**Forecast the Airlines Passengers 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:**

> library(forecast)

> library(timeSeries)

> plastic <- read.csv(file.choose())

> View(plastic)

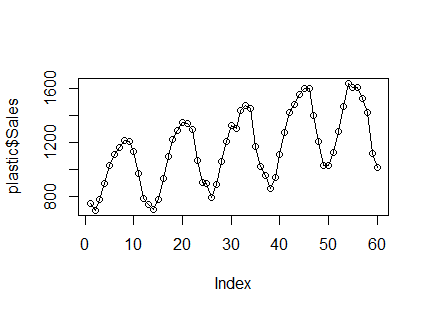
> str(plastic)

'data.frame': 60 obs. of 2 variables:

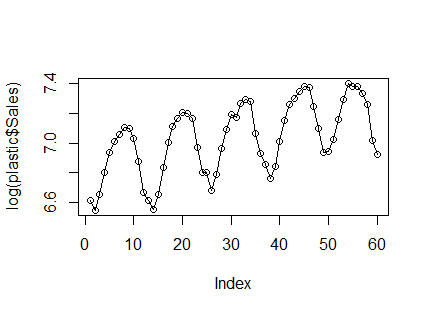
$ Month: Factor w/ 60 levels "Apr-49","Apr-50",..: 21 16 36 1 41 31 26 6 56 51 ...

$ Sales: int 742 697 776 898 1030 1107 1165 1216 1208 1131 ...

> plot(plastic$Sales,type = "o")



> plot(log(plastic$Sales),type = "o")



> summary(plastic)

Month Sales

Apr-49 : 1 Min. : 697.0

Apr-50 : 1 1st Qu.: 947.8

Apr-51 : 1 Median :1148.0

Apr-52 : 1 Mean :1162.4

Apr-53 : 1 3rd Qu.:1362.5

Aug-49 : 1 Max. :1637.0

(Other):54

**Pre-processing the data**

**Creating dummy variables**

> month <- data.frame(outer(rep(month.abb,length=60),month.abb,"==")+0)

> colnames(month) <- month.abb

> View(month)

> plastic2<- cbind(plastic,month)

> plastic2["log1"] <- log(plastic$Sales)

> plastic2["time1"] <- 1:60

> plastic2["time1\_sq"] <- (plastic2$time1)\*(plastic2$time1)

> attach(plastic2)

**Splitting data to test and train**

> train1 <- plastic2[1:54,]

> test1 <- plastic2[54:60,]

1. **Linear model**

> linear\_model <- lm(Sales~time1,data = train1)

> summary(linear\_model)

Call:

lm(formula = Sales ~ time1, data = train1)

Residuals:

Min 1Q Median 3Q Max

-372.84 -206.96 -1.62 207.95 306.15

Coefficients:

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

(Intercept) 886.384 59.721 14.84 < 2e-16 \*\*\*

time1 9.144 1.889 4.84 1.2e-05 \*\*\*

---

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

Residual standard error: 216.4 on 52 degrees of freedom

Multiple R-squared: 0.3105, Adjusted R-squared: 0.2973

F-statistic: 23.42 on 1 and 52 DF, p-value: 1.204e-05

**R^2 = 0.3105**

> linear\_pred <- data.frame(predict(linear\_model,newdata = test1,interval = "predict"))

> linear\_model\_rmse <- sqrt(mean((linear\_pred$fit-test1$Sales)^2,na.rm = T))

> linear\_model\_rmse

[1] 252.3652

**RMSE = 252.3652**

1. **Exp model**

> exp\_model <- lm(log1~time1,data = train1)

> summary(exp\_model)

Call:

lm(formula = log1 ~ time1, data = train1)

Residuals:

Min 1Q Median 3Q Max

-0.34794 -0.18127 0.01598 0.17687 0.25854

Coefficients:

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

(Intercept) 6.783366 0.054469 124.537 < 2e-16 \*\*\*

time1 0.008261 0.001723 4.794 1.41e-05 \*\*\*

---

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

Residual standard error: 0.1974 on 52 degrees of freedom

Multiple R-squared: 0.3065, Adjusted R-squared: 0.2932

F-statistic: 22.98 on 1 and 52 DF, p-value: 1.41e-05

**R^2 = 0.3065**

> exp\_pred <- data.frame(predict(exp\_model,interval = "predict",newdata = test1))

> exp\_model\_rmse <- sqrt(mean((exp\_pred$fit-test1$Sales)^2))

> exp\_model\_rmse

[1] 1431.609

**RMSE = 1431.609**

1. **quadratic model**

> quad\_mod <- lm(Sales~time1+time1\_sq,data = train1)

> summary(quad\_mod)

Call:

lm(formula = Sales ~ time1 + time1\_sq, data = train1)

Residuals:

Min 1Q Median 3Q Max

-375.97 -207.22 -1.99 205.64 307.65

Coefficients:

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

(Intercept) 874.29084 92.58399 9.443 8.76e-13 \*\*\*

time1 10.43928 7.76637 1.344 0.185

time1\_sq -0.02356 0.13688 -0.172 0.864

---

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

Residual standard error: 218.4 on 51 degrees of freedom

Multiple R-squared: 0.3109, Adjusted R-squared: 0.2839

F-statistic: 11.51 on 2 and 51 DF, p-value: 7.507e-05

**R^2 = 0.3109**

> quad\_pred <- data.frame(predict(quad\_mod,interval = "predict",newdata = test1))

> quad\_rmse <- sqrt(mean((test1$Sales-quad\_pred$fit)^2,na.rm = T))

> quad\_rmse

[1] 250.891

**RMSE = 250.891**

1. **Additive seasonality**

> add\_seas <- lm(Sales~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train1)

> summary(add\_seas)

Call:

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

Aug + Sep + Oct + Nov, data = train1)

Residuals:

Min 1Q Median 3Q Max

-239.0 -122.8 -12.9 79.0 294.0

Coefficients:

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

(Intercept) 979.00 81.89 11.955 4.19e-15 \*\*\*

Jan -107.00 109.87 -0.974 0.335694

Feb -162.40 109.87 -1.478 0.146840

Mar -79.20 109.87 -0.721 0.474995

Apr 76.80 109.87 0.699 0.488402

May 236.00 109.87 2.148 0.037529 \*

Jun 364.00 109.87 3.313 0.001906 \*\*

Jul 332.00 115.81 2.867 0.006458 \*\*

Aug 410.00 115.81 3.540 0.000993 \*\*\*

Sep 427.50 115.81 3.691 0.000637 \*\*\*

Oct 391.00 115.81 3.376 0.001593 \*\*

Nov 173.50 115.81 1.498 0.141587

---

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

Residual standard error: 163.8 on 42 degrees of freedom

Multiple R-squared: 0.681, Adjusted R-squared: 0.5974

F-statistic: 8.15 on 11 and 42 DF, p-value: 2.193e-07

**R^2 = 0.1492**

> add\_seas\_pred <- data.frame(predict(add\_seas,interval = "predict",newdata = test1))

> add\_seas\_rmse <- sqrt(mean((test1$Sales-add\_seas\_pred$fit)^2,na.rm = T))

> add\_seas\_rmse

[1] 186.6768

**RMSE = 134.3448**

1. **additive seasonality with linear**

> add\_seast <- lm(Sales~time1+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train1)

> summary(add\_seast)

Call:

lm(formula = Sales ~ time1 + Jan + Feb + Mar + Apr + May + Jun +

Jul + Aug + Sep + Oct + Nov, data = train1)

Residuals:

Min 1Q Median 3Q Max

-78.60 -24.42 -10.33 21.83 94.27

Coefficients:

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

(Intercept) 711.6667 27.5684 25.815 < 2e-16 \*\*\*

time1 8.9111 0.4285 20.798 < 2e-16 \*\*\*

Jan -62.4444 32.7906 -1.904 0.063899 .

Feb -126.7556 32.7654 -3.869 0.000384 \*\*\*

Mar -52.4667 32.7457 -1.602 0.116779

Apr 94.6222 32.7317 2.891 0.006119 \*\*

May 244.9111 32.7233 7.484 3.44e-09 \*\*\*

Jun 364.0000 32.7205 11.125 5.87e-14 \*\*\*

Jul 376.5556 34.5569 10.897 1.11e-13 \*\*\*

Aug 445.6444 34.5330 12.905 5.01e-16 \*\*\*

Sep 454.2333 34.5144 13.161 2.61e-16 \*\*\*

Oct 408.8222 34.5011 11.850 8.04e-15 \*\*\*

Nov 182.4111 34.4931 5.288 4.42e-06 \*\*\*

---

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

Residual standard error: 48.78 on 41 degrees of freedom

Multiple R-squared: 0.9724, Adjusted R-squared: 0.9643

F-statistic: 120.3 on 12 and 41 DF, p-value: < 2.2e-16

**R^2 = 0.9724**

> add\_seast\_pred <- data.frame(predict(add\_seast,interval = "predict",newdata = test1))

> add\_seast\_rmse <- sqrt(mean((test1$Sales-add\_seast\_pred$fit)^2,na.rm = T))

> add\_seast\_rmse

[1] 178.5941

**RMSE = 178.5941**

1. **additive seasonality with quadratic**

> add\_seasq <- lm(Sales~time1+time1\_sq+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train1)

> summary(add\_seasq)

Call:

lm(formula = Sales ~ time1 + time1\_sq + Jan + Feb + Mar + Apr +

May + Jun + Jul + Aug + Sep + Oct + Nov, data = train1)

Residuals:

Min 1Q Median 3Q Max

-93.315 -21.215 2.149 24.585 66.977

Coefficients:

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

(Intercept) 778.23304 27.18956 28.622 < 2e-16 \*\*\*

time1 2.48804 1.47634 1.685 0.09972 .

time1\_sq 0.11678 0.02606 4.481 6.08e-05 \*\*\*

Jan -75.05702 27.23315 -2.756 0.00877 \*\*

Feb -138.90100 27.20186 -5.106 8.43e-06 \*\*\*

Mar -64.37854 27.18059 -2.369 0.02278 \*

Apr 82.71034 27.16907 3.044 0.00411 \*\*

May 232.76567 27.16730 8.568 1.37e-10 \*\*\*

Jun 351.38742 27.17559 12.930 7.18e-16 \*\*\*

Jul 376.55556 28.54646 13.191 3.75e-16 \*\*\*

Aug 446.11158 28.52690 15.638 < 2e-16 \*\*\*

Sep 454.93403 28.51176 15.956 < 2e-16 \*\*\*

Oct 409.52292 28.50077 14.369 < 2e-16 \*\*\*

Nov 182.87824 28.49394 6.418 1.22e-07 \*\*\*

---

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

Residual standard error: 40.29 on 40 degrees of freedom

Multiple R-squared: 0.9816, Adjusted R-squared: 0.9756

F-statistic: 164.2 on 13 and 40 DF, p-value: < 2.2e-16

**R^2 = 0.9816**

> add\_seasq\_pred <- data.frame(predict(add\_seasq,interval = "predict",newdata = test1))

> add\_seasq\_rmse <- sqrt(mean((test1$Sales-add\_seasq\_pred$fit)^2,na.rm = T))

> add\_seasq\_rmse

[1] 247.8046

**RMSE = 247.8046**

1. **Multiplicative seasonality**

> mul\_seas\_model <- lm(log1~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train1)

> summary(mul\_seas\_model)

Call:

lm(formula = log1 ~ Jan + Feb + Mar + Apr + May + Jun + Jul +

Aug + Sep + Oct + Nov, data = train1)

Residuals:

Min 1Q Median 3Q Max

-0.210538 -0.125843 -0.003103 0.066857 0.245046

Coefficients:

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

(Intercept) 6.87367 0.07270 94.555 < 2e-16 \*\*\*

Jan -0.11165 0.09753 -1.145 0.258801

Feb -0.17946 0.09753 -1.840 0.072832 .

Mar -0.08137 0.09753 -0.834 0.408816

Apr 0.08003 0.09753 0.821 0.416500

May 0.22122 0.09753 2.268 0.028520 \*

Jun 0.32010 0.09753 3.282 0.002080 \*\*

Jul 0.30112 0.10281 2.929 0.005476 \*\*

Aug 0.35865 0.10281 3.489 0.001153 \*\*

Sep 0.36963 0.10281 3.595 0.000845 \*\*\*

Oct 0.34059 0.10281 3.313 0.001906 \*\*

Nov 0.16661 0.10281 1.621 0.112584

---

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

Residual standard error: 0.1454 on 42 degrees of freedom

Multiple R-squared: 0.696, Adjusted R-squared: 0.6164

F-statistic: 8.742 on 11 and 42 DF, p-value: 8.715e-08

**R^2 = 0.696**

> mul\_seas\_pred <- data.frame(predict(mul\_seas\_model,interval = 'predict',newdata = test1))

> mul\_seas\_rmse <- sqrt(mean((test1$Sales-mul\_seas\_pred$fit)^2,na.rm = T))

> mul\_seas\_rmse

[1] 1431.702

**RMSE = 1431.702**

1. **multiplicative seasonality with linear**

> mul\_seast\_model <- lm(log1~time1+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train1)

> summary(mul\_seast\_model)

Call:

lm(formula = log1 ~ time1 + Jan + Feb + Mar + Apr + May + Jun +

Jul + Aug + Sep + Oct + Nov, data = train1)

Residuals:

Min 1Q Median 3Q Max

-0.066405 -0.021732 0.000112 0.019295 0.079745

Coefficients:

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

(Intercept) 6.633449 0.021043 315.235 < 2e-16 \*\*\*

time1 0.008007 0.000327 24.484 < 2e-16 \*\*\*

Jan -0.071610 0.025029 -2.861 0.006616 \*\*

Feb -0.147433 0.025010 -5.895 6.10e-07 \*\*\*

Mar -0.057350 0.024995 -2.295 0.026955 \*

Apr 0.096050 0.024984 3.844 0.000413 \*\*\*

May 0.229226 0.024978 9.177 1.71e-11 \*\*\*

Jun 0.320096 0.024976 12.816 6.29e-16 \*\*\*

Jul 0.341152 0.026377 12.934 4.66e-16 \*\*\*

Aug 0.390681 0.026359 14.822 < 2e-16 \*\*\*

Sep 0.393653 0.026345 14.942 < 2e-16 \*\*\*

Oct 0.356604 0.026335 13.541 < 2e-16 \*\*\*

Nov 0.174618 0.026329 6.632 5.46e-08 \*\*\*

---

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

Residual standard error: 0.03723 on 41 degrees of freedom

Multiple R-squared: 0.9805, Adjusted R-squared: 0.9748

F-statistic: 172.2 on 12 and 41 DF, p-value: < 2.2e-16

**R^2 = 0.9805**

> mul\_seast\_pred <- data.frame(predict(mul\_seast\_model,interval = 'predict',newdata = test1))

> mul\_seast\_rmse <- sqrt(mean((test1$Sales-mul\_seast\_pred$fit)^2,na.rm = T))

> mul\_seast\_rmse

[1] 1431.473

**RMSE = 1431.473**

> table\_formate <- data.frame(c("linear\_model\_rmse","exp\_model\_rmse","quad\_rmse","add\_seas\_rmse","add\_seasq\_rmse","add\_seast\_rmse","mul\_seas\_rmse","mul\_seast\_rmse"),c(linear\_model\_rmse,exp\_model\_rmse,quad\_rmse,add\_seas\_rmse,add\_seasq\_rmse,add\_seast\_rmse,mul\_seas\_rmse,mul\_seast\_rmse))

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

> View(table\_formate)

> table\_formate

model RMSE

1 linear\_model\_rmse 252.3652

2 exp\_model\_rmse 1431.6090

3 quad\_rmse 250.8910

4 add\_seas\_rmse 186.6768

5 add\_seasq\_rmse 247.8046

6 add\_seast\_rmse 178.5941

7 mul\_seas\_rmse 1431.7018

8 mul\_seast\_rmse 1431.4726

**Additive seasonality with linear has less RMSE , So choosing it for forecasting**

**Final model**

> finalmodel <- lm(Sales~time1+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = plastic2)

> finalmodel

Call:

lm(formula = Sales ~ time1 + Jan + Feb + Mar + Apr + May + Jun +

Jul + Aug + Sep + Oct + Nov, data = plastic2)

Coefficients:

(Intercept) time1 Jan Feb Mar Apr

710.650 7.643 -29.726 -92.769 -17.212 131.144

May Jun Jul Aug Sep Oct

282.701 403.058 423.415 477.572 467.929 409.486

Nov

167.643

> summary(finalmodel)

Call:

lm(formula = Sales ~ time1 + Jan + Feb + Mar + Apr + May + Jun +

Jul + Aug + Sep + Oct + Nov, data = plastic2)

Residuals:

Min 1Q Median 3Q Max

-210.233 -32.242 -1.183 33.275 165.483

Coefficients:

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

(Intercept) 710.6500 37.7255 18.837 < 2e-16 \*\*\*

time1 7.6431 0.5473 13.966 < 2e-16 \*\*\*

Jan -29.7264 45.8952 -0.648 0.520330

Feb -92.7694 45.8266 -2.024 0.048637 \*

Mar -17.2125 45.7645 -0.376 0.708527

Apr 131.1444 45.7088 2.869 0.006148 \*\*

May 282.7014 45.6597 6.191 1.38e-07 \*\*\*

Jun 403.0583 45.6170 8.836 1.49e-11 \*\*\*

Jul 423.4153 45.5809 9.289 3.27e-12 \*\*\*

Aug 477.5722 45.5513 10.484 6.81e-14 \*\*\*

Sep 467.9292 45.5283 10.278 1.31e-13 \*\*\*

Oct 409.4861 45.5119 8.997 8.64e-12 \*\*\*

Nov 167.6431 45.5020 3.684 0.000592 \*\*\*

---

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

Residual standard error: 71.94 on 47 degrees of freedom

Multiple R-squared: 0.9419, Adjusted R-squared: 0.9271

F-statistic: 63.52 on 12 and 47 DF, p-value: < 2.2e-16

**Auto.arima method**

> library(tseries)

> plastic\_ts <- as.ts(plastic$Sales)

> plastic\_ts <- ts(plastic\_ts,start = c(1949,1),end = c(1953,12),frequency = 12)

> class(plastic\_ts)

[1] "ts"

> start(plastic\_ts)

[1] 1949 1

> end(plastic\_ts)

[1] 1953 12

> sum(is.na(plastic\_ts))

[1] 0

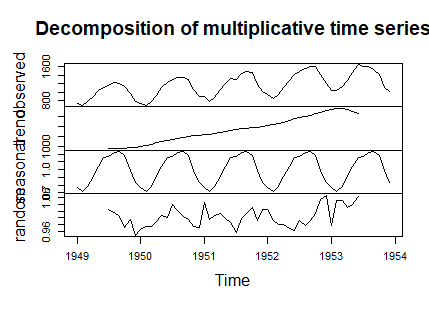
> summary(plastic\_ts)

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

697.0 947.8 1148.0 1162.4 1362.5 1637.0

> decompdata1 <- decompose(plastic\_ts,"multiplicative")

> plot(decompdata1)



> cycle(plastic\_ts)

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

1949 1 2 3 4 5 6 7 8 9 10 11 12

1950 1 2 3 4 5 6 7 8 9 10 11 12

1951 1 2 3 4 5 6 7 8 9 10 11 12

1952 1 2 3 4 5 6 7 8 9 10 11 12

1953 1 2 3 4 5 6 7 8 9 10 11 12

**Model Building**

> newmodel1 <- auto.arima(plastic\_ts,ic = "aic",trace = T)

ARIMA(2,0,2)(1,1,1)[12] with drift : Inf

ARIMA(0,0,0)(0,1,0)[12] with drift : 582.5789

ARIMA(1,0,0)(1,1,0)[12] with drift : 525.6583

ARIMA(0,0,1)(0,1,1)[12] with drift : Inf

ARIMA(0,0,0)(0,1,0)[12] : 607.2886

ARIMA(1,0,0)(0,1,0)[12] with drift : 527.2309

ARIMA(1,0,0)(1,1,1)[12] with drift : Inf

ARIMA(1,0,0)(0,1,1)[12] with drift : 522.292

ARIMA(0,0,0)(0,1,1)[12] with drift : Inf

ARIMA(2,0,0)(0,1,1)[12] with drift : 522.8511

ARIMA(1,0,0)(1,1,1)[12] : Inf

ARIMA(1,0,0)(1,1,0)[12] : 524.4439

ARIMA(0,0,0)(0,1,1)[12] : 606.2486

ARIMA(2,0,0)(0,1,1)[12] : 523.7636

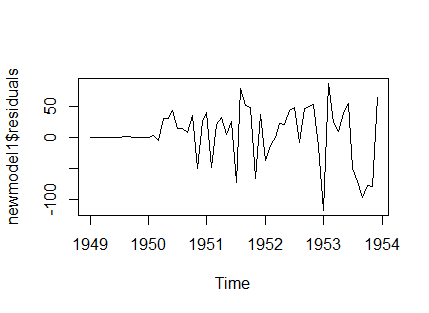
ARIMA(1,0,1)(0,1,1)[12] : 523.7954

ARIMA(0,0,1)(0,1,1)[12] : 567.8857

ARIMA(2,0,1)(0,1,1)[12] : 525.7218

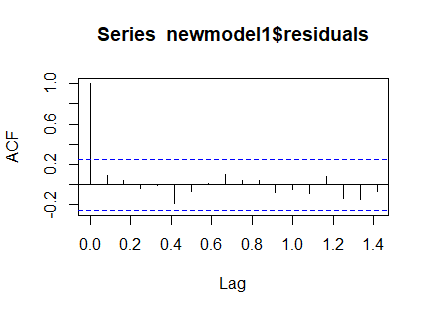
Best model: ARIMA(1,0,0)(0,1,1)[12]

> plot.ts(newmodel1$residuals)

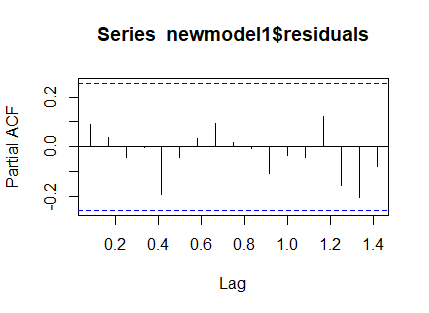


**Verifying p,d,q values using acf and pacf**

> acf(newmodel1$residuals) #q=0



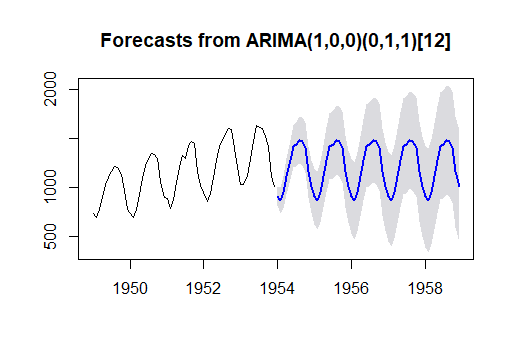
> pacf(newmodel1$residuals) #p=1



**Forecasting the model**

> forecasting1 <- forecast(newmodel1,level = c(95),h=5\*12)

> plot(forecasting1)



**Model Testing**

> Box.test(newmodel1$residuals,lag = 5,type ="Ljung-Box" )

Box-Ljung test

data: newmodel1$residuals

X-squared = 3.1709, df = 5, p-value = 0.6737

> Box.test(newmodel1$residuals,lag = 2,type = "Ljung-Box")

Box-Ljung test

data: newmodel1$residuals

X-squared = 0.62243, df = 2, p-value = 0.7326

**p values are smaller**