Title: 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

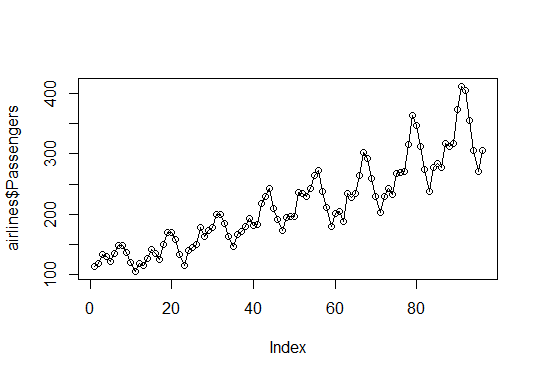
Forecasting.

Ans::

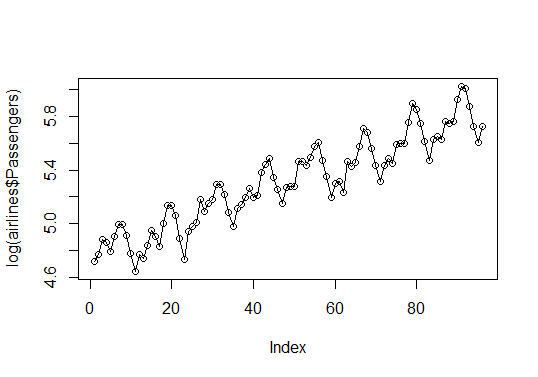
airlines <- readxl::read\_xlsx(file.choose())

> View(airlines)

> plot(airlines$Passengers,type = "o")



> plot(log(airlines$Passengers),type = "o")



> summary(airlines)

Month Passengers

Min. :1995-01-01 00:00:00 Min. :104.0

1st Qu.:1996-12-24 06:00:00 1st Qu.:156.0

Median :1998-12-16 12:00:00 Median :200.0

Mean :1998-12-16 05:00:00 Mean :213.7

3rd Qu.:2000-12-08 18:00:00 3rd Qu.:264.8

Max. :2002-12-01 00:00:00 Max. :413.0

> class(airlines)

[1] "tbl\_df" "tbl" "data.frame"

> str(airlines)

tibble [96 x 2] (S3: tbl\_df/tbl/data.frame)

$ Month : POSIXct[1:96], format: "1995-01-01" "1995-02-01" ...

$ Passengers: num [1:96] 112 118 132 129 121 135 148 148 136 119 ...

> airlines$Month <- as.Date(airlines$Month)

> str(airlines)

tibble [96 x 2] (S3: tbl\_df/tbl/data.frame)

$ Month : Date[1:96], format: "1995-01-01" "1995-02-01" ...

$ Passengers: num [1:96] 112 118 132 129 121 135 148 148 136 119 ...

> ######################## pre processing the data ###############################

>

> #creating dummy variables

>

> month <- data.frame(outer(rep(month.abb,length=96),month.abb,"==")+0)

> colnames(month) <- month.abb

> View(month)

> airlines2 <- cbind(airlines,month)

> airlines2["log2"] <- log(airlines2$Passengers)

> airlines2["time"] <- 1:96

> airlines2["time\_sq"] <- (airlines2$time)\*(airlines2$time)

> attach(airlines2)

> #splitting data to test and train

> train <- airlines2[1:85,]

> test <- airlines2[86:96,]

#\*\*\*\* linear model\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

>

> linear\_model <- lm(Passengers~time,data = train)

> summary(linear\_model)

Call:

lm(formula = Passengers ~ time, data = train)

Residuals:

Min 1Q Median 3Q Max

-55.265 -17.111 -1.101 16.431 88.620

Coefficients:

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

(Intercept) 106.3748 5.8584 18.16 <2e-16 \*\*\*

time 2.1393 0.1183 18.08 <2e-16 \*\*\*

---

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

Residual standard error: 26.77 on 83 degrees of freedom

Multiple R-squared: 0.7975, Adjusted R-squared: 0.795

F-statistic: 326.8 on 1 and 83 DF, p-value: < 2.2e-16

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

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

> linear\_model\_rmse

[1] 55.67417

> #\*\*\*\*\*\*\*\*\* Exp model\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

>

> exp\_model <- lm(log2~time,data = train)

> summary(exp\_model) #R^2 = 0.8273

Call:

lm(formula = log2 ~ time, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.28935 -0.07104 -0.01469 0.07899 0.25367

Coefficients:

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

(Intercept) 4.7717595 0.0274010 174.15 <2e-16 \*\*\*

time 0.0110345 0.0005535 19.94 <2e-16 \*\*\*

---

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

Residual standard error: 0.1252 on 83 degrees of freedom

Multiple R-squared: 0.8273, Adjusted R-squared: 0.8252

F-statistic: 397.5 on 1 and 83 DF, p-value: < 2.2e-16

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

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

> exp\_model\_rmse

[1] 329.6918

#############quadratic model ##############

> quad\_mod <- lm(Passengers~time+time\_sq,data = train)

> summary(quad\_mod) #0.8009

Call:

lm(formula = Passengers ~ time + time\_sq, data = train)

Residuals:

Min 1Q Median 3Q Max

-56.425 -14.952 -3.919 16.069 84.198

Coefficients:

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

(Intercept) 1.143e+02 8.898e+00 12.849 < 2e-16 \*\*\*

time 1.591e+00 4.775e-01 3.332 0.00129 \*\*

time\_sq 6.373e-03 5.380e-03 1.184 0.23966

---

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

Residual standard error: 26.7 on 82 degrees of freedom

Multiple R-squared: 0.8009, Adjusted R-squared: 0.796

F-statistic: 164.9 on 2 and 82 DF, p-value: < 2.2e-16

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

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

> quad\_rmse #50.65955

[1] 50.65955

#additive seasonality #

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

> summary(add\_seas) #0.1492

Call:

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

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

Residuals:

Min 1Q Median 3Q Max

-91.571 -51.143 2.571 38.571 124.429

Coefficients:

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

(Intercept) 189.429 22.112 8.567 1.21e-12 \*\*\*

Jan -5.804 30.278 -0.192 0.849

Feb -19.286 31.271 -0.617 0.539

Mar 8.000 31.271 0.256 0.799

Apr 1.857 31.271 0.059 0.953

May 1.143 31.271 0.037 0.971

Jun 25.143 31.271 0.804 0.424

Jul 50.143 31.271 1.604 0.113

Aug 49.286 31.271 1.576 0.119

Sep 24.143 31.271 0.772 0.443

Oct -1.000 31.271 -0.032 0.975

Nov -24.286 31.271 -0.777 0.440

---

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

Residual standard error: 58.5 on 73 degrees of freedom

Multiple R-squared: 0.1492, Adjusted R-squared: 0.02099

F-statistic: 1.164 on 11 and 73 DF, p-value: 0.3271

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

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

> add\_seas\_rmse

[1] 134.3448

> #additive seasonality with linear #

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

> summary(add\_seast) #0.9559

Call:

lm(formula = Passengers ~ time + Jan + Feb + Mar + Apr + May +

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

Residuals:

Min 1Q Median 3Q Max

-34.174 -8.603 -0.413 7.270 46.381

Coefficients:

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

(Intercept) 85.36571 5.82568 14.653 < 2e-16 \*\*\*

time 2.16798 0.05976 36.279 < 2e-16 \*\*\*

Jan 5.03631 6.94964 0.725 0.470993

Feb 2.39405 7.19577 0.333 0.740326

Mar 27.51179 7.19106 3.826 0.000275 \*\*\*

Apr 19.20095 7.18684 2.672 0.009327 \*\*

May 16.31869 7.18311 2.272 0.026087 \*

Jun 38.15071 7.17988 5.314 1.15e-06 \*\*\*

Jul 60.98274 7.17714 8.497 1.81e-12 \*\*\*

Aug 57.95762 7.17490 8.078 1.10e-11 \*\*\*

Sep 30.64679 7.17316 4.272 5.83e-05 \*\*\*

Oct 3.33595 7.17191 0.465 0.643234

Nov -22.11774 7.17117 -3.084 0.002893 \*\*

---

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

Residual standard error: 13.42 on 72 degrees of freedom

Multiple R-squared: 0.9559, Adjusted R-squared: 0.9485

F-statistic: 130 on 12 and 72 DF, p-value: < 2.2e-16

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

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

> add\_seast\_rmse #36.42285

[1] 36.42285

#additive seasonality with quadratic #

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

> summary(add\_seasq) #0.9609

Call:

lm(formula = Passengers ~ time + time\_sq + Jan + Feb + Mar +

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

Residuals:

Min 1Q Median 3Q Max

-32.317 -7.779 0.385 7.744 40.808

Coefficients:

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

(Intercept) 95.025556 6.392071 14.866 < 2e-16 \*\*\*

time 1.502314 0.228590 6.572 7.08e-09 \*\*\*

time\_sq 0.007740 0.002575 3.006 0.003659 \*\*

Jan 3.836570 6.603618 0.581 0.563094

Feb 2.394048 6.824999 0.351 0.726795

Mar 27.581448 6.820565 4.044 0.000132 \*\*\*

Apr 19.324797 6.816646 2.835 0.005965 \*\*

May 16.481236 6.813200 2.419 0.018128 \*

Jun 38.336480 6.810200 5.629 3.37e-07 \*\*\*

Jul 61.176245 6.807629 8.986 2.49e-13 \*\*\*

Aug 58.143385 6.805482 8.544 1.64e-12 \*\*\*

Sep 30.809331 6.803763 4.528 2.34e-05 \*\*\*

Oct 3.459797 6.802493 0.509 0.612604

Nov -22.048076 6.801699 -3.242 0.001812 \*\*

---

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

Residual standard error: 12.72 on 71 degrees of freedom

Multiple R-squared: 0.9609, Adjusted R-squared: 0.9537

F-statistic: 134.1 on 13 and 71 DF, p-value: < 2.2e-16

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

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

> add\_seasq\_rmse #27.41271

[1] 27.41271

> # multiplicative seasonality #

> mul\_seas\_model <- lm(log2~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train)

> summary(mul\_seas\_model) #0.1376

Call:

lm(formula = log2 ~ Jan + Feb + Mar + Apr + May + Jun + Jul +

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

Residuals:

Min 1Q Median 3Q Max

-0.44600 -0.27460 0.05112 0.22560 0.48448

Coefficients:

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

(Intercept) 5.208117 0.112741 46.195 <2e-16 \*\*\*

Jan -0.043620 0.154378 -0.283 0.778

Feb -0.097238 0.159440 -0.610 0.544

Mar 0.047126 0.159440 0.296 0.768

Apr 0.011394 0.159440 0.071 0.943

May 0.001676 0.159440 0.011 0.992

Jun 0.120037 0.159440 0.753 0.454

Jul 0.227301 0.159440 1.426 0.158

Aug 0.227676 0.159440 1.428 0.158

Sep 0.120513 0.159440 0.756 0.452

Oct -0.006967 0.159440 -0.044 0.965

Nov -0.138701 0.159440 -0.870 0.387

---

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

Residual standard error: 0.2983 on 73 degrees of freedom

Multiple R-squared: 0.1376, Adjusted R-squared: 0.007655

F-statistic: 1.059 on 11 and 73 DF, p-value: 0.4059

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

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

> mul\_seas\_rmse #330.1927

[1] 330.1927

>

> #multiplicative seasonality with linaer #

> mul\_seast\_model <- lm(log2~time+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = train)

> summary(mul\_seast\_model) #0.9768

Call:

lm(formula = log2 ~ time + Jan + Feb + Mar + Apr + May + Jun +

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

Residuals:

Min 1Q Median 3Q Max

-0.143201 -0.030855 0.002289 0.030705 0.105692

Coefficients:

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

(Intercept) 4.6705892 0.0213872 218.383 < 2e-16 \*\*\*

time 0.0111985 0.0002194 51.046 < 2e-16 \*\*\*

Jan 0.0123721 0.0255134 0.485 0.629202

Feb 0.0147468 0.0264171 0.558 0.578418

Mar 0.1479123 0.0263997 5.603 3.64e-07 \*\*\*

Apr 0.1009823 0.0263842 3.827 0.000274 \*\*\*

May 0.0800651 0.0263706 3.036 0.003335 \*\*

Jun 0.1872284 0.0263587 7.103 7.13e-10 \*\*\*

Jul 0.2832931 0.0263487 10.752 < 2e-16 \*\*\*

Aug 0.2724700 0.0263404 10.344 6.93e-16 \*\*\*

Sep 0.1541084 0.0263340 5.852 1.33e-07 \*\*\*

Oct 0.0154298 0.0263295 0.586 0.559689

Nov -0.1275026 0.0263267 -4.843 7.10e-06 \*\*\*

---

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

Residual standard error: 0.04925 on 72 degrees of freedom

Multiple R-squared: 0.9768, Adjusted R-squared: 0.9729

F-statistic: 252.7 on 12 and 72 DF, p-value: < 2.2e-16

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

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

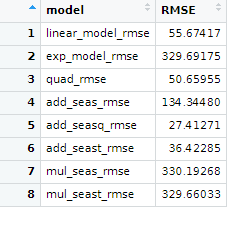
> mul\_seast\_rmse #329.6603

[1] 329.6603

> > 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)



#additive seasonality with quadratic has less RMSE and higher R^2 value

> #choosing it for forecasting

>

> # final model #

> finalmodel <- lm(Passengers~time+time\_sq+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,data = airlines2)

> finalmodel

Call:

lm(formula = Passengers ~ time + time\_sq + Jan + Feb + Mar +

Apr + May + Jun + Jul + Aug + Sep + Oct + Nov, data = airlines2)

Coefficients:

(Intercept) time time\_sq Jan Feb

95.09654 1.23946 0.01143 5.45536 3.09645

Mar Apr May Jun Jul

29.71468 21.56005 19.25755 44.93220 69.33399

Aug Sep Oct Nov

65.21291 34.56897 4.02718 -23.16248

> summary(finalmodel)

Call:

lm(formula = Passengers ~ time + time\_sq + Jan + Feb + Mar +

Apr + May + Jun + Jul + Aug + Sep + Oct + Nov, data = airlines2)

Residuals:

Min 1Q Median 3Q Max

-30.978 -10.111 -0.145 8.304 41.123

Coefficients:

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

(Intercept) 95.096544 6.833399 13.916 < 2e-16 \*\*\*

time 1.239459 0.217218 5.706 1.78e-07 \*\*\*

time\_sq 0.011430 0.002169 5.271 1.08e-06 \*\*\*

Jan 5.455357 7.319357 0.745 0.45820

Feb 3.096448 7.315181 0.423 0.67319

Mar 29.714677 7.311442 4.064 0.00011 \*\*\*

Apr 21.560046 7.308113 2.950 0.00414 \*\*

May 19.257553 7.305169 2.636 0.01003 \*

Jun 44.932200 7.302596 6.153 2.66e-08 \*\*\*

Jul 69.333986 7.300387 9.497 7.39e-15 \*\*\*

Aug 65.212910 7.298541 8.935 9.69e-14 \*\*\*

Sep 34.568974 7.297067 4.737 8.97e-06 \*\*\*

Oct 4.027177 7.295981 0.552 0.58247

Nov -23.162481 7.295306 -3.175 0.00211 \*\*

---

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

Residual standard error: 14.59 on 82 degrees of freedom

Multiple R-squared: 0.9645, Adjusted R-squared: 0.9588

F-statistic: 171.2 on 13 and 82 DF, p-value: < 2.2e-16

# auto.arima method #

>

> library(tseries)

‘tseries’ version: 0.10-48

‘tseries’ is a package for time series analysis and

computational finance.

See ‘library(help="tseries")’ for details.

Warning message:

package ‘tseries’ was built under R version 4.0.3

> airlines\_ts <- as.ts(airlines$Passengers)

> airlines\_ts <- ts(airlines\_ts,start = c(1995,1),end = c(2002,12),frequency = 12)

> class(airlines\_ts)

[1] "ts"

> start(airlines\_ts)

[1] 1995 1

> end(airlines\_ts)

[1] 2002 12

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

[1] 0

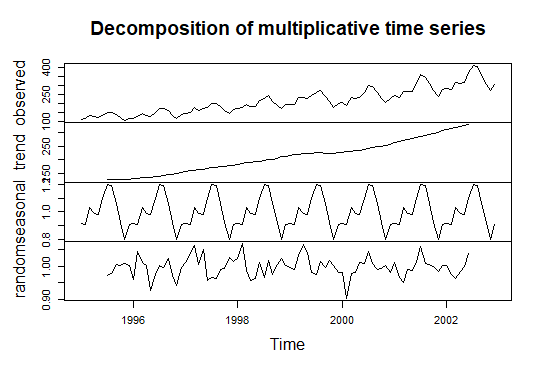
> summary(airlines\_ts)

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

104.0 156.0 200.0 213.7 264.8 413.0

decompdata <- decompose(airlines\_ts,"multiplicative")

> plot(decompdata)



> cycle(airlines\_ts)

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

1995 1 2 3 4 5 6 7 8 9 10 11 12

1996 1 2 3 4 5 6 7 8 9 10 11 12

1997 1 2 3 4 5 6 7 8 9 10 11 12

1998 1 2 3 4 5 6 7 8 9 10 11 12

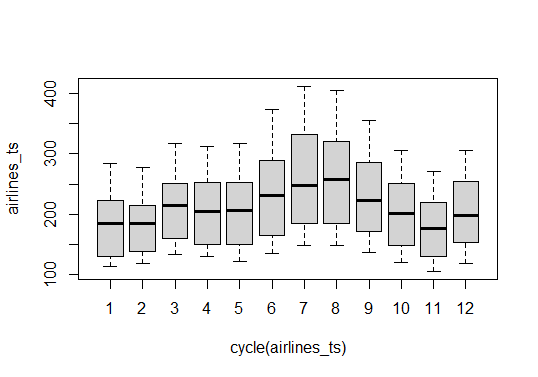
1999 1 2 3 4 5 6 7 8 9 10 11 12

2000 1 2 3 4 5 6 7 8 9 10 11 12

2001 1 2 3 4 5 6 7 8 9 10 11 12

2002 1 2 3 4 5 6 7 8 9 10 11 12

> boxplot(airlines\_ts~cycle(airlines\_ts))



>

> #################### model ######################

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

ARIMA(2,1,2)(1,1,1)[12] : 619.8465

ARIMA(0,1,0)(0,1,0)[12] : 622.0124

ARIMA(1,1,0)(1,1,0)[12] : 615.9655

ARIMA(0,1,1)(0,1,1)[12] : 616.6286

ARIMA(1,1,0)(0,1,0)[12] : 618.1919

ARIMA(1,1,0)(2,1,0)[12] : 617.8138

ARIMA(1,1,0)(1,1,1)[12] : 617.8815

ARIMA(1,1,0)(0,1,1)[12] : 616.5315

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

ARIMA(0,1,0)(1,1,0)[12] : 618.2212

ARIMA(2,1,0)(1,1,0)[12] : 617.8626

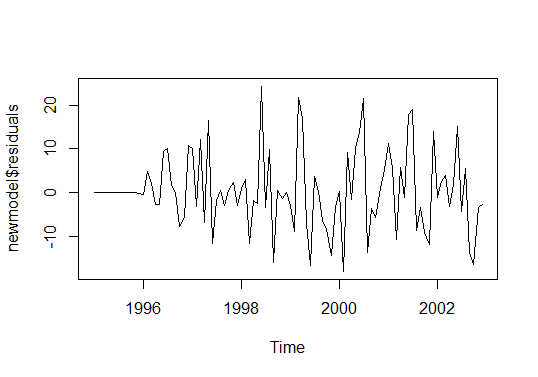
ARIMA(1,1,1)(1,1,0)[12] : 617.4616

ARIMA(0,1,1)(1,1,0)[12] : 616.0758

ARIMA(2,1,1)(1,1,0)[12] : 619.3943

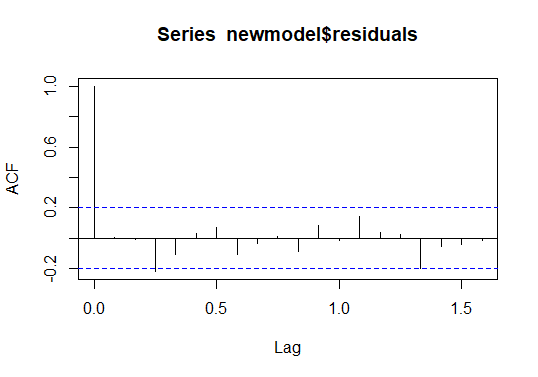
Best model: ARIMA(1,1,0)(1,1,0)[12]

> plot.ts(newmodel$residuals)

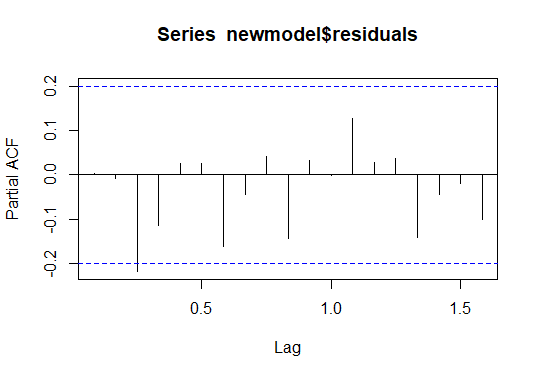


> # verifying p,d,q values using acf and pacf

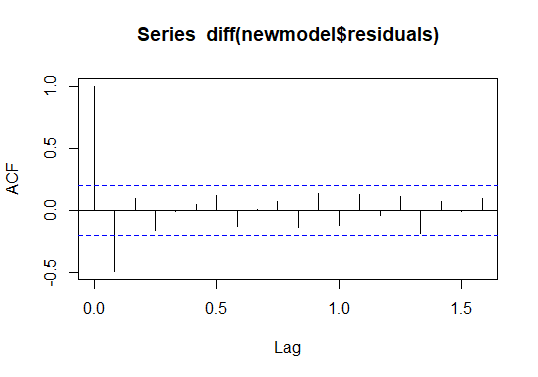
> acf(newmodel$residuals) #q=0



> pacf(newmodel$residuals) #p=1



> acf(diff(newmodel$residuals)) #d=1

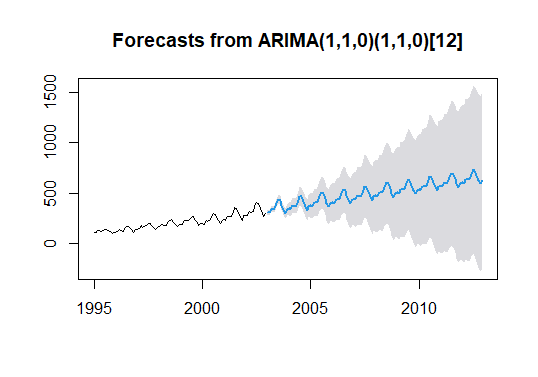


>

> #forecasting the model

> forecasting <- forecast(newmodel,level = c(95),h=10\*12)

> plot(forecasting)



######################## testing the model ##############

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

Box-Ljung test

data: newmodel$residuals

X-squared = 6.1585, df = 5, p-value = 0.2911

> Box.test(newmodel$residuals,lag = 10,type = "Ljung-Box")

Box-Ljung test

data: newmodel$residuals

X-squared = 8.8668, df = 10, p-value = 0.5448

>

> # p values are smaller

>