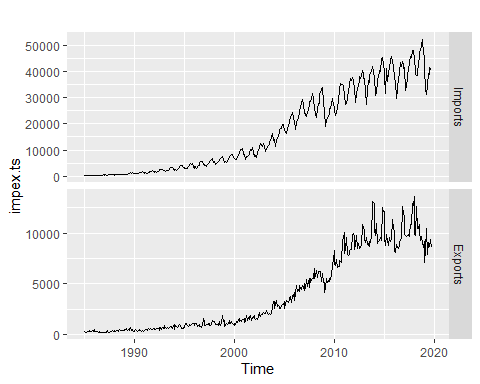
MA 611 Project Final changes

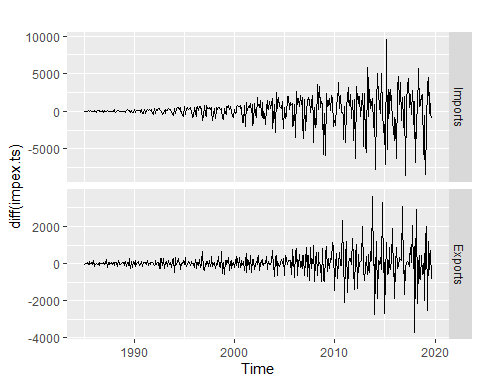
Nicolas Renaud

12/7/2019

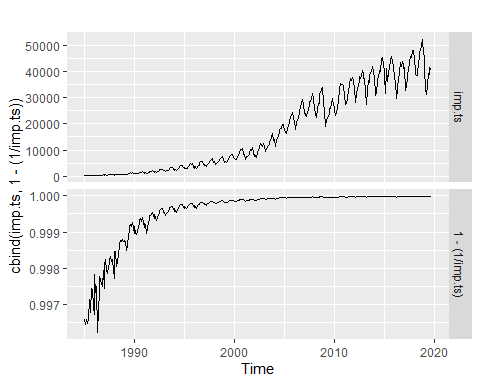
#graphing raw data side by side  
autoplot(impex.ts, facets = T)



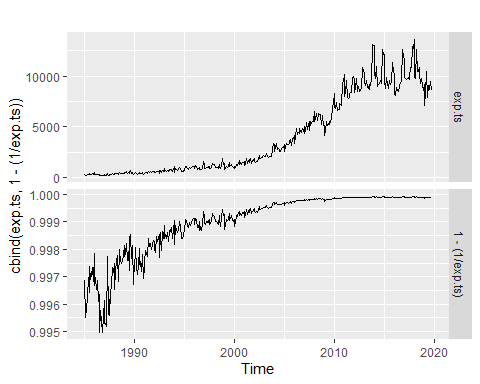
#graphing differenced data side by side  
autoplot(diff(impex.ts), facets = T)



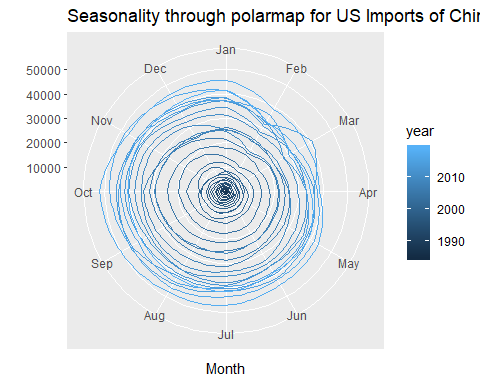
#graphing boxcox data  
autoplot(cbind(imp.ts,1-(1/imp.ts)), facets = T)



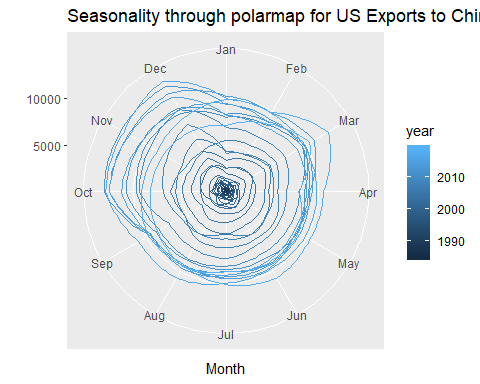
autoplot(cbind(exp.ts,1-(1/exp.ts)), facets = T)



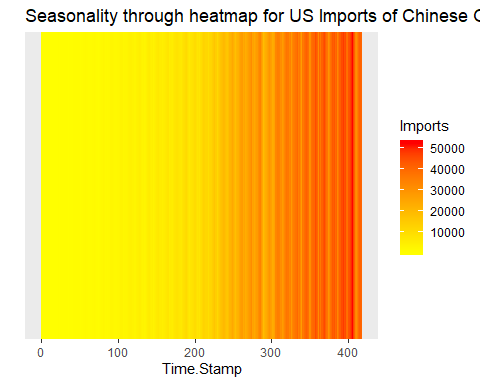
#polar plots  
ggseasonplot(imp.ts, year.labels=FALSE, continuous=TRUE, polar = TRUE)+  
 ggtitle("Seasonality through polarmap for US Imports of Chinese Goods")



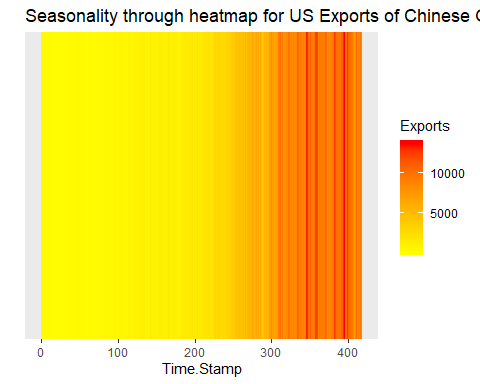
ggseasonplot(exp.ts, year.labels=FALSE, continuous=TRUE, polar = TRUE)+  
 ggtitle("Seasonality through polarmap for US Exports to China")



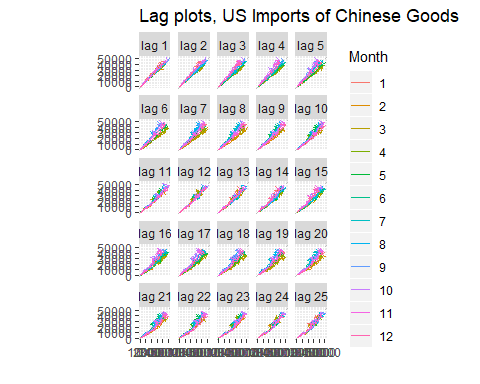
#heatmaps  
Time.Stamp = seq(1,nrow(imp),1)  
data.imp = cbind(Time.Stamp, imp)  
data.exp = cbind(Time.Stamp, exp)  
ggplot(data.imp,aes(x = Time.Stamp, y = 1)) +  
 geom\_tile(aes(fill = Imports)) +  
 scale\_fill\_gradient2(low = "navy", mid = "yellow",  
 high = "red", midpoint=28) + ggtitle("Seasonality through heatmap for US Imports of Chinese Goods")+  
 ylab("") + scale\_y\_discrete(expand=c(0,0))



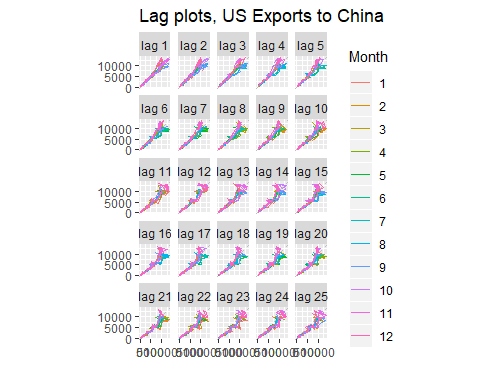
ggplot(data.exp,aes(x = Time.Stamp, y = 1)) +  
 geom\_tile(aes(fill = Exports)) +  
 scale\_fill\_gradient2(low = "navy", mid = "yellow",  
 high = "red", midpoint=28) + ggtitle("Seasonality through heatmap for US Exports of Chinese Goods")+  
 ylab("") + scale\_y\_discrete(expand=c(0,0))



#lag plots  
gglagplot(imp.ts,lags=25,set.lags = 1:25)+  
 ggtitle("Lag plots, US Imports of Chinese Goods")



gglagplot(exp.ts,lags=25,set.lags = 1:25)+  
 ggtitle("Lag plots, US Exports to China")



#checking distance  
diss(impex.ts, METHOD = "EUCL")

## Imports  
## Exports 358571.2

diss(impex.ts, METHOD = "COR")

## Imports  
## Exports 0.2604179

#checking outliers  
tsoutliers(imp.ts, lambda = "auto")

## $index  
## [1] 15 22 290 363  
##   
## $replacements  
## [1] 277.6325 519.7730 21335.2673 33007.5785

tsoutliers(exp.ts, lambda = "auto")

## $index  
## integer(0)  
##   
## $replacements  
## numeric(0)

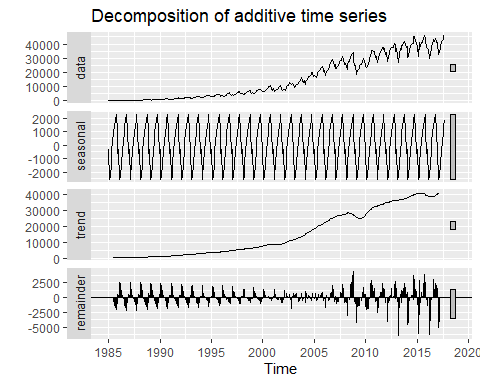
#check entropy  
SampEn(imp.ts)

## [1] 0.1134533

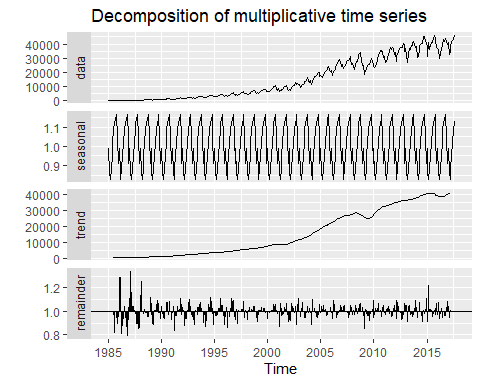
SampEn(exp.ts)

## [1] 0.1184296

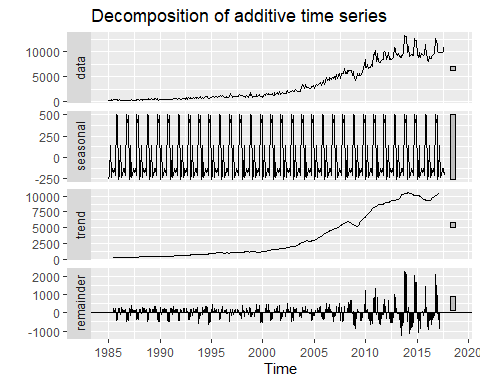
#basic decomposition  
decomp.imp.add <- decompose(train.imp.ts, type = "additive")  
decomp.imp.mult <- decompose(train.imp.ts, type = "multiplicative")  
autoplot(decomp.imp.add)



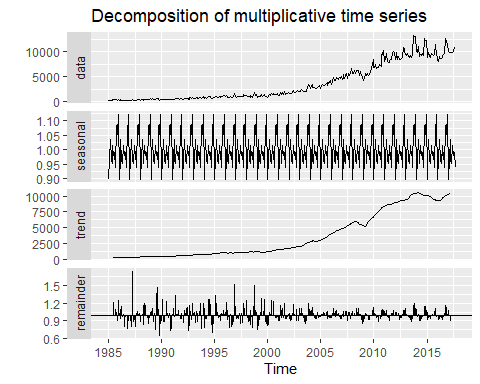
autoplot(decomp.imp.mult)



decomp.exp.add <- decompose(train.exp.ts, type = "additive")  
decomp.exp.mult <- decompose(train.exp.ts, type = "multiplicative")  
autoplot(decomp.exp.add)



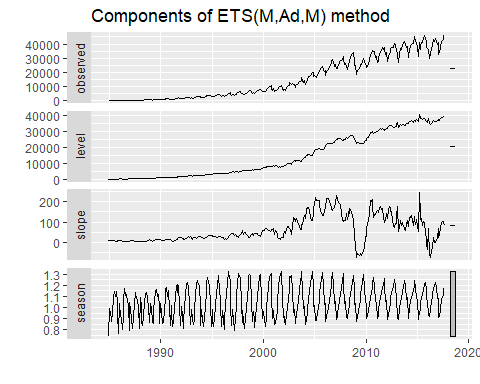
autoplot(decomp.exp.mult)



#ETS  
best.ets.imp <- ets(train.imp.ts, damped = T)  
best.ets.imp

## ETS(M,Ad,M)   
##   
## Call:  
## ets(y = train.imp.ts, damped = T)   
##   
## Smoothing parameters:  
## alpha = 0.4598   
## beta = 0.019   
## gamma = 0.2489   
## phi = 0.98   
##   
## Initial states:  
## l = 302.8255   
## b = 13.0961   
## s = 0.7433 0.9598 1.0256 1.0982 1.1458 1.1837  
## 1.1084 0.9679 0.8826 0.8985 0.9688 1.0176  
##   
## sigma: 0.0782  
##   
## AIC AICc BIC   
## 7325.354 7327.183 7396.882

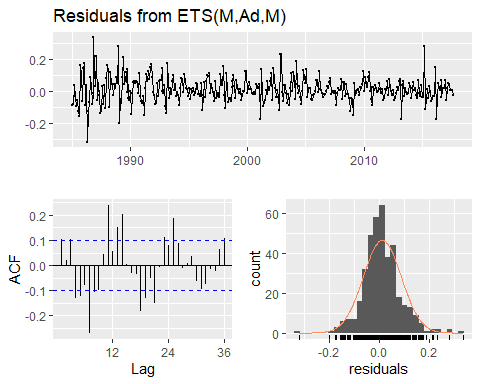
autoplot(best.ets.imp)



accuracy(forecast(best.ets.imp),imp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 54.9893 1162.354 644.2055 0.5429829 5.494712 0.3703234  
## Test set -348.9521 3524.005 3067.0773 -1.5712144 7.537235 1.7631180  
## ACF1 Theil's U  
## Training set -0.06901977 NA  
## Test set 0.75490701 1.092759

checkresiduals(forecast(best.ets.imp))

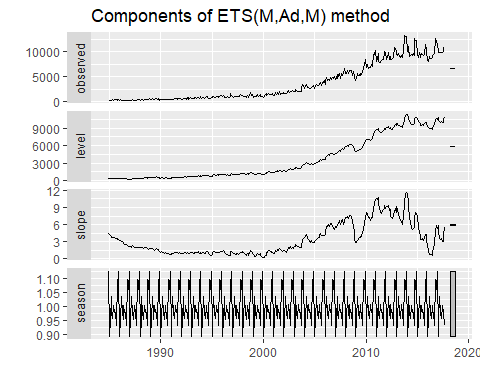


##   
## Ljung-Box test  
##   
## data: Residuals from ETS(M,Ad,M)  
## Q\* = 154.31, df = 7, p-value < 2.2e-16  
##   
## Model df: 17. Total lags used: 24

best.ets.exp <- ets(train.exp.ts)  
best.ets.exp

## ETS(M,Ad,M)   
##   
## Call:  
## ets(y = train.exp.ts)   
##   
## Smoothing parameters:  
## alpha = 0.3806   
## beta = 0.0011   
## gamma = 1e-04   
## phi = 0.98   
##   
## Initial states:  
## l = 344.1313   
## b = 4.3836   
## s = 1.1251 1.0619 1.0974 0.9334 0.9789 0.9785  
## 1.0055 0.9557 0.9801 1.035 0.9543 0.8942  
##   
## sigma: 0.1551  
##   
## AIC AICc BIC   
## 6813.701 6815.529 6885.229

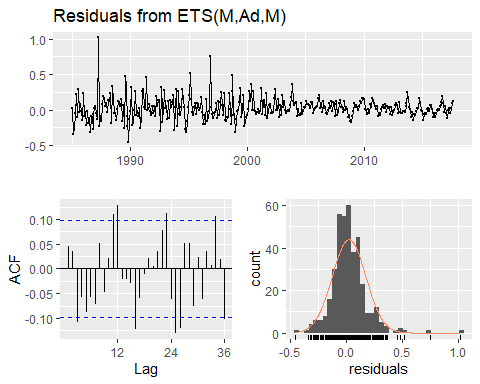
autoplot(best.ets.exp)



accuracy(forecast(best.ets.exp),exp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 62.92264 420.841 257.0866 0.1297779 10.37534 0.5103051  
## Test set -1045.83546 1742.102 1451.8324 -12.7413460 15.89129 2.8818206  
## ACF1 Theil's U  
## Training set 0.4627870 NA  
## Test set 0.6846683 1.369482

checkresiduals(forecast(best.ets.exp))



##   
## Ljung-Box test  
##   
## data: Residuals from ETS(M,Ad,M)  
## Q\* = 45.992, df = 7, p-value = 8.773e-08  
##   
## Model df: 17. Total lags used: 24

#ndiffs  
ndiffs(train.imp.ts)

## [1] 1

ndiffs(BoxCox(train.imp.ts,lambda = BoxCox.lambda(train.imp.ts)))

## [1] 1

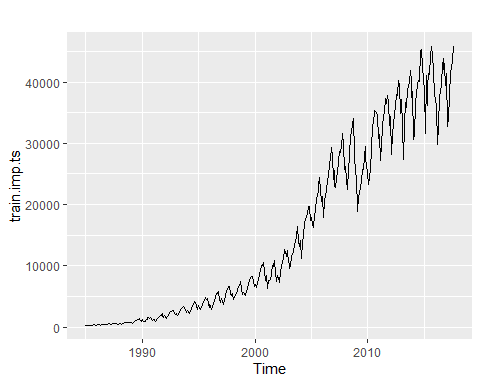
ndiffs(train.exp.ts)

## [1] 1

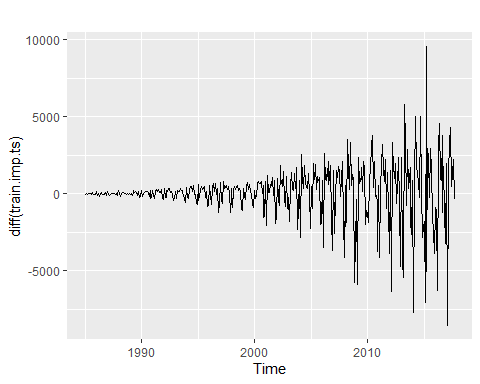
ndiffs(BoxCox(train.exp.ts,lambda = BoxCox.lambda(train.exp.ts)))

## [1] 1

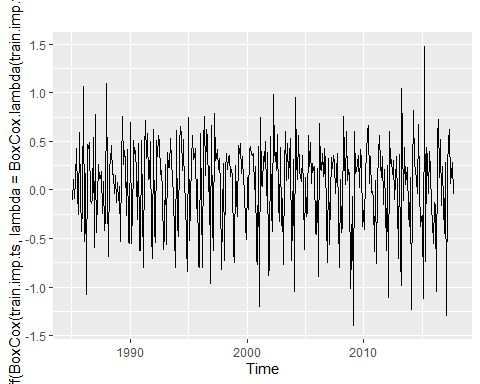
autoplot(train.imp.ts)



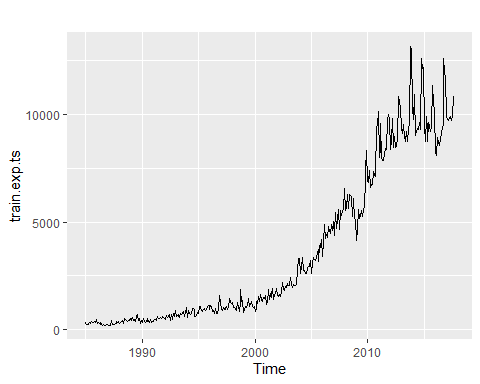
autoplot(diff(train.imp.ts))



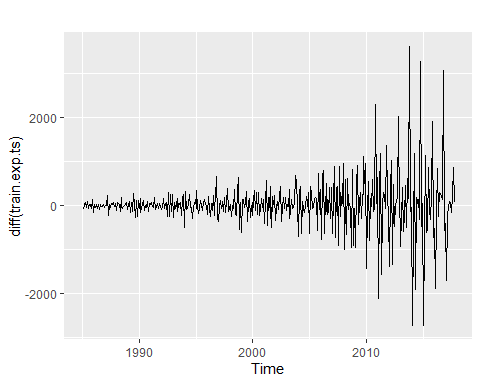
autoplot(diff(BoxCox(train.imp.ts,lambda = BoxCox.lambda(train.imp.ts))))



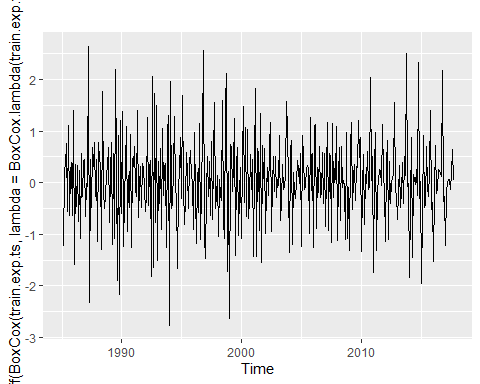
autoplot(train.exp.ts)



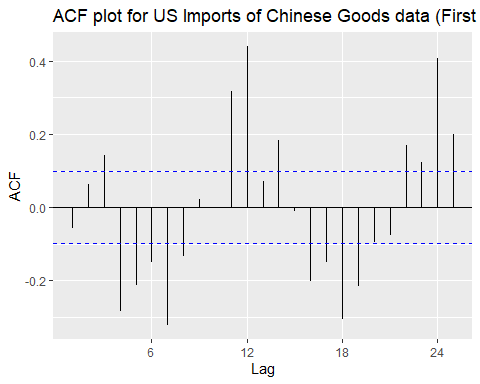
autoplot(diff(train.exp.ts))



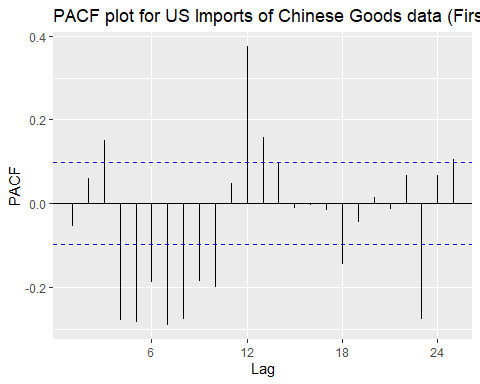
autoplot(diff(BoxCox(train.exp.ts,lambda = BoxCox.lambda(train.exp.ts))))



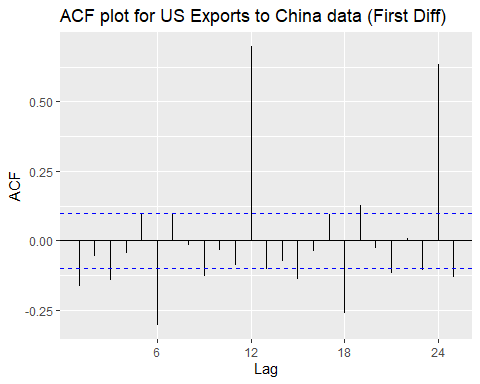
#acf and pacf plots  
ggAcf(diff(train.imp.ts),lag.max = 25)+ggtitle("ACF plot for US Imports of Chinese Goods data (First Diff)")



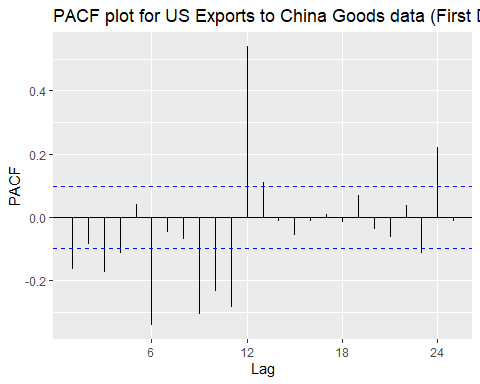
ggPacf(diff(train.imp.ts),lag.max = 25)+ggtitle("PACF plot for US Imports of Chinese Goods data (First Diff)")



ggAcf(diff(train.exp.ts),lag.max = 25)+ggtitle("ACF plot for US Exports to China data (First Diff)")



ggPacf(diff(train.exp.ts),lag.max = 25)+ggtitle("PACF plot for US Exports to China Goods data (First Diff)")



#SARIMA  
noseason.arima.imp.nolambda <- auto.arima(train.imp.ts,D=NA,max.q=0,max.P = 0,max.Q = 0,seasonal = F,stepwise = F,trace=T, lambda = NULL)

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0) : 6969.805  
## ARIMA(0,1,0) with drift : 6970.194  
## ARIMA(1,1,0) : 6971.79  
## ARIMA(1,1,0) with drift : 6971.997  
## ARIMA(2,1,0) : 6973.183  
## ARIMA(2,1,0) with drift : 6973.604  
## ARIMA(3,1,0) : 6966.874  
## ARIMA(3,1,0) with drift : 6967.698  
## ARIMA(4,1,0) : 6938.593  
## ARIMA(4,1,0) with drift : 6938.575  
## ARIMA(5,1,0) : 6909.696  
## ARIMA(5,1,0) with drift : 6908.25  
##   
## Now re-fitting the best model(s) without approximations...  
##   
##   
##   
##   
## Best model: ARIMA(5,1,0) with drift

noseason.arima.imp.yeslambda <- auto.arima(train.imp.ts,D=NA,max.q=0,max.P = 0,max.Q = 0,seasonal = F,stepwise = F,trace=T, lambda = "auto")

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0) : 515.3495  
## ARIMA(0,1,0) with drift : 512.6801  
## ARIMA(1,1,0) : 518.0667  
## ARIMA(1,1,0) with drift : 515.0349  
## ARIMA(2,1,0) : 521.0413  
## ARIMA(2,1,0) with drift : 517.94  
## ARIMA(3,1,0) : 512.0489  
## ARIMA(3,1,0) with drift : 510.3645  
## ARIMA(4,1,0) : 498.6016  
## ARIMA(4,1,0) with drift : 495.0875  
## ARIMA(5,1,0) : 475.2692  
## ARIMA(5,1,0) with drift : 468.2586  
##   
## Now re-fitting the best model(s) without approximations...  
##   
##   
##   
##   
## Best model: ARIMA(5,1,0) with drift

summary(noseason.arima.imp.nolambda)

## Series: train.imp.ts   
## ARIMA(5,1,0) with drift   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 drift  
## -0.0992 0.1237 0.1557 -0.2851 -0.2847 111.1858  
## s.e. 0.0483 0.0464 0.0461 0.0463 0.0485 58.9127  
##   
## sigma^2 estimated as 2648779: log likelihood=-3452.38  
## AIC=6918.75 AICc=6919.04 BIC=6946.55  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.4197096 1612.948 968.2555 -5.589382 10.90033 0.5566044  
## ACF1  
## Training set -0.05593988

summary(noseason.arima.imp.yeslambda)

## Series: train.imp.ts   
## ARIMA(5,1,0) with drift   
## Box Cox transformation: lambda= 0.1637399   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 drift  
## -0.0577 0.0268 0.1503 -0.2120 -0.2660 0.0502  
## s.e. 0.0486 0.0474 0.0467 0.0473 0.0485 0.0160  
##   
## sigma^2 estimated as 0.1866: log likelihood=-224.43  
## AIC=462.85 AICc=463.14 BIC=490.65  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -37.72435 1665.825 968.1145 -0.5201503 8.215522 0.5565234  
## ACF1  
## Training set -0.09214115

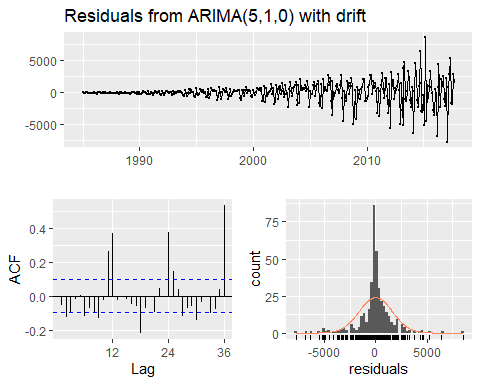
accuracy(forecast(noseason.arima.imp.nolambda),imp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.4197096 1612.948 968.2555 -5.589382 10.90033 0.5566044  
## Test set -2514.6031825 6091.425 4901.6352 -7.726018 12.61410 2.8177188  
## ACF1 Theil's U  
## Training set -0.05593988 NA  
## Test set 0.78134720 2.003936

accuracy(forecast(noseason.arima.imp.yeslambda),imp.ts)

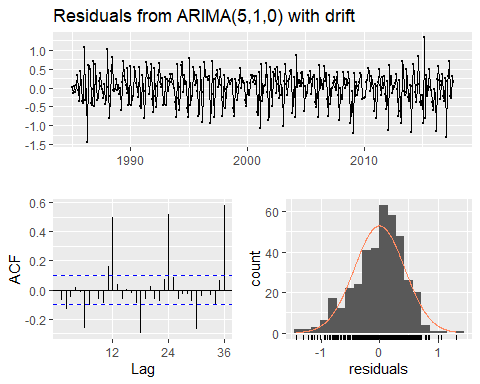
## ME RMSE MAE MPE MAPE  
## Training set -37.72435 1665.825 968.1145 -0.5201503 8.215522  
## Test set -6535.99239 9579.743 7464.4986 -17.6262380 19.515919  
## MASE ACF1 Theil's U  
## Training set 0.5565234 -0.09214115 NA  
## Test set 4.2909881 0.81354490 3.173049

checkresiduals(noseason.arima.imp.nolambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(5,1,0) with drift  
## Q\* = 204.89, df = 18, p-value < 2.2e-16  
##   
## Model df: 6. Total lags used: 24

checkresiduals(noseason.arima.imp.yeslambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(5,1,0) with drift  
## Q\* = 320.53, df = 18, p-value < 2.2e-16  
##   
## Model df: 6. Total lags used: 24

best.arima.imp.nolambda <- auto.arima(train.imp.ts, stepwise = F, trace = T, lambda = NULL)

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0)(0,1,0)[12] : 6591.417  
## ARIMA(0,1,0)(0,1,1)[12] : 6366.686  
## ARIMA(0,1,0)(0,1,2)[12] : 6343.18  
## ARIMA(0,1,0)(1,1,0)[12] : 6503.06  
## ARIMA(0,1,0)(1,1,1)[12] : 6371.635  
## ARIMA(0,1,0)(1,1,2)[12] : Inf  
## ARIMA(0,1,0)(2,1,0)[12] : 6358.418  
## ARIMA(0,1,0)(2,1,1)[12] : 6342.923  
## ARIMA(0,1,0)(2,1,2)[12] : 6314.218  
## ARIMA(0,1,1)(0,1,0)[12] : 6459.199  
## ARIMA(0,1,1)(0,1,1)[12] : 6275.079  
## ARIMA(0,1,1)(0,1,2)[12] : 6262.578  
## ARIMA(0,1,1)(1,1,0)[12] : 6367.146  
## ARIMA(0,1,1)(1,1,1)[12] : 6282.22  
## ARIMA(0,1,1)(1,1,2)[12] : Inf  
## ARIMA(0,1,1)(2,1,0)[12] : 6271.138  
## ARIMA(0,1,1)(2,1,1)[12] : 6268.336  
## ARIMA(0,1,1)(2,1,2)[12] : 6254.167  
## ARIMA(0,1,2)(0,1,0)[12] : 6446.429  
## ARIMA(0,1,2)(0,1,1)[12] : 6274.705  
## ARIMA(0,1,2)(0,1,2)[12] : 6263.731  
## ARIMA(0,1,2)(1,1,0)[12] : 6359.862  
## ARIMA(0,1,2)(1,1,1)[12] : 6282.655  
## ARIMA(0,1,2)(1,1,2)[12] : Inf  
## ARIMA(0,1,2)(2,1,0)[12] : 6267.502  
## ARIMA(0,1,2)(2,1,1)[12] : 6268.077  
## ARIMA(0,1,3)(0,1,0)[12] : 6432.869  
## ARIMA(0,1,3)(0,1,1)[12] : 6269.439  
## ARIMA(0,1,3)(0,1,2)[12] : 6257.197  
## ARIMA(0,1,3)(1,1,0)[12] : 6343.046  
## ARIMA(0,1,3)(1,1,1)[12] : 6276.444  
## ARIMA(0,1,3)(2,1,0)[12] : 6254.345  
## ARIMA(0,1,4)(0,1,0)[12] : 6434.395  
## ARIMA(0,1,4)(0,1,1)[12] : 6263.04  
## ARIMA(0,1,4)(1,1,0)[12] : 6340.998  
## ARIMA(0,1,5)(0,1,0)[12] : 6398.521  
## ARIMA(1,1,0)(0,1,0)[12] : 6493.085  
## ARIMA(1,1,0)(0,1,1)[12] : 6301.643  
## ARIMA(1,1,0)(0,1,2)[12] : 6289.724  
## ARIMA(1,1,0)(1,1,0)[12] : 6414.004  
## ARIMA(1,1,0)(1,1,1)[12] : 6310.312  
## ARIMA(1,1,0)(1,1,2)[12] : Inf  
## ARIMA(1,1,0)(2,1,0)[12] : 6291.367  
## ARIMA(1,1,0)(2,1,1)[12] : 6286.61  
## ARIMA(1,1,0)(2,1,2)[12] : 6264.425  
## ARIMA(1,1,1)(0,1,0)[12] : 6454.737  
## ARIMA(1,1,1)(0,1,1)[12] : 6276.766  
## ARIMA(1,1,1)(0,1,2)[12] : 6265.144  
## ARIMA(1,1,1)(1,1,0)[12] : 6365.639  
## ARIMA(1,1,1)(1,1,1)[12] : 6284.449  
## ARIMA(1,1,1)(1,1,2)[12] : Inf  
## ARIMA(1,1,1)(2,1,0)[12] : 6271.619  
## ARIMA(1,1,1)(2,1,1)[12] : 6270.27  
## ARIMA(1,1,2)(0,1,0)[12] : 6436.453  
## ARIMA(1,1,2)(0,1,1)[12] : 6276.826  
## ARIMA(1,1,2)(0,1,2)[12] : 6257.55  
## ARIMA(1,1,2)(1,1,0)[12] : 6355.712  
## ARIMA(1,1,2)(1,1,1)[12] : 6285.005  
## ARIMA(1,1,2)(2,1,0)[12] : 6268.447  
## ARIMA(1,1,3)(0,1,0)[12] : 6415.939  
## ARIMA(1,1,3)(0,1,1)[12] : 6261.756  
## ARIMA(1,1,3)(1,1,0)[12] : 6333.878  
## ARIMA(1,1,4)(0,1,0)[12] : 6415.679  
## ARIMA(2,1,0)(0,1,0)[12] : 6433.031  
## ARIMA(2,1,0)(0,1,1)[12] : 6259.309  
## ARIMA(2,1,0)(0,1,2)[12] : 6245.526  
## ARIMA(2,1,0)(1,1,0)[12] : 6335.307  
## ARIMA(2,1,0)(1,1,1)[12] : 6264.968  
## ARIMA(2,1,0)(1,1,2)[12] : Inf  
## ARIMA(2,1,0)(2,1,0)[12] : 6248.404  
## ARIMA(2,1,0)(2,1,1)[12] : 6249.338  
## ARIMA(2,1,1)(0,1,0)[12] : 6433.427  
## ARIMA(2,1,1)(0,1,1)[12] : 6254.077  
## ARIMA(2,1,1)(0,1,2)[12] : 6235.662  
## ARIMA(2,1,1)(1,1,0)[12] : 6332.719  
## ARIMA(2,1,1)(1,1,1)[12] : 6257.333  
## ARIMA(2,1,1)(2,1,0)[12] : 6228.538  
## ARIMA(2,1,2)(0,1,0)[12] : 6435.409  
## ARIMA(2,1,2)(0,1,1)[12] : 6255.107  
## ARIMA(2,1,2)(1,1,0)[12] : 6334.471  
## ARIMA(2,1,3)(0,1,0)[12] : 6413.105  
## ARIMA(3,1,0)(0,1,0)[12] : 6434.72  
## ARIMA(3,1,0)(0,1,1)[12] : 6256.398  
## ARIMA(3,1,0)(0,1,2)[12] : 6239.329  
## ARIMA(3,1,0)(1,1,0)[12] : 6335.444  
## ARIMA(3,1,0)(1,1,1)[12] : 6260.414  
## ARIMA(3,1,0)(2,1,0)[12] : 6230.776  
## ARIMA(3,1,1)(0,1,0)[12] : 6436.836  
## ARIMA(3,1,1)(0,1,1)[12] : 6256.874  
## ARIMA(3,1,1)(1,1,0)[12] : 6335.737  
## ARIMA(3,1,2)(0,1,0)[12] : 6438.541  
## ARIMA(4,1,0)(0,1,0)[12] : 6436.923  
## ARIMA(4,1,0)(0,1,1)[12] : 6255.3  
## ARIMA(4,1,0)(1,1,0)[12] : 6332.217  
## ARIMA(4,1,1)(0,1,0)[12] : 6438.987  
## ARIMA(5,1,0)(0,1,0)[12] : 6439.996  
##   
## Now re-fitting the best model(s) without approximations...  
##   
##   
##   
##   
## Best model: ARIMA(2,1,1)(2,1,0)[12]

best.arima.imp.yeslambda <- auto.arima(train.imp.ts, stepwise = F, trace = T, lambda = "auto")

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0)(0,1,0)[12] : 459.3749  
## ARIMA(0,1,0)(0,1,1)[12] : 258.9774  
## ARIMA(0,1,0)(0,1,2)[12] : 255.2146  
## ARIMA(0,1,0)(1,1,0)[12] : 338.4986  
## ARIMA(0,1,0)(1,1,1)[12] : 263.7265  
## ARIMA(0,1,0)(1,1,2)[12] : 250.2723  
## ARIMA(0,1,0)(2,1,0)[12] : 233.6395  
## ARIMA(0,1,0)(2,1,1)[12] : 211.6003  
## ARIMA(0,1,0)(2,1,2)[12] : 200.3274  
## ARIMA(0,1,1)(0,1,0)[12] : 339.888  
## ARIMA(0,1,1)(0,1,1)[12] : 177.7234  
## ARIMA(0,1,1)(0,1,2)[12] : 174.9004  
## ARIMA(0,1,1)(1,1,0)[12] : 215.6836  
## ARIMA(0,1,1)(1,1,1)[12] : 170.3614  
## ARIMA(0,1,1)(1,1,2)[12] : 161.7845  
## ARIMA(0,1,1)(2,1,0)[12] : 147.7407  
## ARIMA(0,1,1)(2,1,1)[12] : 147.9331  
## ARIMA(0,1,1)(2,1,2)[12] : 136.5901  
## ARIMA(0,1,2)(0,1,0)[12] : 335.7651  
## ARIMA(0,1,2)(0,1,1)[12] : 179.5335  
## ARIMA(0,1,2)(0,1,2)[12] : 176.7671  
## ARIMA(0,1,2)(1,1,0)[12] : 211.3734  
## ARIMA(0,1,2)(1,1,1)[12] : 170.7308  
## ARIMA(0,1,2)(1,1,2)[12] : 161.5354  
## ARIMA(0,1,2)(2,1,0)[12] : 148.3539  
## ARIMA(0,1,2)(2,1,1)[12] : 149.8641  
## ARIMA(0,1,3)(0,1,0)[12] : 327.7125  
## ARIMA(0,1,3)(0,1,1)[12] : 177.2251  
## ARIMA(0,1,3)(0,1,2)[12] : 174.2344  
## ARIMA(0,1,3)(1,1,0)[12] : 204.9298  
## ARIMA(0,1,3)(1,1,1)[12] : 168.9732  
## ARIMA(0,1,3)(2,1,0)[12] : 143.3078  
## ARIMA(0,1,4)(0,1,0)[12] : 327.281  
## ARIMA(0,1,4)(0,1,1)[12] : 169.9702  
## ARIMA(0,1,4)(1,1,0)[12] : 200.0081  
## ARIMA(0,1,5)(0,1,0)[12] : 325.8261  
## ARIMA(1,1,0)(0,1,0)[12] : 373.3351  
## ARIMA(1,1,0)(0,1,1)[12] : 202.9009  
## ARIMA(1,1,0)(0,1,2)[12] : 201.5312  
## ARIMA(1,1,0)(1,1,0)[12] : 249.8883  
## ARIMA(1,1,0)(1,1,1)[12] : 192.0213  
## ARIMA(1,1,0)(1,1,2)[12] : 183.6239  
## ARIMA(1,1,0)(2,1,0)[12] : 171.3848  
## ARIMA(1,1,0)(2,1,1)[12] : 167.5324  
## ARIMA(1,1,0)(2,1,2)[12] : 153.4016  
## ARIMA(1,1,1)(0,1,0)[12] : 336.3198  
## ARIMA(1,1,1)(0,1,1)[12] : 174.7221  
## ARIMA(1,1,1)(0,1,2)[12] : 173.0749  
## ARIMA(1,1,1)(1,1,0)[12] : 213.6499  
## ARIMA(1,1,1)(1,1,1)[12] : 171.0159  
## ARIMA(1,1,1)(1,1,2)[12] : 166.7771  
## ARIMA(1,1,1)(2,1,0)[12] : 150.5047  
## ARIMA(1,1,1)(2,1,1)[12] : 150.302  
## ARIMA(1,1,2)(0,1,0)[12] : 327.007  
## ARIMA(1,1,2)(0,1,1)[12] : 174.7143  
## ARIMA(1,1,2)(0,1,2)[12] : 172.9679  
## ARIMA(1,1,2)(1,1,0)[12] : 212.7633  
## ARIMA(1,1,2)(1,1,1)[12] : 143.5549  
## ARIMA(1,1,2)(2,1,0)[12] : 151.6133  
## ARIMA(1,1,3)(0,1,0)[12] : 317.947  
## ARIMA(1,1,3)(0,1,1)[12] : 171.4925  
## ARIMA(1,1,3)(1,1,0)[12] : 185.0105  
## ARIMA(1,1,4)(0,1,0)[12] : 319.656  
## ARIMA(2,1,0)(0,1,0)[12] : 322.3833  
## ARIMA(2,1,0)(0,1,1)[12] : 164.918  
## ARIMA(2,1,0)(0,1,2)[12] : 162.6835  
## ARIMA(2,1,0)(1,1,0)[12] : 190.0484  
## ARIMA(2,1,0)(1,1,1)[12] : 153.0369  
## ARIMA(2,1,0)(1,1,2)[12] : 149.8713  
## ARIMA(2,1,0)(2,1,0)[12] : 129.7072  
## ARIMA(2,1,0)(2,1,1)[12] : 130.7216  
## ARIMA(2,1,1)(0,1,0)[12] : 324.335  
## ARIMA(2,1,1)(0,1,1)[12] : 164.9235  
## ARIMA(2,1,1)(0,1,2)[12] : 162.162  
## ARIMA(2,1,1)(1,1,0)[12] : 189.7893  
## ARIMA(2,1,1)(1,1,1)[12] : 150.4265  
## ARIMA(2,1,1)(2,1,0)[12] : 122.6802  
## ARIMA(2,1,2)(0,1,0)[12] : 325.7779  
## ARIMA(2,1,2)(0,1,1)[12] : 166.9388  
## ARIMA(2,1,2)(1,1,0)[12] : 191.8425  
## ARIMA(2,1,3)(0,1,0)[12] : 315.093  
## ARIMA(3,1,0)(0,1,0)[12] : 315.4695  
## ARIMA(3,1,0)(0,1,1)[12] : 172.1197  
## ARIMA(3,1,0)(0,1,2)[12] : 168.3853  
## ARIMA(3,1,0)(1,1,0)[12] : 190.1195  
## ARIMA(3,1,0)(1,1,1)[12] : 141.1605  
## ARIMA(3,1,0)(2,1,0)[12] : 124.7128  
## ARIMA(3,1,1)(0,1,0)[12] : 317.2372  
## ARIMA(3,1,1)(0,1,1)[12] : 173.6408  
## ARIMA(3,1,1)(1,1,0)[12] : 192.0755  
## ARIMA(3,1,2)(0,1,0)[12] : 318.9437  
## ARIMA(4,1,0)(0,1,0)[12] : 317.4693  
## ARIMA(4,1,0)(0,1,1)[12] : 173.9525  
## ARIMA(4,1,0)(1,1,0)[12] : 192.7202  
## ARIMA(4,1,1)(0,1,0)[12] : 319.5311  
## ARIMA(5,1,0)(0,1,0)[12] : 320.2833  
##   
## Now re-fitting the best model(s) without approximations...  
##   
##   
##   
##   
## Best model: ARIMA(2,1,1)(2,1,0)[12]

summary(best.arima.imp.nolambda)

## Series: train.imp.ts   
## ARIMA(2,1,1)(2,1,0)[12]   
##   
## Coefficients:  
## ar1 ar2 ma1 sar1 sar2  
## -0.9614 -0.5410 0.4616 -0.7938 -0.5975  
## s.e. 0.0731 0.0462 0.0799 0.0466 0.0512  
##   
## sigma^2 estimated as 1180427: log likelihood=-3200.48  
## AIC=6412.97 AICc=6413.19 BIC=6436.61  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 8.237035 1061.302 631.1061 -0.0213644 5.356059 0.3627931  
## ACF1  
## Training set 0.02655289

summary(best.arima.imp.yeslambda)

## Series: train.imp.ts   
## ARIMA(2,1,1)(2,1,0)[12]   
## Box Cox transformation: lambda= 0.1637399   
##   
## Coefficients:  
## ar1 ar2 ma1 sar1 sar2  
## -0.8298 -0.4428 0.3331 -0.7430 -0.4380  
## s.e. 0.1037 0.0538 0.1119 0.0477 0.0519  
##   
## sigma^2 estimated as 0.07768: log likelihood=-55.83  
## AIC=123.66 AICc=123.88 BIC=147.3  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -34.37358 1076.89 597.2436 -0.4262426 4.977908 0.3433272  
## ACF1  
## Training set -0.04637189

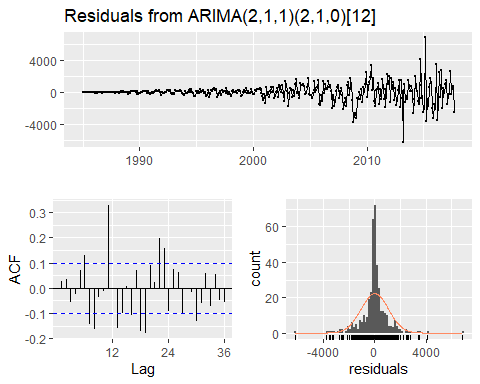
accuracy(forecast(best.arima.imp.nolambda),imp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 8.237035 1061.302 631.1061 -0.0213644 5.356059 0.3627931  
## Test set 390.503477 3280.559 2884.8431 0.2099417 6.900162 1.6583602  
## ACF1 Theil's U  
## Training set 0.02655289 NA  
## Test set 0.64841666 0.9497607

accuracy(forecast(best.arima.imp.yeslambda),imp.ts)

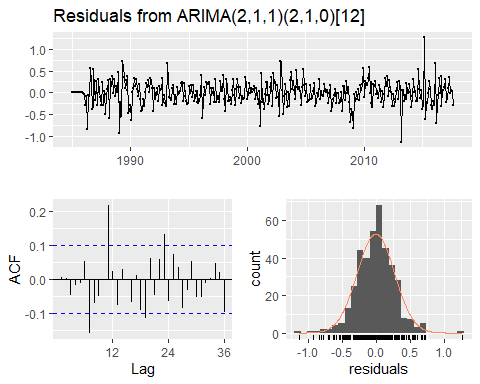
## ME RMSE MAE MPE MAPE MASE  
## Training set -34.37358 1076.890 597.2436 -0.4262426 4.977908 0.3433272  
## Test set 643.62276 3440.545 3070.5692 0.8346843 7.287937 1.7651253  
## ACF1 Theil's U  
## Training set -0.04637189 NA  
## Test set 0.70960777 0.9910815

checkresiduals(best.arima.imp.nolambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1)(2,1,0)[12]  
## Q\* = 154.92, df = 19, p-value < 2.2e-16  
##   
## Model df: 5. Total lags used: 24

checkresiduals(best.arima.imp.yeslambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1)(2,1,0)[12]  
## Q\* = 60.985, df = 19, p-value = 2.7e-06  
##   
## Model df: 5. Total lags used: 24

noseason.arima.exp.nolambda <- auto.arima(train.exp.ts,D=NA,max.q=0,max.P = 0,max.Q = 0,seasonal = F,stepwise = F,trace=T, lambda = NULL)

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0) : 6088.81  
## ARIMA(0,1,0) with drift : 6089.978  
## ARIMA(1,1,0) : 6081.412  
## ARIMA(1,1,0) with drift : 6082.239  
## ARIMA(2,1,0) : 6081.792  
## ARIMA(2,1,0) with drift : 6082.424  
## ARIMA(3,1,0) : 6073.306  
## ARIMA(3,1,0) with drift : 6073.417  
## ARIMA(4,1,0) : 6071.647  
## ARIMA(4,1,0) with drift : 6071.321  
## ARIMA(5,1,0) : 6073.909  
## ARIMA(5,1,0) with drift : 6073.76  
##   
## Now re-fitting the best model(s) without approximations...  
##   
##   
##   
##   
## Best model: ARIMA(4,1,0) with drift

noseason.arima.exp.yeslambda <- auto.arima(train.exp.ts,D=NA,max.q=0,max.P = 0,max.Q = 0,seasonal = F,stepwise = F,trace=T, lambda = "auto")

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0) : 1005.913  
## ARIMA(0,1,0) with drift : 1006.716  
## ARIMA(1,1,0) : 945.687  
## ARIMA(1,1,0) with drift : 944.7161  
## ARIMA(2,1,0) : 940.3536  
## ARIMA(2,1,0) with drift : 938.3228  
## ARIMA(3,1,0) : 932.4606  
## ARIMA(3,1,0) with drift : 928.9403  
## ARIMA(4,1,0) : 931.6533  
## ARIMA(4,1,0) with drift : 927.2834  
## ARIMA(5,1,0) : 934.2594  
## ARIMA(5,1,0) with drift : 929.282  
##   
## Now re-fitting the best model(s) without approximations...  
##   
##   
##   
##   
## Best model: ARIMA(4,1,0) with drift

summary(noseason.arima.exp.nolambda)

## Series: train.exp.ts   
## ARIMA(4,1,0) with drift   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 drift  
## -0.2127 -0.1298 -0.1957 -0.1137 26.4232  
## s.e. 0.0501 0.0504 0.0504 0.0501 17.0396  
##   
## sigma^2 estimated as 313070: log likelihood=-3034.02  
## AIC=6080.04 AICc=6080.26 BIC=6103.87  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.1555558 555.2387 311.5878 -4.757934 13.33342 0.6184875  
## ACF1  
## Training set 0.00399125

summary(noseason.arima.exp.yeslambda)

## Series: train.exp.ts   
## ARIMA(4,1,0) with drift   
## Box Cox transformation: lambda= 0.220129   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 drift  
## -0.4883 -0.2543 -0.2232 -0.1002 0.0493  
## s.e. 0.0503 0.0549 0.0549 0.0502 0.0190  
##   
## sigma^2 estimated as 0.6104: log likelihood=-457.11  
## AIC=926.22 AICc=926.44 BIC=950.05  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 10.19858 577.7833 317.036 -1.510817 11.66505 0.629302  
## ACF1  
## Training set 0.2744815

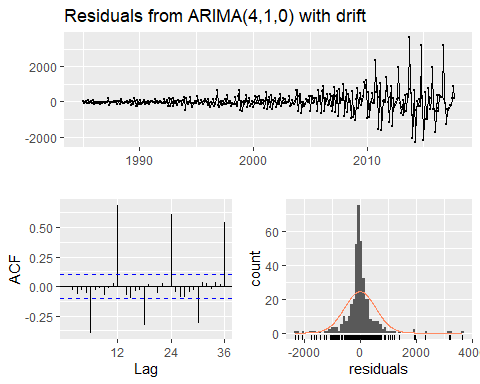
accuracy(forecast(noseason.arima.exp.nolambda))

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.1555558 555.2387 311.5878 -4.757934 13.33342 0.6184875  
## ACF1  
## Training set 0.00399125

accuracy(forecast(noseason.arima.exp.yeslambda))

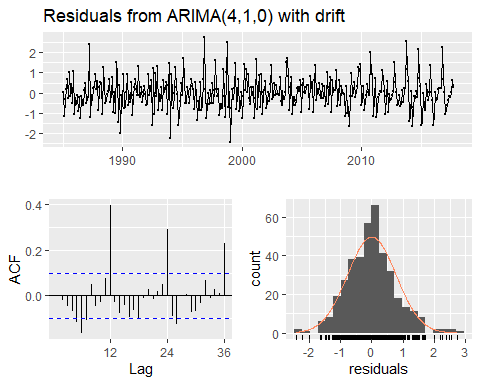
## ME RMSE MAE MPE MAPE MASE  
## Training set 10.19858 577.7833 317.036 -1.510817 11.66505 0.629302  
## ACF1  
## Training set 0.2744815

checkresiduals(noseason.arima.exp.nolambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,1,0) with drift  
## Q\* = 469.8, df = 19, p-value < 2.2e-16  
##   
## Model df: 5. Total lags used: 24

checkresiduals(noseason.arima.exp.yeslambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,1,0) with drift  
## Q\* = 142.35, df = 19, p-value < 2.2e-16  
##   
## Model df: 5. Total lags used: 24

best.arima.exp.nolambda <- auto.arima(train.exp.ts, stepwise = F, trace = T, lambda = NULL)

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0)(0,1,0)[12] : 5503.952  
## ARIMA(0,1,0)(0,1,1)[12] : 5393.719  
## ARIMA(0,1,0)(0,1,2)[12] : 5395.329  
## ARIMA(0,1,0)(1,1,0)[12] : 5442.184  
## ARIMA(0,1,0)(1,1,1)[12] : 5406.59  
## ARIMA(0,1,0)(1,1,2)[12] : 5406.325  
## ARIMA(0,1,0)(2,1,0)[12] : 5423.988  
## ARIMA(0,1,0)(2,1,1)[12] : 5417.054  
## ARIMA(0,1,0)(2,1,2)[12] : 5415.915  
## ARIMA(0,1,1)(0,1,0)[12] : 5476.324  
## ARIMA(0,1,1)(0,1,1)[12] : 5379.4  
## ARIMA(0,1,1)(0,1,2)[12] : 5381.406  
## ARIMA(0,1,1)(1,1,0)[12] : 5426.377  
## ARIMA(0,1,1)(1,1,1)[12] : 5392.751  
## ARIMA(0,1,1)(1,1,2)[12] : 5394.415  
## ARIMA(0,1,1)(2,1,0)[12] : 5404.607  
## ARIMA(0,1,1)(2,1,1)[12] : 5400.859  
## ARIMA(0,1,1)(2,1,2)[12] : 5401.04  
## ARIMA(0,1,2)(0,1,0)[12] : 5477.958  
## ARIMA(0,1,2)(0,1,1)[12] : 5381.306  
## ARIMA(0,1,2)(0,1,2)[12] : 5383.312  
## ARIMA(0,1,2)(1,1,0)[12] : 5428.404  
## ARIMA(0,1,2)(1,1,1)[12] : 5394.643  
## ARIMA(0,1,2)(1,1,2)[12] : 5396.362  
## ARIMA(0,1,2)(2,1,0)[12] : 5406.564  
## ARIMA(0,1,2)(2,1,1)[12] : 5402.774  
## ARIMA(0,1,3)(0,1,0)[12] : 5479.833  
## ARIMA(0,1,3)(0,1,1)[12] : 5382.332  
## ARIMA(0,1,3)(0,1,2)[12] : 5384.345  
## ARIMA(0,1,3)(1,1,0)[12] : 5430.099  
## ARIMA(0,1,3)(1,1,1)[12] : 5395.719  
## ARIMA(0,1,3)(2,1,0)[12] : 5408.014  
## ARIMA(0,1,4)(0,1,0)[12] : 5477.616  
## ARIMA(0,1,4)(0,1,1)[12] : 5377.851  
## ARIMA(0,1,4)(1,1,0)[12] : 5427.682  
## ARIMA(0,1,5)(0,1,0)[12] : 5477.586  
## ARIMA(1,1,0)(0,1,0)[12] : 5477.57  
## ARIMA(1,1,0)(0,1,1)[12] : 5381.149  
## ARIMA(1,1,0)(0,1,2)[12] : 5383.157  
## ARIMA(1,1,0)(1,1,0)[12] : 5428.267  
## ARIMA(1,1,0)(1,1,1)[12] : 5394.812  
## ARIMA(1,1,0)(1,1,2)[12] : 5396.441  
## ARIMA(1,1,0)(2,1,0)[12] : 5407.167  
## ARIMA(1,1,0)(2,1,1)[12] : 5403.132  
## ARIMA(1,1,0)(2,1,2)[12] : 5403.098  
## ARIMA(1,1,1)(0,1,0)[12] : 5478.856  
## ARIMA(1,1,1)(0,1,1)[12] : 5375.32  
## ARIMA(1,1,1)(0,1,2)[12] : 5377.274  
## ARIMA(1,1,1)(1,1,0)[12] : 5429.42  
## ARIMA(1,1,1)(1,1,1)[12] : 5387.768  
## ARIMA(1,1,1)(1,1,2)[12] : 5389.061  
## ARIMA(1,1,1)(2,1,0)[12] : 5402.963  
## ARIMA(1,1,1)(2,1,1)[12] : 5397.09  
## ARIMA(1,1,2)(0,1,0)[12] : 5480.9  
## ARIMA(1,1,2)(0,1,1)[12] : 5384.171  
## ARIMA(1,1,2)(0,1,2)[12] : 5386.192  
## ARIMA(1,1,2)(1,1,0)[12] : 5431.494  
## ARIMA(1,1,2)(1,1,1)[12] : 5386.022  
## ARIMA(1,1,2)(2,1,0)[12] : 5409.643  
## ARIMA(1,1,3)(0,1,0)[12] : 5464.492  
## ARIMA(1,1,3)(0,1,1)[12] : 5375.035  
## ARIMA(1,1,3)(1,1,0)[12] : 5416.302  
## ARIMA(1,1,4)(0,1,0)[12] : 5462.566  
## ARIMA(2,1,0)(0,1,0)[12] : 5479.724  
## ARIMA(2,1,0)(0,1,1)[12] : 5383.348  
## ARIMA(2,1,0)(0,1,2)[12] : 5385.362  
## ARIMA(2,1,0)(1,1,0)[12] : 5430.532  
## ARIMA(2,1,0)(1,1,1)[12] : 5396.968  
## ARIMA(2,1,0)(1,1,2)[12] : 5398.656  
## ARIMA(2,1,0)(2,1,0)[12] : 5408.755  
## ARIMA(2,1,0)(2,1,1)[12] : 5405.051  
## ARIMA(2,1,1)(0,1,0)[12] : 5470.187  
## ARIMA(2,1,1)(0,1,1)[12] : 5372.225  
## ARIMA(2,1,1)(0,1,2)[12] : 5374.287  
## ARIMA(2,1,1)(1,1,0)[12] : 5432.589  
## ARIMA(2,1,1)(1,1,1)[12] : Inf  
## ARIMA(2,1,1)(2,1,0)[12] : 5410.85  
## ARIMA(2,1,2)(0,1,0)[12] : 5482.817  
## ARIMA(2,1,2)(0,1,1)[12] : 5373.993  
## ARIMA(2,1,2)(1,1,0)[12] : 5417.897  
## ARIMA(2,1,3)(0,1,0)[12] : 5485.4  
## ARIMA(3,1,0)(0,1,0)[12] : 5482.401  
## ARIMA(3,1,0)(0,1,1)[12] : 5386.411  
## ARIMA(3,1,0)(0,1,2)[12] : 5388.436  
## ARIMA(3,1,0)(1,1,0)[12] : 5433.277  
## ARIMA(3,1,0)(1,1,1)[12] : 5400.236  
## ARIMA(3,1,0)(2,1,0)[12] : 5411.809  
## ARIMA(3,1,1)(0,1,0)[12] : 5473.457  
## ARIMA(3,1,1)(0,1,1)[12] : 5374.809  
## ARIMA(3,1,1)(1,1,0)[12] : 5433.627  
## ARIMA(3,1,2)(0,1,0)[12] : Inf  
## ARIMA(4,1,0)(0,1,0)[12] : 5483.521  
## ARIMA(4,1,0)(0,1,1)[12] : 5386.645  
## ARIMA(4,1,0)(1,1,0)[12] : 5434.578  
## ARIMA(4,1,1)(0,1,0)[12] : 5467.661  
## ARIMA(5,1,0)(0,1,0)[12] : 5485.461  
##   
## Now re-fitting the best model(s) without approximations...  
##   
##   
##   
##   
## Best model: ARIMA(2,1,1)(0,1,1)[12]

best.arima.exp.yeslambda <- auto.arima(train.exp.ts, stepwise = F, trace = T, lambda = "auto")

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(0,1,0) : 1005.913  
## ARIMA(0,1,0) with drift : 1006.716  
## ARIMA(0,1,0)(0,0,1)[12] : 964.2505  
## ARIMA(0,1,0)(0,0,1)[12] with drift : 965.444  
## ARIMA(0,1,0)(0,0,2)[12] : 948.0946  
## ARIMA(0,1,0)(0,0,2)[12] with drift : 949.5279  
## ARIMA(0,1,0)(1,0,0)[12] : 942.5781  
## ARIMA(0,1,0)(1,0,0)[12] with drift : 943.9747  
## ARIMA(0,1,0)(1,0,1)[12] : Inf  
## ARIMA(0,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(0,1,0)(1,0,2)[12] : Inf  
## ARIMA(0,1,0)(1,0,2)[12] with drift : Inf  
## ARIMA(0,1,0)(2,0,0)[12] : 931.6063  
## ARIMA(0,1,0)(2,0,0)[12] with drift : 933.0975  
## ARIMA(0,1,0)(2,0,1)[12] : Inf  
## ARIMA(0,1,0)(2,0,1)[12] with drift : Inf  
## ARIMA(0,1,0)(2,0,2)[12] : Inf  
## ARIMA(0,1,0)(2,0,2)[12] with drift : Inf  
## ARIMA(0,1,1) : 926.4344  
## ARIMA(0,1,1) with drift : 920.5922  
## ARIMA(0,1,1)(0,0,1)[12] : 875.8666  
## ARIMA(0,1,1)(0,0,1)[12] with drift : 872.3887  
## ARIMA(0,1,1)(0,0,2)[12] : 860.5826  
## ARIMA(0,1,1)(0,0,2)[12] with drift : 858.7202  
## ARIMA(0,1,1)(1,0,0)[12] : 854.4632  
## ARIMA(0,1,1)(1,0,0)[12] with drift : 853.0151  
## ARIMA(0,1,1)(1,0,1)[12] : 843.852  
## ARIMA(0,1,1)(1,0,1)[12] with drift : 844.5788  
## ARIMA(0,1,1)(1,0,2)[12] : Inf  
## ARIMA(0,1,1)(1,0,2)[12] with drift : Inf  
## ARIMA(0,1,1)(2,0,0)[12] : 847.3027  
## ARIMA(0,1,1)(2,0,0)[12] with drift : 846.2381  
## ARIMA(0,1,1)(2,0,1)[12] : Inf  
## ARIMA(0,1,1)(2,0,1)[12] with drift : Inf  
## ARIMA(0,1,1)(2,0,2)[12] : Inf  
## ARIMA(0,1,1)(2,0,2)[12] with drift : Inf  
## ARIMA(0,1,2) : 924.8555  
## ARIMA(0,1,2) with drift : 914.3282  
## ARIMA(0,1,2)(0,0,1)[12] : 876.6551  
## ARIMA(0,1,2)(0,0,1)[12] with drift : 871.4988  
## ARIMA(0,1,2)(0,0,2)[12] : 861.8264  
## ARIMA(0,1,2)(0,0,2)[12] with drift : 859.0215  
## ARIMA(0,1,2)(1,0,0)[12] : 855.2555  
## ARIMA(0,1,2)(1,0,0)[12] with drift : 852.6512  
## ARIMA(0,1,2)(1,0,1)[12] : 843.8751  
## ARIMA(0,1,2)(1,0,1)[12] with drift : 844.107  
## ARIMA(0,1,2)(1,0,2)[12] : Inf  
## ARIMA(0,1,2)(1,0,2)[12] with drift : Inf  
## ARIMA(0,1,2)(2,0,0)[12] : 847.1365  
## ARIMA(0,1,2)(2,0,0)[12] with drift : 844.9226  
## ARIMA(0,1,2)(2,0,1)[12] : Inf  
## ARIMA(0,1,2)(2,0,1)[12] with drift : Inf  
## ARIMA(0,1,3) : 921.3673  
## ARIMA(0,1,3) with drift : 907.0183  
## ARIMA(0,1,3)(0,0,1)[12] : 873.2791  
## ARIMA(0,1,3)(0,0,1)[12] with drift : 864.5193  
## ARIMA(0,1,3)(0,0,2)[12] : 856.3859  
## ARIMA(0,1,3)(0,0,2)[12] with drift : 851.0774  
## ARIMA(0,1,3)(1,0,0)[12] : 851.1582  
## ARIMA(0,1,3)(1,0,0)[12] with drift : 845.6509  
## ARIMA(0,1,3)(1,0,1)[12] : Inf  
## ARIMA(0,1,3)(1,0,1)[12] with drift : 833.6  
## ARIMA(0,1,3)(2,0,0)[12] : 841.8859  
## ARIMA(0,1,3)(2,0,0)[12] with drift : 837.9483  
## ARIMA(0,1,4) : 923.2897  
## ARIMA(0,1,4) with drift : 909.0778  
## ARIMA(0,1,4)(0,0,1)[12] : 875.3415  
## ARIMA(0,1,4)(0,0,1)[12] with drift : 866.3124  
## ARIMA(0,1,4)(1,0,0)[12] : 853.1654  
## ARIMA(0,1,4)(1,0,0)[12] with drift : 847.1304  
## ARIMA(0,1,5) : 925.1355  
## ARIMA(0,1,5) with drift : 910.4001  
## ARIMA(1,1,0) : 945.687  
## ARIMA(1,1,0) with drift : 944.7161  
## ARIMA(1,1,0)(0,0,1)[12] : 893.7141  
## ARIMA(1,1,0)(0,0,1)[12] with drift : 893.5373  
## ARIMA(1,1,0)(0,0,2)[12] : 874.5489  
## ARIMA(1,1,0)(0,0,2)[12] with drift : 874.9852  
## ARIMA(1,1,0)(1,0,0)[12] : 872.7078  
## ARIMA(1,1,0)(1,0,0)[12] with drift : 873.5478  
## ARIMA(1,1,0)(1,0,1)[12] : Inf  
## ARIMA(1,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(1,1,0)(1,0,2)[12] : Inf  
## ARIMA(1,1,0)(1,0,2)[12] with drift : Inf  
## ARIMA(1,1,0)(2,0,0)[12] : 866.9931  
## ARIMA(1,1,0)(2,0,0)[12] with drift : 867.8574  
## ARIMA(1,1,0)(2,0,1)[12] : Inf  
## ARIMA(1,1,0)(2,0,1)[12] with drift : Inf  
## ARIMA(1,1,0)(2,0,2)[12] : Inf  
## ARIMA(1,1,0)(2,0,2)[12] with drift : Inf  
## ARIMA(1,1,1) : 924.3898  
## ARIMA(1,1,1) with drift : 911.6609  
## ARIMA(1,1,1)(0,0,1)[12] : 877.4531  
## ARIMA(1,1,1)(0,0,1)[12] with drift : 872.379  
## ARIMA(1,1,1)(0,0,2)[12] : 862.3696  
## ARIMA(1,1,1)(0,0,2)[12] with drift : 859.6493  
## ARIMA(1,1,1)(1,0,0)[12] : 856.6573  
## ARIMA(1,1,1)(1,0,0)[12] with drift : 854.4225  
## ARIMA(1,1,1)(1,0,1)[12] : Inf  
## ARIMA(1,1,1)(1,0,1)[12] with drift : Inf  
## ARIMA(1,1,1)(1,0,2)[12] : Inf  
## ARIMA(1,1,1)(1,0,2)[12] with drift : Inf  
## ARIMA(1,1,1)(2,0,0)[12] : 845.9218  
## ARIMA(1,1,1)(2,0,0)[12] with drift : 842.569  
## ARIMA(1,1,1)(2,0,1)[12] : Inf  
## ARIMA(1,1,1)(2,0,1)[12] with drift : Inf  
## ARIMA(1,1,2) : 926.4281  
## ARIMA(1,1,2) with drift : 913.665  
## ARIMA(1,1,2)(0,0,1)[12] : 878.5337  
## ARIMA(1,1,2)(0,0,1)[12] with drift : 874.3145  
## ARIMA(1,1,2)(0,0,2)[12] : 862.0242  
## ARIMA(1,1,2)(0,0,2)[12] with drift : 860.3502  
## ARIMA(1,1,2)(1,0,0)[12] : 853.5156  
## ARIMA(1,1,2)(1,0,0)[12] with drift : 856.4186  
## ARIMA(1,1,2)(1,0,1)[12] : Inf  
## ARIMA(1,1,2)(1,0,1)[12] with drift : Inf  
## ARIMA(1,1,2)(2,0,0)[12] : 843.7047  
## ARIMA(1,1,2)(2,0,0)[12] with drift : Inf  
## ARIMA(1,1,3) : 923.0001  
## ARIMA(1,1,3) with drift : 908.3974  
## ARIMA(1,1,3)(0,0,1)[12] : 875.696  
## ARIMA(1,1,3)(0,0,1)[12] with drift : 867.811  
## ARIMA(1,1,3)(1,0,0)[12] : 855.016  
## ARIMA(1,1,3)(1,0,0)[12] with drift : 851.1549  
## ARIMA(1,1,4) : 924.937  
## ARIMA(1,1,4) with drift : 909.9566  
## ARIMA(2,1,0) : 940.3536  
## ARIMA(2,1,0) with drift : 938.3228  
## ARIMA(2,1,0)(0,0,1)[12] : 888.3648  
## ARIMA(2,1,0)(0,0,1)[12] with drift : 887.4363  
## ARIMA(2,1,0)(0,0,2)[12] : 871.9258  
## ARIMA(2,1,0)(0,0,2)[12] with drift : 871.8874  
## ARIMA(2,1,0)(1,0,0)[12] : 868.0646  
## ARIMA(2,1,0)(1,0,0)[12] with drift : 868.465  
## ARIMA(2,1,0)(1,0,1)[12] : Inf  
## ARIMA(2,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(2,1,0)(1,0,2)[12] : Inf  
## ARIMA(2,1,0)(1,0,2)[12] with drift : Inf  
## ARIMA(2,1,0)(2,0,0)[12] : 863.1562  
## ARIMA(2,1,0)(2,0,0)[12] with drift : 863.6452  
## ARIMA(2,1,0)(2,0,1)[12] : Inf  
## ARIMA(2,1,0)(2,0,1)[12] with drift : Inf  
## ARIMA(2,1,1) : 923.0001  
## ARIMA(2,1,1) with drift : 907.334  
## ARIMA(2,1,1)(0,0,1)[12] : 874.681  
## ARIMA(2,1,1)(0,0,1)[12] with drift : 863.2085  
## ARIMA(2,1,1)(0,0,2)[12] : 858.4751  
## ARIMA(2,1,1)(0,0,2)[12] with drift : 849.5829  
## ARIMA(2,1,1)(1,0,0)[12] : 850.6477  
## ARIMA(2,1,1)(1,0,0)[12] with drift : 840.1022  
## ARIMA(2,1,1)(1,0,1)[12] : Inf  
## ARIMA(2,1,1)(1,0,1)[12] with drift : Inf  
## ARIMA(2,1,1)(2,0,0)[12] : 843.4408  
## ARIMA(2,1,1)(2,0,0)[12] with drift : 838.1644  
## ARIMA(2,1,2) : 924.9239  
## ARIMA(2,1,2) with drift : 909.38  
## ARIMA(2,1,2)(0,0,1)[12] : 875.3814  
## ARIMA(2,1,2)(0,0,1)[12] with drift : 863.9633  
## ARIMA(2,1,2)(1,0,0)[12] : 852.4738  
## ARIMA(2,1,2)(1,0,0)[12] with drift : 841.5905  
## ARIMA(2,1,3) : 915.0312  
## ARIMA(2,1,3) with drift : 910.122  
## ARIMA(3,1,0) : 932.4606  
## ARIMA(3,1,0) with drift : 928.9403  
## ARIMA(3,1,0)(0,0,1)[12] : 882.4376  
## ARIMA(3,1,0)(0,0,1)[12] with drift : 880.6706  
## ARIMA(3,1,0)(0,0,2)[12] : 867.3229  
## ARIMA(3,1,0)(0,0,2)[12] with drift : 866.5499  
## ARIMA(3,1,0)(1,0,0)[12] : 863.6745  
## ARIMA(3,1,0)(1,0,0)[12] with drift : 863.499  
## ARIMA(3,1,0)(1,0,1)[12] : Inf  
## ARIMA(3,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(3,1,0)(2,0,0)[12] : 845.3769  
## ARIMA(3,1,0)(2,0,0)[12] with drift : 845.7701  
## ARIMA(3,1,1) : 923.2987  
## ARIMA(3,1,1) with drift : 908.4638  
## ARIMA(3,1,1)(0,0,1)[12] : 873.7247  
## ARIMA(3,1,1)(0,0,1)[12] with drift : 863.1132  
## ARIMA(3,1,1)(1,0,0)[12] : 855.0627  
## ARIMA(3,1,1)(1,0,0)[12] with drift : 845.4102  
## ARIMA(3,1,2) : 915.9593  
## ARIMA(3,1,2) with drift : 910.5355  
## ARIMA(4,1,0) : 931.6533  
## ARIMA(4,1,0) with drift : 927.2834  
## ARIMA(4,1,0)(0,0,1)[12] : 880.4138  
## ARIMA(4,1,0)(0,0,1)[12] with drift : 878.1754  
## ARIMA(4,1,0)(1,0,0)[12] : 860.5466  
## ARIMA(4,1,0)(1,0,0)[12] with drift : 859.6467  
## ARIMA(4,1,1) : 925.8326  
## ARIMA(4,1,1) with drift : 912.5626  
## ARIMA(5,1,0) : 934.2594  
## ARIMA(5,1,0) with drift : 929.282  
##   
## Now re-fitting the best model(s) without approximations...  
##   
## ARIMA(0,1,0) : 1005.635  
## ARIMA(0,1,0) with drift : 1006.439  
## ARIMA(0,1,0)(0,0,1)[12] : 963.5773  
## ARIMA(0,1,0)(0,0,1)[12] with drift : 964.7716  
## ARIMA(0,1,0)(0,0,2)[12] : 947.8946  
## ARIMA(0,1,0)(0,0,2)[12] with drift : 949.3054  
## ARIMA(0,1,0)(1,0,0)[12] : 944.5969  
## ARIMA(0,1,0)(1,0,0)[12] with drift : 946.0834  
## ARIMA(0,1,0)(1,0,1)[12] : Inf  
## ARIMA(0,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(0,1,0)(1,0,2)[12] : Inf  
## ARIMA(0,1,0)(1,0,2)[12] with drift : Inf  
## ARIMA(0,1,0)(2,0,0)[12] : 932.0587  
## ARIMA(0,1,0)(2,0,0)[12] with drift : 933.7485  
## ARIMA(0,1,0)(2,0,1)[12] : Inf  
## ARIMA(0,1,0)(2,0,1)[12] with drift : Inf  
## ARIMA(0,1,0)(2,0,2)[12] : Inf  
## ARIMA(0,1,0)(2,0,2)[12] with drift : Inf  
## ARIMA(0,1,1) : 925.9706  
## ARIMA(0,1,1) with drift : 919.9582  
## ARIMA(0,1,1)(0,0,1)[12] : 875.7383  
## ARIMA(0,1,1)(0,0,1)[12] with drift : 872.3622  
## ARIMA(0,1,1)(0,0,2)[12] : 861.0172  
## ARIMA(0,1,1)(0,0,2)[12] with drift : 859.1578  
## ARIMA(0,1,1)(1,0,0)[12] : 854.6893  
## ARIMA(0,1,1)(1,0,0)[12] with drift : 853.3982  
## ARIMA(0,1,1)(1,0,1)[12] : Inf  
## ARIMA(0,1,1)(1,0,1)[12] with drift : Inf  
## ARIMA(0,1,1)(1,0,2)[12] : Inf  
## ARIMA(0,1,1)(1,0,2)[12] with drift : Inf  
## ARIMA(0,1,1)(2,0,0)[12] : 845.9842  
## ARIMA(0,1,1)(2,0,0)[12] with drift : 845.793  
## ARIMA(0,1,1)(2,0,1)[12] : Inf  
## ARIMA(0,1,1)(2,0,1)[12] with drift : Inf  
## ARIMA(0,1,1)(2,0,2)[12] : Inf  
## ARIMA(0,1,1)(2,0,2)[12] with drift : Inf  
## ARIMA(0,1,2) : 924.583  
## ARIMA(0,1,2) with drift : 913.9565  
## ARIMA(0,1,2)(0,0,1)[12] : 876.4865  
## ARIMA(0,1,2)(0,0,1)[12] with drift : 871.4559  
## ARIMA(0,1,2)(0,0,2)[12] : 862.237  
## ARIMA(0,1,2)(0,0,2)[12] with drift : 859.4397  
## ARIMA(0,1,2)(1,0,0)[12] : 855.9779  
## ARIMA(0,1,2)(1,0,0)[12] with drift : 853.9528  
## ARIMA(0,1,2)(1,0,1)[12] : Inf  
## ARIMA(0,1,2)(1,0,1)[12] with drift : Inf  
## ARIMA(0,1,2)(1,0,2)[12] : Inf  
## ARIMA(0,1,2)(1,0,2)[12] with drift : Inf  
## ARIMA(0,1,2)(2,0,0)[12] : 846.8858  
## ARIMA(0,1,2)(2,0,0)[12] with drift : 846.115  
## ARIMA(0,1,2)(2,0,1)[12] : Inf  
## ARIMA(0,1,2)(2,0,1)[12] with drift : Inf  
## ARIMA(0,1,3) : 921.2603  
## ARIMA(0,1,3) with drift : 906.733  
## ARIMA(0,1,3)(0,0,1)[12] : 873.3632  
## ARIMA(0,1,3)(0,0,1)[12] with drift : 864.8109  
## ARIMA(0,1,3)(0,0,2)[12] : 857.237  
## ARIMA(0,1,3)(0,0,2)[12] with drift : 851.9678  
## ARIMA(0,1,3)(1,0,0)[12] : 851.7665  
## ARIMA(0,1,3)(1,0,0)[12] with drift : 847.4677  
## ARIMA(0,1,3)(1,0,1)[12] : Inf  
## ARIMA(0,1,3)(1,0,1)[12] with drift : Inf  
## ARIMA(0,1,3)(2,0,0)[12] : 840.58  
## ARIMA(0,1,3)(2,0,0)[12] with drift : 838.3199  
## ARIMA(0,1,4) : 923.1803  
## ARIMA(0,1,4) with drift : 908.7896  
## ARIMA(0,1,4)(0,0,1)[12] : 875.4257  
## ARIMA(0,1,4)(0,0,1)[12] with drift : 866.5939  
## ARIMA(0,1,4)(1,0,0)[12] : 853.7925  
## ARIMA(0,1,4)(1,0,0)[12] with drift : 849.1639  
## ARIMA(0,1,5) : 925.0141  
## ARIMA(0,1,5) with drift : 910.0205  
## ARIMA(1,1,0) : 946.5716  
## ARIMA(1,1,0) with drift : 945.7787  
## ARIMA(1,1,0)(0,0,1)[12] : 895.553  
## ARIMA(1,1,0)(0,0,1)[12] with drift : 895.5777  
## ARIMA(1,1,0)(0,0,2)[12] : 876.8438  
## ARIMA(1,1,0)(0,0,2)[12] with drift : 877.3866  
## ARIMA(1,1,0)(1,0,0)[12] : 872.3642  
## ARIMA(1,1,0)(1,0,0)[12] with drift : 873.1381  
## ARIMA(1,1,0)(1,0,1)[12] : Inf  
## ARIMA(1,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(1,1,0)(1,0,2)[12] : Inf  
## ARIMA(1,1,0)(1,0,2)[12] with drift : Inf  
## ARIMA(1,1,0)(2,0,0)[12] : 862.0686  
## ARIMA(1,1,0)(2,0,0)[12] with drift : 863.2717  
## ARIMA(1,1,0)(2,0,1)[12] : Inf  
## ARIMA(1,1,0)(2,0,1)[12] with drift : Inf  
## ARIMA(1,1,0)(2,0,2)[12] : Inf  
## ARIMA(1,1,0)(2,0,2)[12] with drift : Inf  
## ARIMA(1,1,1) : 922.6684  
## ARIMA(1,1,1) with drift : 908.9357  
## ARIMA(1,1,1)(0,0,1)[12] : 875.5957  
## ARIMA(1,1,1)(0,0,1)[12] with drift : 868.365  
## ARIMA(1,1,1)(0,0,2)[12] : 861.4558  
## ARIMA(1,1,1)(0,0,2)[12] with drift : 857.2722  
## ARIMA(1,1,1)(1,0,0)[12] : 855.348  
## ARIMA(1,1,1)(1,0,0)[12] with drift : 852.2288  
## ARIMA(1,1,1)(1,0,1)[12] : Inf  
## ARIMA(1,1,1)(1,0,1)[12] with drift : Inf  
## ARIMA(1,1,1)(1,0,2)[12] : Inf  
## ARIMA(1,1,1)(1,0,2)[12] with drift : Inf  
## ARIMA(1,1,1)(2,0,0)[12] : 845.6577  
## ARIMA(1,1,1)(2,0,0)[12] with drift : 843.9622  
## ARIMA(1,1,1)(2,0,1)[12] : Inf  
## ARIMA(1,1,1)(2,0,1)[12] with drift : Inf  
## ARIMA(1,1,2) : 923.1921  
## ARIMA(1,1,2) with drift : 908.2262  
## ARIMA(1,1,2)(0,0,1)[12] : Inf  
## ARIMA(1,1,2)(0,0,1)[12] with drift : 863.2577  
## ARIMA(1,1,2)(0,0,2)[12] : 864.6295  
## ARIMA(1,1,2)(0,0,2)[12] with drift : 862.9728  
## ARIMA(1,1,2)(1,0,0)[12] : Inf  
## ARIMA(1,1,2)(1,0,0)[12] with drift : 844.0538  
## ARIMA(1,1,2)(1,0,1)[12] : Inf  
## ARIMA(1,1,2)(1,0,1)[12] with drift : Inf  
## ARIMA(1,1,2)(2,0,0)[12] : 842.8409  
## ARIMA(1,1,2)(2,0,0)[12] with drift : 836.1863  
## ARIMA(1,1,3) : 923.152  
## ARIMA(1,1,3) with drift : 908.7782  
## ARIMA(1,1,3)(0,0,1)[12] : 875.4255  
## ARIMA(1,1,3)(0,0,1)[12] with drift : 864.4384  
## ARIMA(1,1,3)(1,0,0)[12] : 853.5819  
## ARIMA(1,1,3)(1,0,0)[12] with drift : 845.16  
## ARIMA(1,1,4) : 925.2152  
## ARIMA(1,1,4) with drift : 910.8684  
## ARIMA(2,1,0) : 940.3563  
## ARIMA(2,1,0) with drift : 938.5721  
## ARIMA(2,1,0)(0,0,1)[12] : 889.2314  
## ARIMA(2,1,0)(0,0,1)[12] with drift : 888.5576  
## ARIMA(2,1,0)(0,0,2)[12] : 873.0251  
## ARIMA(2,1,0)(0,0,2)[12] with drift : 873.0985  
## ARIMA(2,1,0)(1,0,0)[12] : 867.1666  
## ARIMA(2,1,0)(1,0,0)[12] with drift : 867.5266  
## ARIMA(2,1,0)(1,0,1)[12] : Inf  
## ARIMA(2,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(2,1,0)(1,0,2)[12] : Inf  
## ARIMA(2,1,0)(1,0,2)[12] with drift : Inf  
## ARIMA(2,1,0)(2,0,0)[12] : 858.2425  
## ARIMA(2,1,0)(2,0,0)[12] with drift : 859.1749  
## ARIMA(2,1,0)(2,0,1)[12] : Inf  
## ARIMA(2,1,0)(2,0,1)[12] with drift : Inf  
## ARIMA(2,1,1) : 922.2742  
## ARIMA(2,1,1) with drift : 907.1035  
## ARIMA(2,1,1)(0,0,1)[12] : 873.3486  
## ARIMA(2,1,1)(0,0,1)[12] with drift : 861.985  
## ARIMA(2,1,1)(0,0,2)[12] : 857.4357  
## ARIMA(2,1,1)(0,0,2)[12] with drift : 848.774  
## ARIMA(2,1,1)(1,0,0)[12] : 851.0904  
## ARIMA(2,1,1)(1,0,0)[12] with drift : 842.713  
## ARIMA(2,1,1)(1,0,1)[12] : Inf  
## ARIMA(2,1,1)(1,0,1)[12] with drift : Inf  
## ARIMA(2,1,1)(2,0,0)[12] : 840.2399  
## ARIMA(2,1,1)(2,0,0)[12] with drift : 834.5298  
## ARIMA(2,1,2) : 923.3633  
## ARIMA(2,1,2) with drift : 908.3372  
## ARIMA(2,1,2)(0,0,1)[12] : 875.0889  
## ARIMA(2,1,2)(0,0,1)[12] with drift : 864.0539  
## ARIMA(2,1,2)(1,0,0)[12] : 853.0445  
## ARIMA(2,1,2)(1,0,0)[12] with drift : 844.6933  
## ARIMA(2,1,3) : Inf  
## ARIMA(2,1,3) with drift : 895.8837  
## ARIMA(3,1,0) : 931.6611  
## ARIMA(3,1,0) with drift : 928.3333  
## ARIMA(3,1,0)(0,0,1)[12] : 882.6124  
## ARIMA(3,1,0)(0,0,1)[12] with drift : 880.9631  
## ARIMA(3,1,0)(0,0,2)[12] : 867.3651  
## ARIMA(3,1,0)(0,0,2)[12] with drift : 866.7446  
## ARIMA(3,1,0)(1,0,0)[12] : 861.8264  
## ARIMA(3,1,0)(1,0,0)[12] with drift : 861.5865  
## ARIMA(3,1,0)(1,0,1)[12] : Inf  
## ARIMA(3,1,0)(1,0,1)[12] with drift : Inf  
## ARIMA(3,1,0)(2,0,0)[12] : 853.1722  
## ARIMA(3,1,0)(2,0,0)[12] with drift : 853.6958  
## ARIMA(3,1,1) : 923.2904  
## ARIMA(3,1,1) with drift : 908.3891  
## ARIMA(3,1,1)(0,0,1)[12] : 875.1384  
## ARIMA(3,1,1)(0,0,1)[12] with drift : 864.0559  
## ARIMA(3,1,1)(1,0,0)[12] : 853.0644  
## ARIMA(3,1,1)(1,0,0)[12] with drift : 844.7305  
## ARIMA(3,1,2) : 917.8095  
## ARIMA(3,1,2) with drift : 910.3854  
## ARIMA(4,1,0) : 930.8258  
## ARIMA(4,1,0) with drift : 926.4372  
## ARIMA(4,1,0)(0,0,1)[12] : 881.3597  
## ARIMA(4,1,0)(0,0,1)[12] with drift : 878.9569  
## ARIMA(4,1,0)(1,0,0)[12] : 859.8049  
## ARIMA(4,1,0)(1,0,0)[12] with drift : 859.0839  
## ARIMA(4,1,1) : 925.3271  
## ARIMA(4,1,1) with drift : 910.4221  
## ARIMA(5,1,0) : 932.4786  
## ARIMA(5,1,0) with drift : 927.5646  
##   
##   
##   
##   
##   
## Best model: ARIMA(2,1,1)(2,0,0)[12] with drift

summary(best.arima.exp.nolambda)

## Series: train.exp.ts   
## ARIMA(2,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ar1 ar2 ma1 sma1  
## 0.7114 0.1283 -0.9646 -0.5424  
## s.e. 0.0574 0.0559 0.0268 0.0442  
##   
## sigma^2 estimated as 119334: log likelihood=-2760.92  
## AIC=5531.83 AICc=5531.99 BIC=5551.53  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 11.03621 337.893 227.3651 -0.4317361 11.35756 0.4513094  
## ACF1  
## Training set 0.0001588779

summary(best.arima.exp.yeslambda)

## Series: train.exp.ts   
## ARIMA(2,1,1)(2,0,0)[12] with drift   
## Box Cox transformation: lambda= 0.220129   
##   
## Coefficients:  
## ar1 ar2 ma1 sar1 sar2 drift  
## 0.3991 0.2344 -0.9395 0.3487 0.1650 0.0480  
## s.e. 0.0606 0.0582 0.0312 0.0518 0.0511 0.0117  
##   
## sigma^2 estimated as 0.4778: log likelihood=-410.12  
## AIC=834.24 AICc=834.53 BIC=862.04  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 10.36284 402.0039 240.6093 -1.331049 10.81068 0.4775985  
## ACF1  
## Training set 0.2891842

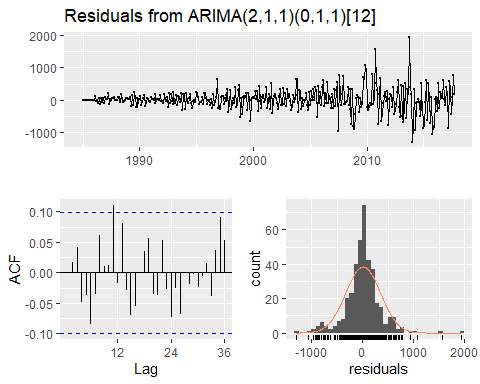
accuracy(forecast(best.arima.exp.nolambda),exp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 11.03621 337.893 227.3651 -0.4317361 11.35756 0.4513094  
## Test set -1482.88331 2239.336 1770.2659 -17.4212900 19.70742 3.5138966  
## ACF1 Theil's U  
## Training set 0.0001588779 NA  
## Test set 0.6795136848 1.767939

accuracy(forecast(best.arima.exp.yeslambda),exp.ts)

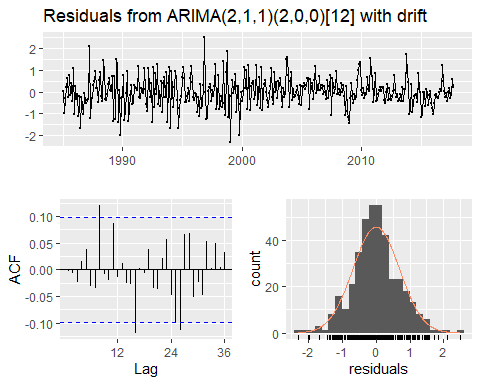
## ME RMSE MAE MPE MAPE MASE  
## Training set 10.36284 402.0039 240.6093 -1.331049 10.81068 0.4775985  
## Test set -1610.97316 2434.7656 2071.0324 -19.342475 22.87962 4.1109044  
## ACF1 Theil's U  
## Training set 0.2891842 NA  
## Test set 0.6678145 1.93414

checkresiduals(best.arima.exp.nolambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1)(0,1,1)[12]  
## Q\* = 25.256, df = 20, p-value = 0.1918  
##   
## Model df: 4. Total lags used: 24

checkresiduals(best.arima.exp.yeslambda)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1)(2,0,0)[12] with drift  
## Q\* = 21.647, df = 18, p-value = 0.248  
##   
## Model df: 6. Total lags used: 24

#non-linear testing  
nonlinearityTest(train.imp.ts)

## \*\* Teraesvirta's neural network test \*\*  
## Null hypothesis: Linearity in "mean"   
## X-squared = 14.54697 df = 2 p-value = 0.0006936909   
##   
## \*\* White neural network test \*\*  
## Null hypothesis: Linearity in "mean"   
## X-squared = 14.3498 df = 2 p-value = 0.000765562   
##   
## \*\* Keenan's one-degree test for nonlinearity \*\*  
## Null hypothesis: The time series follows some AR process  
## F-stat = 20.80022 p-value = 7.11869e-06   
##   
## \*\* McLeod-Li test \*\*  
## Null hypothesis: The time series follows some ARIMA process  
## Maximum p-value = 0   
##   
## \*\* Tsay's Test for nonlinearity \*\*  
## Null hypothesis: The time series follows some AR process  
## F-stat = 110.6 p-value = 3.2e-15   
##   
## \*\* Likelihood ratio test for threshold nonlinearity \*\*  
## Null hypothesis: The time series follows some AR process  
## Alternativce hypothesis: The time series follows some TAR process  
## X-squared = 111.4864 p-value = 1.121134e-10

## $Terasvirta  
##   
## Teraesvirta Neural Network Test  
##   
## data: ts(time.series)  
## X-squared = 14.547, df = 2, p-value = 0.0006937  
##   
##   
## $White  
##   
## White Neural Network Test  
##   
## data: ts(time.series)  
## X-squared = 14.35, df = 2, p-value = 0.0007656  
##   
##   
## $Keenan  
## $Keenan$test.stat  
## [1] 20.80022  
##   
## $Keenan$p.value  
## [1] 7.11869e-06  
##   
## $Keenan$order  
## [1] 25  
##   
##   
## $McLeodLi  
## $McLeodLi$p.values  
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
##   
##   
## $Tsay  
## $Tsay$test.stat  
## [1] 110.6  
##   
## $Tsay$p.value  
## [1] 3.2e-15  
##   
## $Tsay$order  
## [1] 25  
##   
##   
## $TarTest  
## $TarTest$percentiles  
## [1] 25 75  
##   
## $TarTest$test.statistic  
##   
## 111.4864   
##   
## $TarTest$p.value  
##   
## 1.121134e-10

nonlinearityTest(train.exp.ts)

## \*\* Teraesvirta's neural network test \*\*  
## Null hypothesis: Linearity in "mean"   
## X-squared = 39.29675 df = 2 p-value = 2.929673e-09   
##   
## \*\* White neural network test \*\*  
## Null hypothesis: Linearity in "mean"   
## X-squared = 33.46516 df = 2 p-value = 5.409199e-08   
##   
## \*\* Keenan's one-degree test for nonlinearity \*\*  
## Null hypothesis: The time series follows some AR process  
## F-stat = 6.013017 p-value = 0.01470146   
##   
## \*\* McLeod-Li test \*\*  
## Null hypothesis: The time series follows some ARIMA process  
## Maximum p-value = 0   
##   
## \*\* Tsay's Test for nonlinearity \*\*  
## Null hypothesis: The time series follows some AR process  
## F-stat = 9.493 p-value = 1.722e-06   
##   
## \*\* Likelihood ratio test for threshold nonlinearity \*\*  
## Null hypothesis: The time series follows some AR process  
## Alternativce hypothesis: The time series follows some TAR process  
## X-squared = 62.49455 p-value = 0.002001879

## $Terasvirta  
##   
## Teraesvirta Neural Network Test  
##   
## data: ts(time.series)  
## X-squared = 39.297, df = 2, p-value = 2.93e-09  
##   
##   
## $White  
##   
## White Neural Network Test  
##   
## data: ts(time.series)  
## X-squared = 33.465, df = 2, p-value = 5.409e-08  
##   
##   
## $Keenan  
## $Keenan$test.stat  
## [1] 6.013017  
##   
## $Keenan$p.value  
## [1] 0.01470146  
##   
## $Keenan$order  
## [1] 25  
##   
##   
## $McLeodLi  
## $McLeodLi$p.values  
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
##   
##   
## $Tsay  
## $Tsay$test.stat  
## [1] 9.493  
##   
## $Tsay$p.value  
## [1] 1.722e-06  
##   
## $Tsay$order  
## [1] 25  
##   
##   
## $TarTest  
## $TarTest$percentiles  
## [1] 25 75  
##   
## $TarTest$test.statistic  
##   
## 62.49455   
##   
## $TarTest$p.value  
##   
## 0.002001879

#neural networks  
set.seed(100)  
neural.imp <- nnetar(train.imp.ts)  
neural.imp

## Series: train.imp.ts   
## Model: NNAR(4,1,3)[12]   
## Call: nnetar(y = train.imp.ts)  
##   
## Average of 20 networks, each of which is  
## a 5-3-1 network with 22 weights  
## options were - linear output units   
##   
## sigma^2 estimated as 1618448

accuracy(forecast(neural.imp),imp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 1.004195 1272.183 746.6791 -2.172185 7.23142 0.4292306  
## Test set 2113.309300 3970.815 3526.2591 4.157140 8.17489 2.0270800  
## ACF1 Theil's U  
## Training set 0.1262050 NA  
## Test set 0.6928315 1.100258

BoxCox.lambda(train.imp.ts)

## [1] 0.1637399

set.seed(100)  
neural.imp.boxcox <- nnetar(train.imp.ts, lambda = BoxCox.lambda(train.imp.ts))  
neural.imp.boxcox

## Series: train.imp.ts   
## Model: NNAR(1,1,2)[12]   
## Call: nnetar(y = train.imp.ts, lambda = BoxCox.lambda(train.imp.ts))  
##   
## Average of 20 networks, each of which is  
## a 2-2-1 network with 9 weights  
## options were - linear output units   
##   
## sigma^2 estimated as 0.1319

accuracy(forecast(neural.imp.boxcox),imp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 39.81596 1514.976 884.1333 -0.4315061 6.597035 0.5082465  
## Test set 1795.17829 4444.743 3615.2595 3.0381427 8.485026 2.0782421  
## ACF1 Theil's U  
## Training set 0.1137020 NA  
## Test set 0.7510175 1.264006

set.seed(100)  
neural.exp <- nnetar(train.exp.ts)  
neural.exp

## Series: train.exp.ts   
## Model: NNAR(1,1,2)[12]   
## Call: nnetar(y = train.exp.ts)  
##   
## Average of 20 networks, each of which is  
## a 2-2-1 network with 9 weights  
## options were - linear output units   
##   
## sigma^2 estimated as 195589

accuracy(forecast(neural.exp),exp.ts)

## ME RMSE MAE MPE MAPE  
## Training set 0.2292767 442.2549 278.0997 -3.358741 12.53733  
## Test set -1196.3518931 2091.7454 1769.4220 -14.909501 19.45795  
## MASE ACF1 Theil's U  
## Training set 0.5520151 0.2018676 NA  
## Test set 3.5122214 0.6016065 1.652535

BoxCox.lambda(train.exp.ts)

## [1] 0.220129

set.seed(100)  
neural.exp.boxcox <- nnetar(train.exp.ts, lambda = BoxCox.lambda(train.exp.ts))  
neural.exp.boxcox

## Series: train.exp.ts   
## Model: NNAR(2,1,2)[12]   
## Call: nnetar(y = train.exp.ts, lambda = BoxCox.lambda(train.exp.ts))  
##   
## Average of 20 networks, each of which is  
## a 3-2-1 network with 11 weights  
## options were - linear output units   
##   
## sigma^2 estimated as 0.5098

accuracy(forecast(neural.exp.boxcox),exp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 15.52981 457.1013 270.3462 -1.369548 11.04153 0.5366247  
## Test set -789.80332 1744.3656 1487.8390 -10.532445 16.06305 2.9532921  
## ACF1 Theil's U  
## Training set 0.2351623 NA  
## Test set 0.5333260 1.342548

set.seed(100)  
imp.ts <- ts(data = imp[[1]], start = c(1985,1), end = c(2019,9), frequency = 12)  
train.imp.ts <- window(imp.ts, end=end(imp.ts)-c(2,0))  
bag.imp <- baggedModel(train.imp.ts, bootstrapped\_series = bld.mbb.bootstrap(train.imp.ts, 100))  
accuracy(forecast(bag.imp), imp.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 9.890236 1001.747 598.2268 0.1107069 5.215448 0.3438924  
## Test set 69.670790 4186.723 3566.8610 -0.5733451 8.673755 2.0504201  
## ACF1 Theil's U  
## Training set 0.1788646 NA  
## Test set 0.7304160 1.24933

set.seed(100)  
bag.imp.wlambda <- baggedModel(BoxCox(train.imp.ts,lambda = BoxCox.lambda(train.imp.ts)), bootstrapped\_series = bld.mbb.bootstrap(train.imp.ts, 100))  
accuracy(forecast(bag.imp.wlambda), BoxCox(imp.ts,lambda = BoxCox.lambda(train.imp.ts)))

## ME RMSE MAE MPE MAPE MASE  
## Training set -15365.52 21156.07 15365.52 -60220.1 60220.1 22417.89  
## Test set -42498.90 42738.15 42498.90 -147209.9 147209.9 62004.77  
## ACF1 Theil's U  
## Training set 0.9869763 NA  
## Test set 0.6993608 87036.79

set.seed(100)  
exp.ts <- ts(data = exp[[1]], start = c(1985,1), end = c(2019,9), frequency = 12)  
train.exp.ts <- window(exp.ts, end=end(exp.ts)-c(2,0))  
bag.exp <- baggedModel(train.exp.ts, bootstrapped\_series = bld.mbb.bootstrap(train.exp.ts, 100, block\_size = NULL))  
accuracy(forecast(bag.exp), exp.ts)

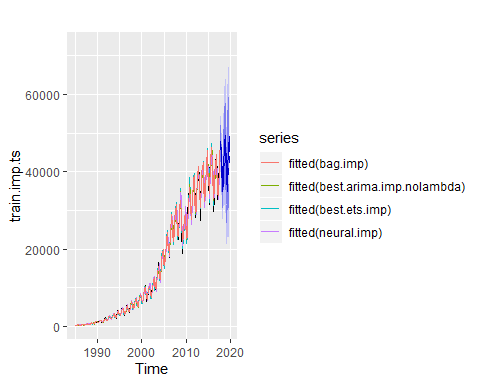
## ME RMSE MAE MPE MAPE MASE  
## Training set 42.11229 391.9716 239.6898 -0.3486852 9.43660 0.4757732  
## Test set -764.05031 1642.8201 1399.6392 -9.9447899 15.04068 2.7782196  
## ACF1 Theil's U  
## Training set 0.5771892 NA  
## Test set 0.7268391 1.264623

set.seed(100)  
bag.exp.wlambda <- baggedModel(BoxCox(train.exp.ts,lambda = BoxCox.lambda(train.exp.ts)), bootstrapped\_series = bld.mbb.bootstrap(train.exp.ts, 100, block\_size = NULL))  
accuracy(forecast(bag.exp.wlambda), BoxCox(exp.ts,lambda = BoxCox.lambda(train.exp.ts)))

## ME RMSE MAE MPE MAPE MASE  
## Training set -3533.075 5038.574 3533.075 -13916.27 13916.27 3543.666  
## Test set -10659.701 10680.271 10659.701 -35757.59 35757.59 10691.656  
## ACF1 Theil's U  
## Training set 0.98596726 NA  
## Test set 0.06434909 9805.401

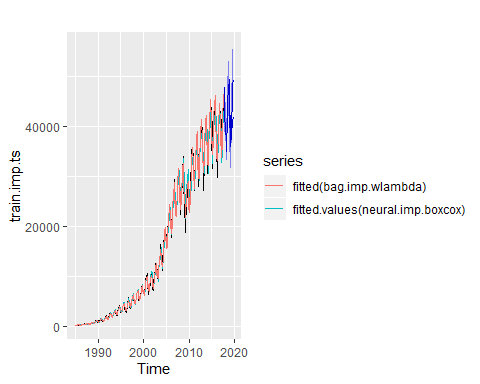
#graphical comparison  
autoplot(train.imp.ts)+autolayer(fitted(best.ets.imp))+autolayer(fitted(best.arima.imp.nolambda))+autolayer(fitted(neural.imp))+autolayer(fitted(bag.imp))+autolayer(forecast(best.ets.imp))+autolayer(forecast(best.arima.imp.nolambda))+autolayer(forecast(neural.imp))+autolayer(forecast(bag.imp))

## Warning: Removed 12 rows containing missing values (geom\_path).

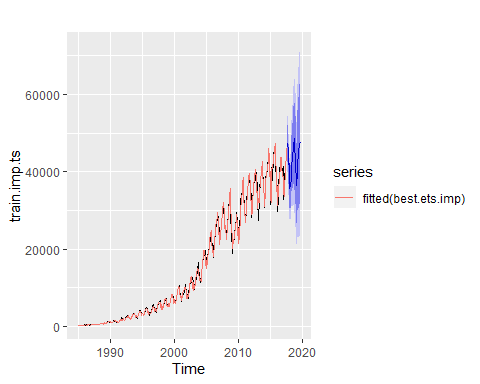


autoplot(train.imp.ts)+autolayer(fitted.values(neural.imp.boxcox))+autolayer(fitted(bag.imp.wlambda))+autolayer(forecast(neural.imp.boxcox))+autolayer(forecast(bag.imp.wlambda))

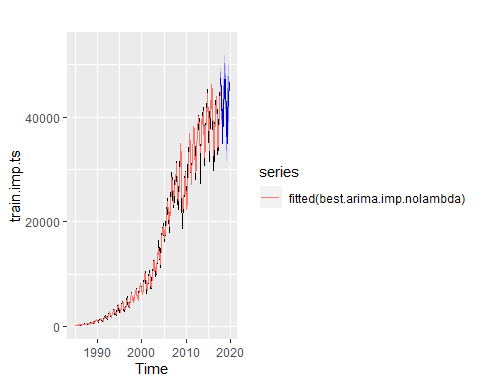
## Warning: Removed 12 rows containing missing values (geom\_path).



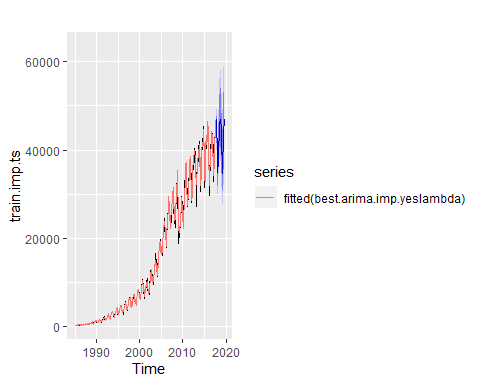
autoplot(train.imp.ts)+autolayer(fitted(best.ets.imp))+autolayer(forecast(best.ets.imp))



autoplot(train.imp.ts)+autolayer(fitted(best.arima.imp.nolambda))+autolayer(forecast(best.arima.imp.nolambda))

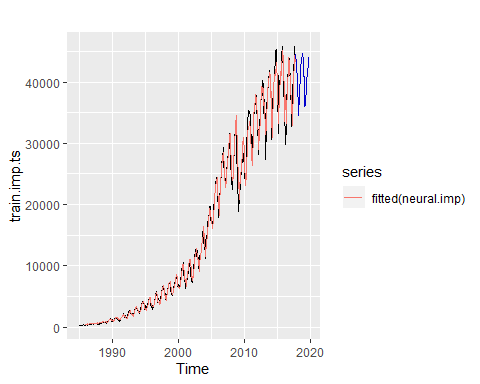


autoplot(train.imp.ts)+autolayer(fitted(best.arima.imp.yeslambda))+autolayer(forecast(best.arima.imp.yeslambda))



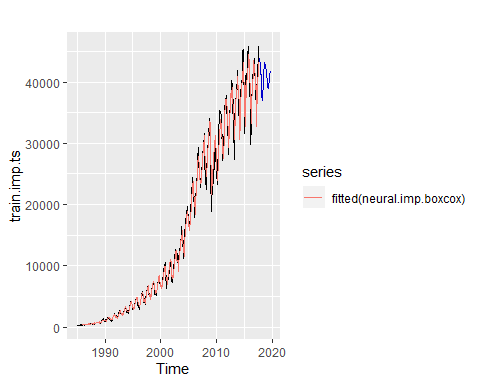
autoplot(train.imp.ts)+autolayer(fitted(neural.imp))+autolayer(forecast(neural.imp))

## Warning: Removed 12 rows containing missing values (geom\_path).

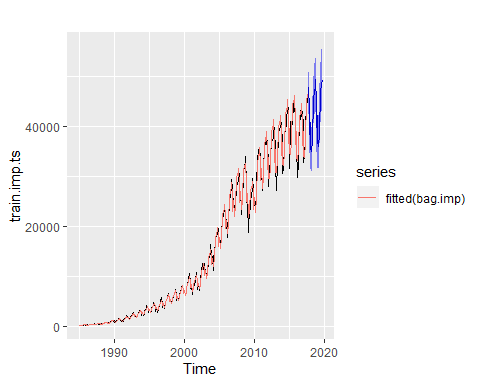


autoplot(train.imp.ts)+autolayer(fitted(neural.imp.boxcox))+autolayer(forecast(neural.imp.boxcox))

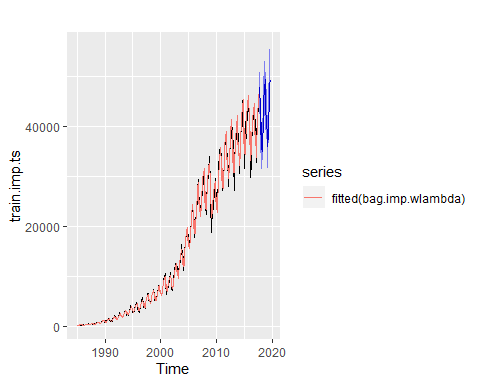
## Warning: Removed 12 rows containing missing values (geom\_path).



autoplot(train.imp.ts)+autolayer(fitted(bag.imp))+autolayer(forecast(bag.imp))

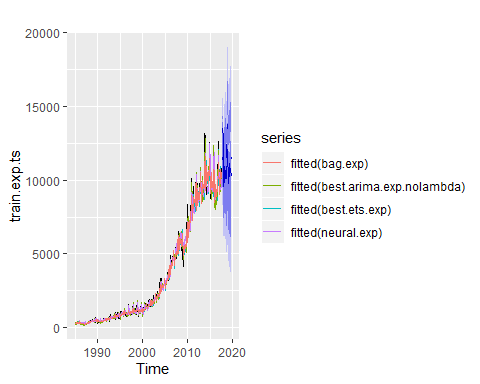


autoplot(train.imp.ts)+autolayer(fitted(bag.imp.wlambda))+autolayer(forecast(bag.imp.wlambda))



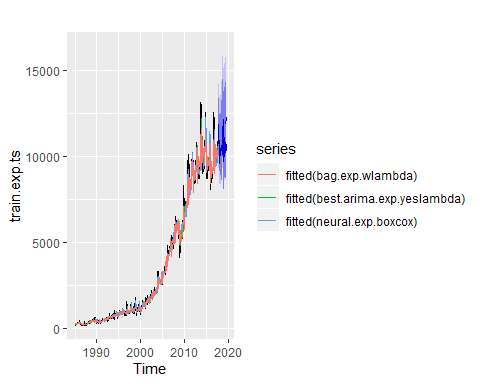
autoplot(train.exp.ts)+autolayer(fitted(best.ets.exp))+autolayer(fitted(best.arima.exp.nolambda))+autolayer(fitted(neural.exp))+autolayer(fitted(bag.exp))+autolayer(forecast(best.ets.exp))+autolayer(forecast(best.arima.exp.nolambda))+autolayer(forecast(neural.exp))+autolayer(forecast(bag.exp))

## Warning: Removed 12 rows containing missing values (geom\_path).

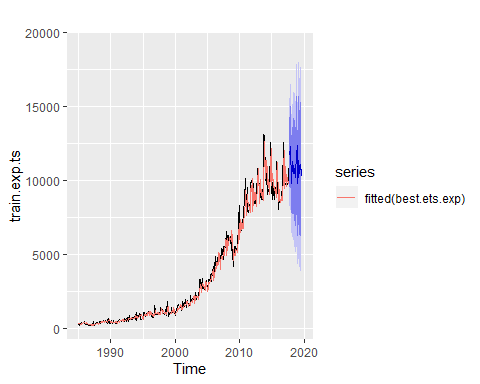


autoplot(train.exp.ts)+autolayer(fitted(best.arima.exp.yeslambda))+autolayer(fitted(neural.exp.boxcox))+autolayer(fitted(bag.exp.wlambda))+autolayer(forecast(best.arima.exp.yeslambda))+autolayer(forecast(neural.exp.boxcox))+autolayer(forecast(bag.exp.wlambda))

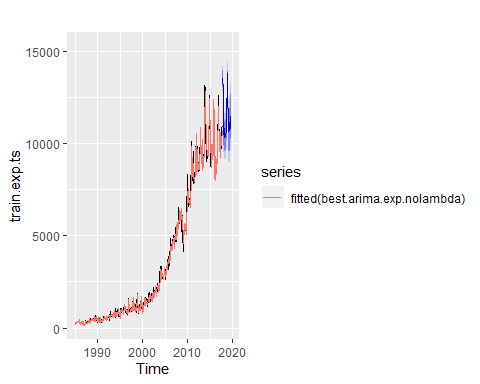
## Warning: Removed 12 rows containing missing values (geom\_path).



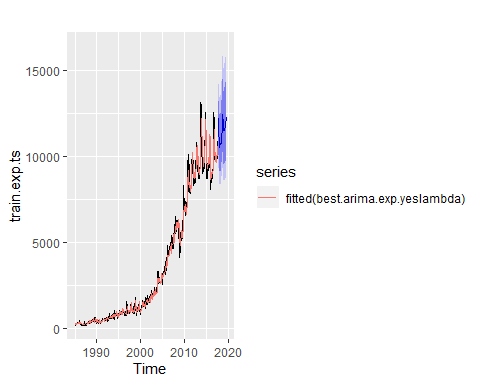
autoplot(train.exp.ts)+autolayer(fitted(best.ets.exp))+autolayer(forecast(best.ets.exp))



autoplot(train.exp.ts)+autolayer(fitted(best.arima.exp.nolambda))+autolayer(forecast(best.arima.exp.nolambda))

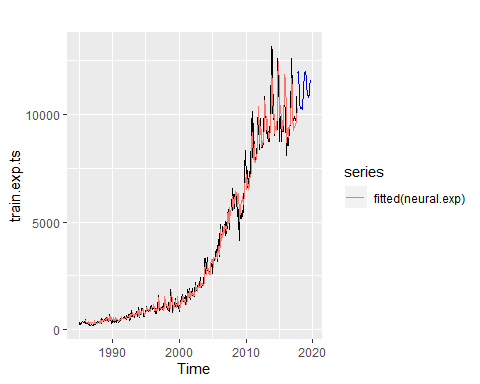


autoplot(train.exp.ts)+autolayer(fitted(best.arima.exp.yeslambda))+autolayer(forecast(best.arima.exp.yeslambda))



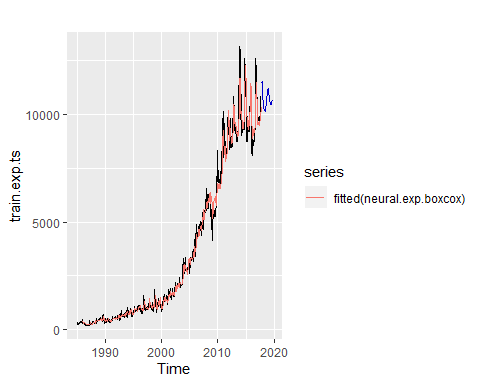
autoplot(train.exp.ts)+autolayer(fitted(neural.exp))+autolayer(forecast(neural.exp))

## Warning: Removed 12 rows containing missing values (geom\_path).

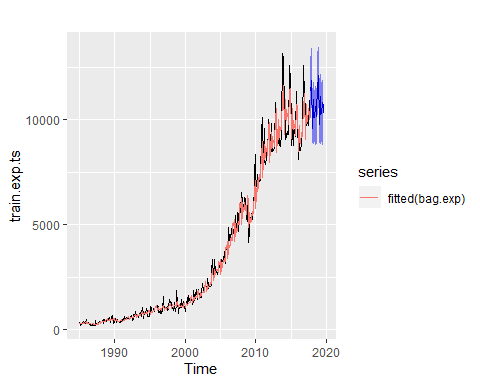


autoplot(train.exp.ts)+autolayer(fitted(neural.exp.boxcox))+autolayer(forecast(neural.exp.boxcox))

## Warning: Removed 12 rows containing missing values (geom\_path).



autoplot(train.exp.ts)+autolayer(fitted(bag.exp))+autolayer(forecast(bag.exp))



autoplot(train.exp.ts)+autolayer(fitted(bag.exp.wlambda))+autolayer(forecast(bag.exp.wlambda))

