Forecast the PlasticSales data set. Prepare a document for each model explaining how many dummy variables you have created and RMSE value for each model. Finally which model you will use for Forecasting.

**Ans :**

**R Code :**

## Forecasting Model Driven

########## PlasticSales Data Set #########

library(readr)

plasticsales <- read.csv('D:\\Data Science\\Excelr\\Assignments\\Assignment\\Forecasting\\PlasticSales.csv')

View(plasticsales) # Seasonality 12 months

windows()

plot(plasticsales$Sales,type="l")

# So creating 11 dummy variables

X<- data.frame(outer(rep(month.abb,length = 60), month.abb,"==") + 0 )# Creating dummies for 12 months

View(X)

colnames(X)<-month.abb # Assigning month names

View(X)

plastic<-cbind(plasticsales,X)

View(plastic)

plastic["t"]<- 1:60

View(plastic)

plastic["log\_sales"]<-log(plastic["Sales"])

plastic["t\_square"]<-plastic["t"]\*plastic["t"]

##Data Partition

train<-plastic[1:48,]

test<-plastic[49:60,]

########################### LINEAR MODEL #############################

linear\_model<-lm(Sales~t,data=train)

summary(linear\_model)

linear\_pred<-data.frame(predict(linear\_model,interval='predict',newdata =test))

View(linear\_pred)

rmse\_linear<-sqrt(mean((test$Sales-linear\_pred$fit)^2,na.rm = T))

rmse\_linear

######################### Exponential #################################

expo\_model<-lm(log\_sales~t,data=train)

summary(expo\_model)

expo\_pred<-data.frame(predict(expo\_model,interval='predict',newdata=test))

rmse\_expo<-sqrt(mean((test$Sales-exp(expo\_pred$fit))^2,na.rm = T))

rmse\_expo

######################### Quadratic ####################################

Quad\_model<-lm(Sales~t+t\_square,data=train)

summary(Quad\_model)

Quad\_pred<-data.frame(predict(Quad\_model,interval='predict',newdata=test))

rmse\_Quad<-sqrt(mean((test$Sales-Quad\_pred$fit)^2,na.rm=T))

rmse\_Quad

######################### Additive Seasonality #########################

sea\_add\_model<-lm(Sales~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=train)

summary(sea\_add\_model)

sea\_add\_pred<-data.frame(predict(sea\_add\_model,newdata=test,interval='predict'))

rmse\_sea\_add<-sqrt(mean((test$Sales-sea\_add\_pred$fit)^2,na.rm = T))

rmse\_sea\_add

######################## Additive Seasonality with Linear #################

Add\_sea\_Linear\_model<-lm(Sales~t+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=train)

summary(Add\_sea\_Linear\_model)

Add\_sea\_Linear\_pred<-data.frame(predict(Add\_sea\_Linear\_model,interval='predict',newdata=test))

rmse\_Add\_sea\_Linear<-sqrt(mean((test$Sales-Add\_sea\_Linear\_pred$fit)^2,na.rm=T))

rmse\_Add\_sea\_Linear

######################## Additive Seasonality with Quadratic #################

Add\_sea\_Quad\_model<-lm(Sales~t+t\_square+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=train)

summary(Add\_sea\_Quad\_model)

Add\_sea\_Quad\_pred<-data.frame(predict(Add\_sea\_Quad\_model,interval='predict',newdata=test))

rmse\_Add\_sea\_Quad<-sqrt(mean((test$Sales-Add\_sea\_Quad\_pred$fit)^2,na.rm=T))

rmse\_Add\_sea\_Quad

######################## Multiplicative Seasonality #########################

multi\_sea\_model<-lm(log\_sales~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train)

summary(multi\_sea\_model)

multi\_sea\_pred<-data.frame(predict(multi\_sea\_model,newdata=test,interval='predict'))

rmse\_multi\_sea<-sqrt(mean((test$Sales-exp(multi\_sea\_pred$fit))^2,na.rm = T))

rmse\_multi\_sea

######################## Multiplicative Seasonality Linear trend ##########################

multi\_add\_sea\_model<-lm(log\_sales~t+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train)

summary(multi\_add\_sea\_model)

multi\_add\_sea\_pred<-data.frame(predict(multi\_add\_sea\_model,newdata=test,interval='predict'))

rmse\_multi\_add\_sea<-sqrt(mean((test$Sales-exp(multi\_add\_sea\_pred$fit))^2,na.rm = T))

rmse\_multi\_add\_sea

# Preparing table on model and it's RMSE values

table\_rmse<-data.frame('Model'=c("rmse\_linear","rmse\_expo","rmse\_Quad","rmse\_sea\_add","rmse\_Add\_sea\_Quad","rmse\_multi\_sea","rmse\_multi\_add\_sea"),'RMSE'=c(rmse\_linear,rmse\_expo,rmse\_Quad,rmse\_sea\_add,rmse\_Add\_sea\_Quad,rmse\_multi\_sea,rmse\_multi\_add\_sea))

View(table\_rmse)

colnames(table\_rmse)<-c("model","RMSE")

View(table\_rmse)

# Use entire data : Multiplicative Seasonality Linear trend has least RMSE value

new\_model <- lm(log\_sales~t+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=plastic)

#predict(new\_model,n.ahead=1)

# Getting residuals

resid <- residuals(new\_model)

resid[1:10]

windows()

hist(resid)

windows()

acf(resid,lag.max = 10)

# By principal of parcimony we will consider lag - 1 as we have so

# many significant lags

# Building Autoregressive model on residuals consider lag-1

k <- arima(resid, order=c(1,0,0))

windows();

acf(k$residuals,lag.max = 15)

pred\_res<- predict(arima(resid,order=c(1,0,0)),n.ahead = 12)

str(pred\_res)

pred\_res$pred

acf(k$residuals)

####################### Predicting new data #############################

library(readxl)

test\_data<-read\_excel(file.choose(),1) #Load Predict\_new\_PlasticData.xlsx

View(test\_data)

#test\_data<-Predict\_new

pred\_new<-data.frame(predict(new\_model,newdata=test\_data,interval = 'predict'))

View(pred\_new)

#pred\_re<-pred\_res$pred[1:12]

pred\_new$fit <- pred\_new$fit+pred\_res$pred[1:12]

View(pred\_new)

**Results :**

> linear\_model<-lm(Sales~t,data=train)

> summary(linear\_model)

Call:

lm(formula = Sales ~ t, data = train)

Residuals:

Min 1Q Median 3Q Max

-403.95 -192.19 13.41 206.68 274.40

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 863.109 62.460 13.819 < 2e-16 \*\*\*

t 10.575 2.219 4.765 1.92e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 213 on 46 degrees of freedom

Multiple R-squared: 0.3305, Adjusted R-squared: 0.3159

F-statistic: 22.71 on 1 and 46 DF, p-value: 1.925e-05

> linear\_pred<-data.frame(predict(linear\_model,interval='predict',newdata =test))

> View(linear\_pred)

> rmse\_linear<-sqrt(mean((test$Sales-linear\_pred$fit)^2,na.rm = T))

> rmse\_linear

[1] 260.9378

> expo\_model<-lm(log\_sales~t,data=train)

> summary(expo\_model)

Call:

lm(formula = log\_sales ~ t, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.36696 -0.17856 0.02323 0.17933 0.26472

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.762209 0.058119 116.351 < 2e-16 \*\*\*

t 0.009549 0.002065 4.624 3.07e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1982 on 46 degrees of freedom

Multiple R-squared: 0.3173, Adjusted R-squared: 0.3025

F-statistic: 21.38 on 1 and 46 DF, p-value: 3.066e-05

> expo\_pred<-data.frame(predict(expo\_model,interval='predict',newdata=test))

> rmse\_expo<-sqrt(mean((test$Sales-exp(expo\_pred$fit))^2,na.rm = T))

> rmse\_expo

[1] 268.6938

> Quad\_model<-lm(Sales~t+t\_square,data=train)

> summary(Quad\_model)

Call:

lm(formula = Sales ~ t + t\_square, data = train)

Residuals:

Min 1Q Median 3Q Max

-403.04 -199.43 3.02 211.08 290.41

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 901.20195 96.98668 9.292 4.93e-12 \*\*\*

t 6.00348 9.13067 0.658 0.514

t\_square 0.09329 0.18066 0.516 0.608

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 214.7 on 45 degrees of freedom

Multiple R-squared: 0.3344, Adjusted R-squared: 0.3048

F-statistic: 11.31 on 2 and 45 DF, p-value: 0.0001052

> Quad\_pred<-data.frame(predict(Quad\_model,interval='predict',newdata=test))

> rmse\_Quad<-sqrt(mean((test$Sales-Quad\_pred$fit)^2,na.rm=T))

> rmse\_Quad

[1] 297.4067

> sea\_add\_model<-lm(Sales~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=train)

> summary(sea\_add\_model)

Call:

lm(formula = Sales ~ Jan + Feb + Mar + Apr + May + Jun + Jul +

Aug + Sep + Oct + Nov, data = train)

Residuals:

Min 1Q Median 3Q Max

-239.00 -80.12 -14.50 85.94 250.50

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 979.00 70.70 13.847 5.44e-16 \*\*\*

Jan -146.50 99.99 -1.465 0.151543

Feb -216.25 99.99 -2.163 0.037275 \*

Mar -135.75 99.99 -1.358 0.183010

Apr 19.50 99.99 0.195 0.846467

May 172.75 99.99 1.728 0.092604 .

Jun 290.50 99.99 2.905 0.006236 \*\*

Jul 332.00 99.99 3.320 0.002068 \*\*

Aug 410.00 99.99 4.101 0.000225 \*\*\*

Sep 427.50 99.99 4.276 0.000134 \*\*\*

Oct 391.00 99.99 3.911 0.000391 \*\*\*

Nov 173.50 99.99 1.735 0.091251 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 141.4 on 36 degrees of freedom

Multiple R-squared: 0.7691, Adjusted R-squared: 0.6985

F-statistic: 10.9 on 11 and 36 DF, p-value: 1.84e-08

> sea\_add\_pred<-data.frame(predict(sea\_add\_model,newdata=test,interval='predict'))

> rmse\_sea\_add<-sqrt(mean((test$Sales-sea\_add\_pred$fit)^2,na.rm = T))

> rmse\_sea\_add

[1] 235.6027

> Add\_sea\_Linear\_model<-lm(Sales~t+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=train)

> summary(Add\_sea\_Linear\_model)

Call:

lm(formula = Sales ~ t + Jan + Feb + Mar + Apr + May + Jun +

Jul + Aug + Sep + Oct + Nov, data = train)

Residuals:

Min 1Q Median 3Q Max

-84.388 -28.256 -6.037 16.319 95.887

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 721.3125 28.8624 24.991 < 2e-16 \*\*\*

t 8.5896 0.5218 16.463 < 2e-16 \*\*\*

Jan -52.0146 34.7706 -1.496 0.143634

Feb -130.3542 34.6883 -3.758 0.000625 \*\*\*

Mar -58.4437 34.6137 -1.688 0.100215

Apr 88.2167 34.5468 2.554 0.015177 \*

May 232.8771 34.4876 6.752 7.99e-08 \*\*\*

Jun 342.0375 34.4363 9.932 1.01e-11 \*\*\*

Jul 374.9479 34.3928 10.902 8.44e-13 \*\*\*

Aug 444.3583 34.3571 12.934 6.80e-15 \*\*\*

Sep 453.2688 34.3294 13.204 3.72e-15 \*\*\*

Oct 408.1792 34.3095 11.897 7.47e-14 \*\*\*

Nov 182.0896 34.2976 5.309 6.29e-06 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 48.5 on 35 degrees of freedom

Multiple R-squared: 0.9736, Adjusted R-squared: 0.9645

F-statistic: 107.5 on 12 and 35 DF, p-value: < 2.2e-16

> Add\_sea\_Linear\_pred<-data.frame(predict(Add\_sea\_Linear\_model,interval='predict',newdata=test))

> rmse\_Add\_sea\_Linear<-sqrt(mean((test$Sales-Add\_sea\_Linear\_pred$fit)^2,na.rm=T))

> rmse\_Add\_sea\_Linear

[1] 135.5536

> Add\_sea\_Quad\_model<-lm(Sales~t+t\_square+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=train)

> summary(Add\_sea\_Quad\_model)

Call:

lm(formula = Sales ~ t + t\_square + Jan + Feb + Mar + Apr + May +

Jun + Jul + Aug + Sep + Oct + Nov, data = train)

Residuals:

Min 1Q Median 3Q Max

-87.014 -25.254 4.075 25.251 52.423

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 778.21577 26.68286 29.165 < 2e-16 \*\*\*

t 1.44020 1.67520 0.860 0.3960

t\_square 0.14591 0.03308 4.410 9.85e-05 \*\*\*

Jan -52.01458 28.13679 -1.849 0.0732 .

Feb -128.89511 28.07214 -4.592 5.78e-05 \*\*\*

Mar -55.81745 28.01612 -1.992 0.0544 .

Apr 91.71841 27.96692 3.280 0.0024 \*\*

May 236.96245 27.92315 8.486 6.54e-10 \*\*\*

Jun 346.41467 27.88389 12.423 3.42e-14 \*\*\*

Jul 379.32509 27.84871 13.621 2.50e-15 \*\*\*

Aug 448.44370 27.81761 16.121 < 2e-16 \*\*\*

Sep 456.77049 27.79107 16.436 < 2e-16 \*\*\*

Oct 410.80547 27.77007 14.793 2.24e-16 \*\*\*

Nov 183.54864 27.75602 6.613 1.39e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 39.25 on 34 degrees of freedom

Multiple R-squared: 0.9832, Adjusted R-squared: 0.9768

F-statistic: 153.1 on 13 and 34 DF, p-value: < 2.2e-16

> Add\_sea\_Quad\_pred<-data.frame(predict(Add\_sea\_Quad\_model,interval='predict',newdata=test))

> rmse\_Add\_sea\_Quad<-sqrt(mean((test$Sales-Add\_sea\_Quad\_pred$fit)^2,na.rm=T))

> rmse\_Add\_sea\_Quad

[1] 218.1939

> multi\_sea\_model<-lm(log\_sales~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train)

> summary(multi\_sea\_model)

Call:

lm(formula = log\_sales ~ Jan + Feb + Mar + Apr + May + Jun +

Jul + Aug + Sep + Oct + Nov, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.210538 -0.083185 -0.007374 0.085859 0.223878

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.87367 0.06189 111.070 < 2e-16 \*\*\*

Jan -0.15547 0.08752 -1.776 0.084122 .

Feb -0.24072 0.08752 -2.751 0.009252 \*\*

Mar -0.13990 0.08752 -1.599 0.118663

Apr 0.02883 0.08752 0.329 0.743730

May 0.17203 0.08752 1.966 0.057098 .

Jun 0.26838 0.08752 3.067 0.004094 \*\*

Jul 0.30112 0.08752 3.441 0.001486 \*\*

Aug 0.35865 0.08752 4.098 0.000226 \*\*\*

Sep 0.36963 0.08752 4.223 0.000156 \*\*\*

Oct 0.34059 0.08752 3.892 0.000413 \*\*\*

Nov 0.16661 0.08752 1.904 0.064970 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1238 on 36 degrees of freedom

Multiple R-squared: 0.7916, Adjusted R-squared: 0.728

F-statistic: 12.43 on 11 and 36 DF, p-value: 3.264e-09

> multi\_sea\_pred<-data.frame(predict(multi\_sea\_model,newdata=test,interval='predict'))

> rmse\_multi\_sea<-sqrt(mean((test$Sales-exp(multi\_sea\_pred$fit))^2,na.rm = T))

> rmse\_multi\_sea

[1] 239.6543

> multi\_add\_sea\_model<-lm(log\_sales~t+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train)

> summary(multi\_add\_sea\_model)

Call:

lm(formula = log\_sales ~ t + Jan + Feb + Mar + Apr + May + Jun +

Jul + Aug + Sep + Oct + Nov, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.073269 -0.021672 0.001997 0.020757 0.086610

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.6448891 0.0222743 298.320 < 2e-16 \*\*\*

t 0.0076260 0.0004027 18.939 < 2e-16 \*\*\*

Jan -0.0715828 0.0268339 -2.668 0.01149 \*

Feb -0.1644641 0.0267704 -6.144 5.00e-07 \*\*\*

Mar -0.0712705 0.0267128 -2.668 0.01148 \*

Apr 0.0898412 0.0266612 3.370 0.00184 \*\*

May 0.2254098 0.0266155 8.469 5.42e-10 \*\*\*

Jun 0.3141391 0.0265759 11.820 8.96e-14 \*\*\*

Jul 0.3392452 0.0265423 12.781 9.60e-15 \*\*\*

Aug 0.3891559 0.0265148 14.677 < 2e-16 \*\*\*

Sep 0.3925089 0.0264934 14.815 < 2e-16 \*\*\*

Oct 0.3558417 0.0264781 13.439 2.21e-15 \*\*\*

Nov 0.1742362 0.0264689 6.583 1.33e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03743 on 35 degrees of freedom

Multiple R-squared: 0.9815, Adjusted R-squared: 0.9751

F-statistic: 154.5 on 12 and 35 DF, p-value: < 2.2e-16

> multi\_add\_sea\_pred<-data.frame(predict(multi\_add\_sea\_model,newdata=test,interval='predict'))

> rmse\_multi\_add\_sea<-sqrt(mean((test$Sales-exp(multi\_add\_sea\_pred$fit))^2,na.rm = T))

> rmse\_multi\_add\_sea

[1] 160.6833

> table\_rmse<-data.frame('Model'=c("rmse\_linear","rmse\_expo","rmse\_Quad","rmse\_sea\_add","rmse\_Add\_sea\_Quad","rmse\_multi\_sea","rmse\_multi\_add\_sea"),'RMSE'=c(rmse\_linear,rmse\_expo,rmse\_Quad,rmse\_sea\_add,rmse\_Add\_sea\_Quad,rmse\_multi\_sea,rmse\_multi\_add\_sea))

> View(table\_rmse)

> colnames(table\_rmse)<-c("model","RMSE")

> View(table\_rmse)

| **model** | | **RMSE** | |
| --- | --- | --- | --- |
|  |  | |  |
| **7** | **rmse\_multi\_add\_sea** | | **160.6833** |
| **5** | **rmse\_Add\_sea\_Quad** | | **218.1939** |
| **4** | **rmse\_sea\_add** | | **235.6027** |
| **6** | **rmse\_multi\_sea** | | **239.6543** |
| **1** | **rmse\_linear** | | **260.9378** |
| **2** | **rmse\_expo** | | **268.6938** |
| **3** | **rmse\_Quad** | | **297.4067** |

> # Use entire data : Multiplicative Seasonality Linear trend has least RMSE value

> new\_model <- lm(log\_sales~t+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data=plastic)

> #predict(new\_model,n.ahead=1)

> # Getting residuals

> resid <- residuals(new\_model)

> resid[1:10]

1 2 3 4 5 6 7 8

0.008782926 0.014035295 0.023312129 0.007922054 0.003881908 -0.022900385 0.005183324 0.002372442

9 10

-0.002805141 -0.030775571

> hist(resid)

> acf(resid,lag.max = 10)

> # By principal of parcimony we will consider lag - 1 as we have so

> # many significant lags

> # Building Autoregressive model on residuals consider lag-1

> k <- arima(resid, order=c(1,0,0))

> acf(k$residuals,lag.max = 15)

> pred\_res<- predict(arima(resid,order=c(1,0,0)),n.ahead = 12)

> str(pred\_res)

List of 2

$ pred: Time-Series [1:12] from 61 to 72: -0.1038 -0.0872 -0.0734 -0.0621 -0.0527 ...

$ se : Time-Series [1:12] from 61 to 72: 0.0315 0.0409 0.0463 0.0496 0.0517 ...

> pred\_res$pred

Time Series:

Start = 61

End = 72

Frequency = 1

[1] -0.10377499 -0.08717168 -0.07344411 -0.06209420 -0.05271014 -0.04495142 -0.03853653 -0.03323273

[9] -0.02884756 -0.02522192 -0.02222424 -0.01974578

> acf(k$residuals)

> ####################### Predicting new data #############################

> library(readxl)

> test\_data<-read\_excel(file.choose(),1) #Load Predict\_new\_PlasticData.xlsx

> View(test\_data)

> #test\_data<-Predict\_new

> pred\_new<-data.frame(predict(new\_model,newdata=test\_data,interval = 'predict'))

> View(pred\_new)

> #pred\_re<-pred\_res$pred[1:12]

> pred\_new$fit <- pred\_new$fit+pred\_res$pred[1:12]

> View(pred\_new)

> ####################### Predicting new data #############################

> library(readxl)

> test\_data<-read\_excel(file.choose(),1) #Load Predict\_new\_PlasticData.xlsx

> View(test\_data)

> #test\_data<-Predict\_new

> pred\_new<-data.frame(predict(new\_model,newdata=test\_data,interval = 'predict'))

> View(pred\_new)

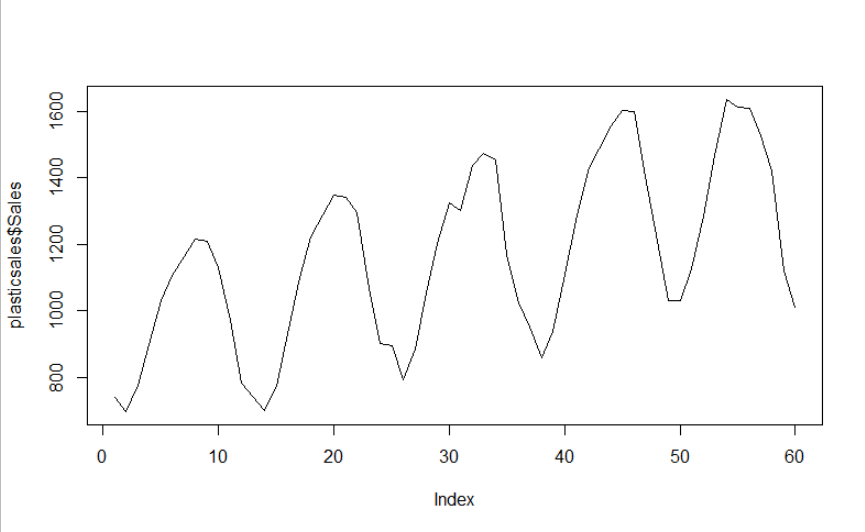
> #pred\_re<-pred\_res$pred[1:12]

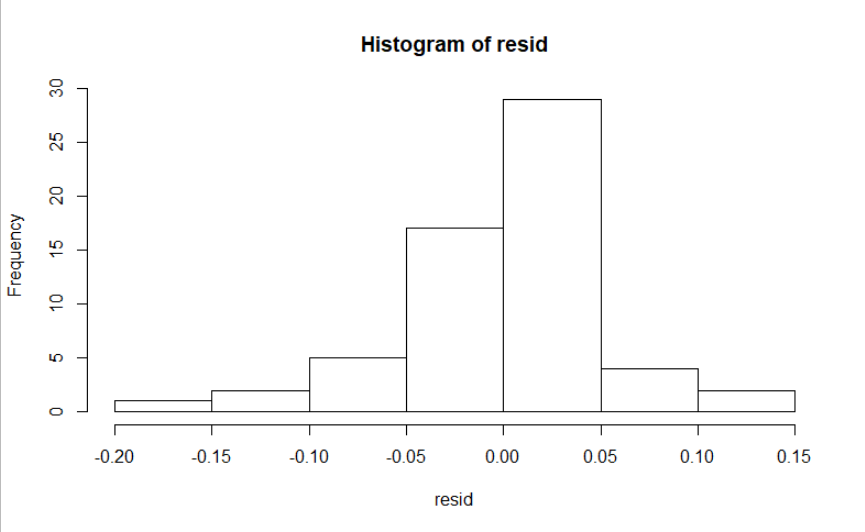
> pred\_new$fit <- pred\_new$fit+pred\_res$pred[1:12]

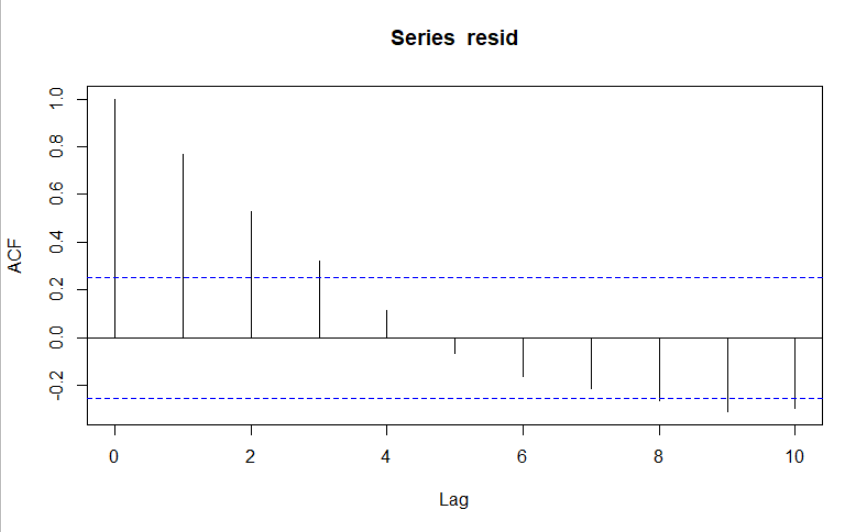
> View(pred\_new)

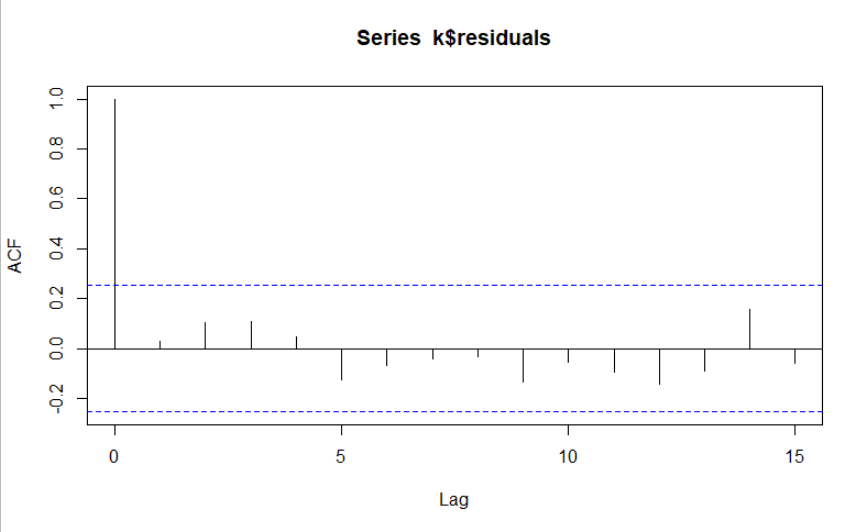
| **fit** | | **lwr** | | **upr** | |
| --- | --- | --- | --- | --- | --- |
|  |  | |  | |  |
| **1** | 6.900435 | | 6.868004 | | 7.140415 |
| **2** | 6.849222 | | 6.800188 | | 7.072599 |
| **3** | 6.961040 | | 6.898278 | | 7.170689 |
| **4** | 7.133797 | | 7.059686 | | 7.332097 |
| **5** | 7.284365 | | 7.200870 | | 7.473281 |
| **6** | 7.391001 | | 7.299747 | | 7.572158 |
| **7** | 7.420400 | | 7.322731 | | 7.595142 |
| **8** | 7.471360 | | 7.368387 | | 7.640799 |
| **9** | 7.474322 | | 7.366964 | | 7.639376 |
| **10** | 7.440055 | | 7.329071 | | 7.601482 |
| **11** | 7.256224 | | 7.142243 | | 7.414654 |
| **12** | 7.105511 | | 6.989051 | | 7.261462 |

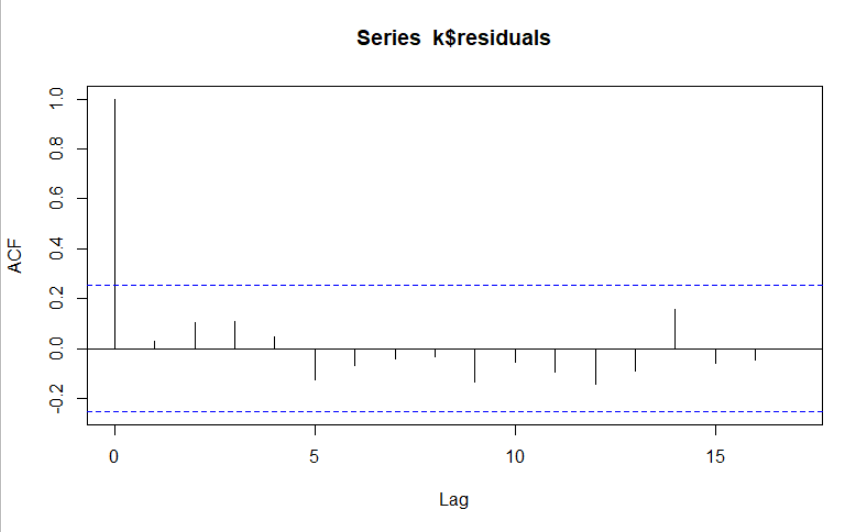
**Plots :**











**Inference :**

The RMSE values of various models are as below :

| **model** | | **RMSE** | |
| --- | --- | --- | --- |
|  |  | |  |
| **7** | **rmse\_multi\_add\_sea** | | **160.6833** |
| **5** | **rmse\_Add\_sea\_Quad** | | **218.1939** |
| **4** | **rmse\_sea\_add** | | **235.6027** |
| **6** | **rmse\_multi\_sea** | | **239.6543** |
| **1** | **rmse\_linear** | | **260.9378** |
| **2** | **rmse\_expo** | | **268.6938** |
| **3** | **rmse\_Quad** | | **297.4067** |

The lowest RMSE value was of Multiplicative Seasonality Linear trend model. So we used this model for forecasting.