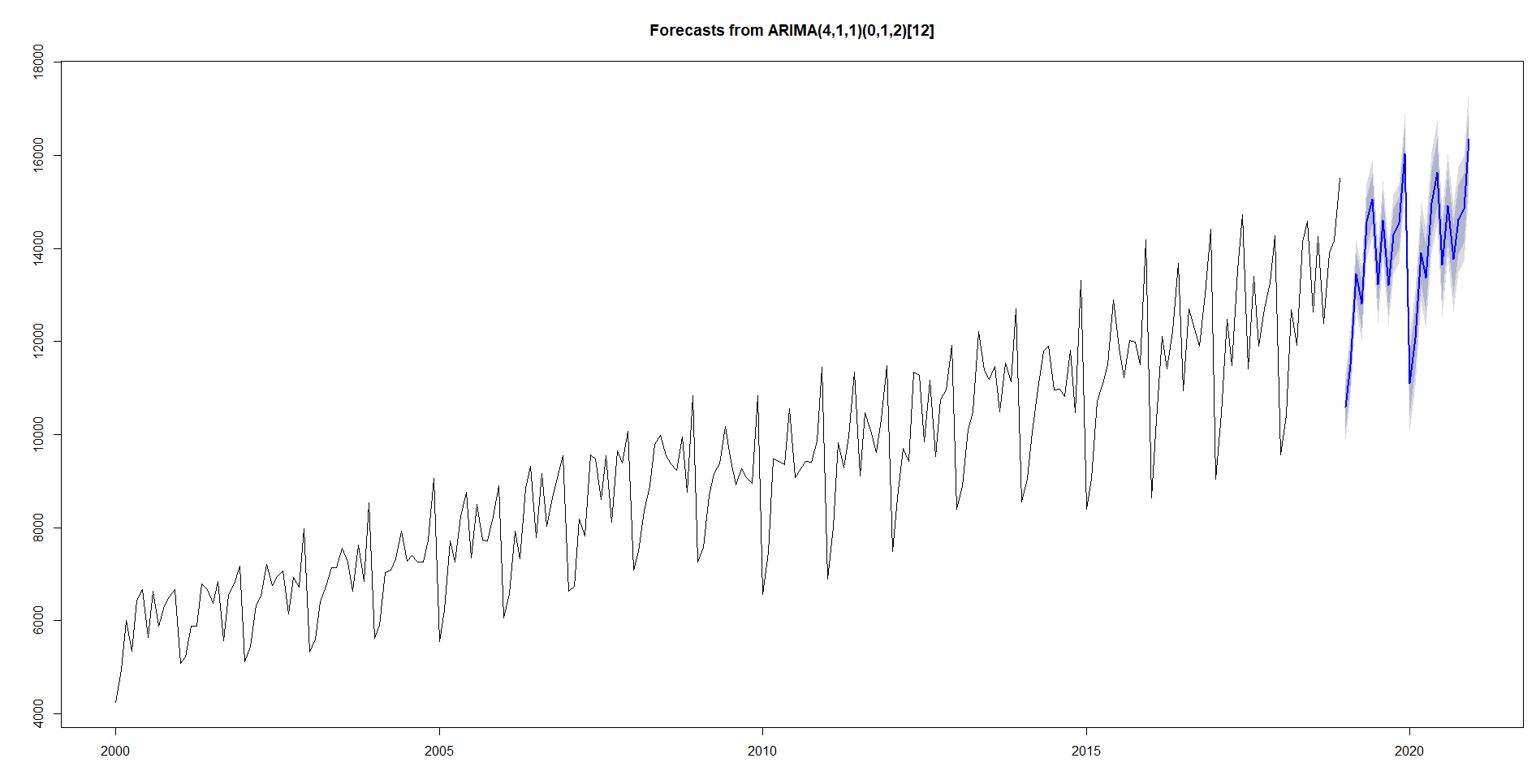
Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set 20.5302 346.3633 266.3443 -0.03927612 2.864685 0.5468665 0.005657076

a\_final\_f <- forecast(a\_final, h=24)

plot(a\_final\_f)



**Conclusion**

In this section, we compare the performance of the selected models, being the seasonal naive

ARIMA. We forecast accuracy is summarized in the tables below.

final\_train <- rbind(a\_train\_n, a\_train\_a[3,])

rownames(final\_train) <- c("snaive", "ARIMA(4,1,2)(1,1,2)[12]")

final\_train

final\_train

RMSE MAE MAPE MASE

snaive 552.7053 456.1056 5.206858 1.0000000

ARIMA(4,1,2)(1,1,2)[12] 333.9986 260.5919 3.003492 0.5713411

final\_test

RMSE MAE MAPE MASE

a\_test\_n 1412.5351 1173.0556 9.222635 2.571895

ARIMA(4,1,2)(1,1,2)[12] 676.3698 568.4955 4.754321 1.246412

We observe that the selected ARIMA(4,1,2)(1,1,2) model performs best on the training and test set. However, the residual diagnostics were not satisfactory for both of the selected models i.e Snaive and ARIMA.

Overall, we consider ARIMA(4,1,2)(1,1,2)as a well performing model. This will be the final model for generating the forecasts up to 2020.

ARIMA(4,1,1)(0,1,2)[12]

Coefficients:

ar1 ar2 ar3 ar4 ma1 sma1 sma2

-0.3542 -0.1328 0.2651 -0.1333 -0.6814 -0.2689 -0.3073

s.e. 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

sigma^2 estimated as 115325: log likelihood=-1573.55

AIC=3149.1 AICc=3149.12 BIC=3152.47

Training set error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set 11.81254 349.3948 267.5671 -0.1245342 2.876961 0.5493772 0.0147229

plot(forecast(model, h=24), include = 80)

