Project\_Tennis

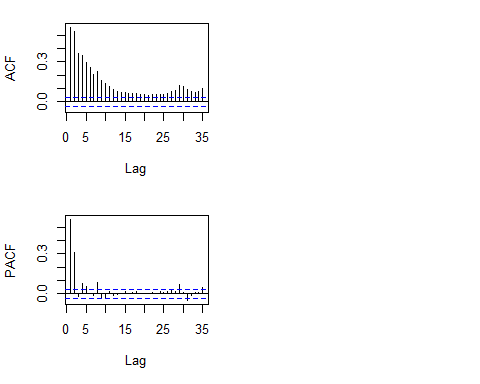
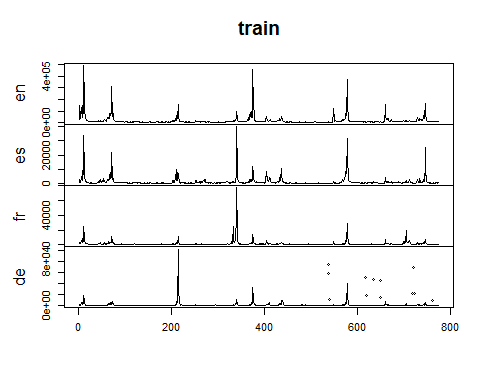
Effy Fang

August 25, 2019

library(fpp)  
library(TSA)  
library(ggplot2)  
library(vars)

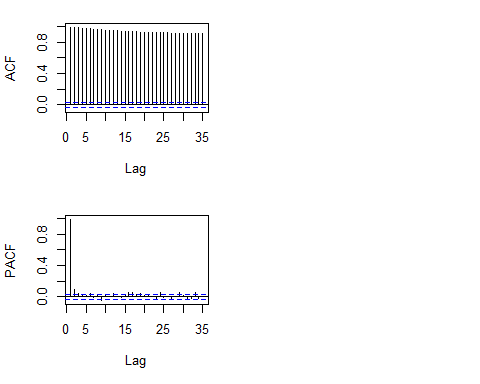
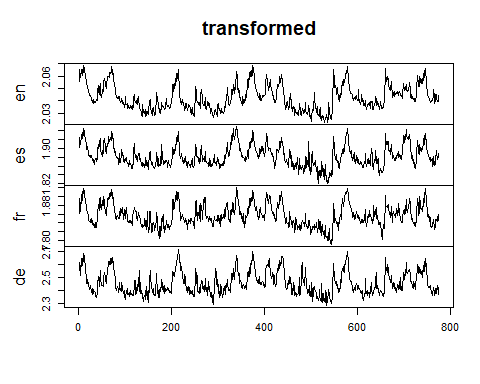
## data preparation

datapath <- "C:/Users/effyf/Documents/MSCA/time series/project/TimeSeries-Project/TSProject"  
project.dat<-readRDS(file=paste(datapath,"new\_train.rda",sep="/"))  
#head(project.dat)  
tennis.en=as.integer(project.dat[,6])  
tennis.fr=as.integer(project.dat[,7])  
tennis.es=as.integer(project.dat[,8])  
tennis.de=as.integer(project.dat[,9])  
tennis=cbind(en=tennis.en,es=tennis.es,fr=tennis.fr,de=tennis.de)  
  
  
# convert to ts  
ts.tennis=ts(tennis,frequency = 1) #,start=c(2015,7))   
train <- window(ts.tennis, start=1, end=775)  
test <-window(ts.tennis,start=776) # test on the last 4 weeks  
  
ts.weekly <- ts(tennis, frequency = 7)  
train.weekly <- window(ts.weekly, end=c(111,5))  
test.weekly <- window(ts.weekly, start=c(111,6))  
tsdisplay(train)



the variance is not stable, need boxcox transformation

lambda1 <- BoxCox.lambda(train[,1])  
transformed.en <- BoxCox(train[,1],lambda=lambda1)  
lambda2 <- BoxCox.lambda(train[,2])  
transformed.fr <- BoxCox(train[,2],lambda=lambda2)  
lambda3 <- BoxCox.lambda(train[,3])  
transformed.es <- BoxCox(train[,3],lambda=lambda3)  
lambda4 <- BoxCox.lambda(train[,4])  
transformed.de <- BoxCox(train[,4],lambda=lambda4)  
transformed=cbind(en=transformed.en,es=transformed.es,fr=transformed.fr,  
 de=transformed.de)  
tsdisplay(transformed)

 ## stationary test

kpss.test(transformed.en) #is level stationary

##   
## KPSS Test for Level Stationarity  
##   
## data: transformed.en  
## KPSS Level = 0.378, Truncation lag parameter = 6, p-value =  
## 0.08664

adf.test(transformed.en)

##   
## Augmented Dickey-Fuller Test  
##   
## data: transformed.en  
## Dickey-Fuller = -4.996, Lag order = 9, p-value = 0.01  
## alternative hypothesis: stationary

adf.test(transformed.es)

##   
## Augmented Dickey-Fuller Test  
##   
## data: transformed.es  
## Dickey-Fuller = -5.6441, Lag order = 9, p-value = 0.01  
## alternative hypothesis: stationary

adf.test(transformed.fr)

##   
## Augmented Dickey-Fuller Test  
##   
## data: transformed.fr  
## Dickey-Fuller = -5.6695, Lag order = 9, p-value = 0.01  
## alternative hypothesis: stationary

adf.test(transformed.de)

##   
## Augmented Dickey-Fuller Test  
##   
## data: transformed.de  
## Dickey-Fuller = -5.8892, Lag order = 9, p-value = 0.01  
## alternative hypothesis: stationary

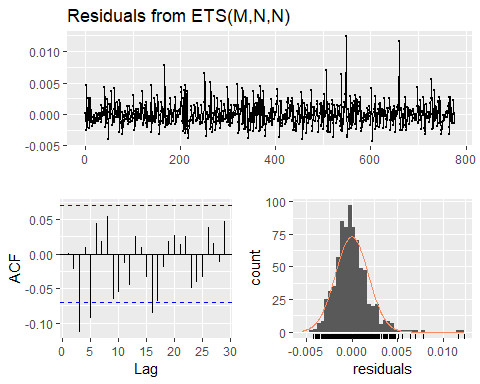
# model fitting

## exponential smoothing

#m.ses <- ses(transformed.en)  
#summary(m.ses)  
#m.holt <- holt(transformed.en)  
#summary(m.holt)  
m.ets <- ets(transformed.en) # best amoung the 3  
summary(m.ets)

## ETS(M,N,N)   
##   
## Call:  
## ets(y = transformed.en)   
##   
## Smoothing parameters:  
## alpha = 0.9542   
##   
## Initial states:  
## l = 2.0608   
##   
## sigma: 0.0018  
##   
## AIC AICc BIC   
## -3523.282 -3523.251 -3509.324   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE  
## Training set -2.672513e-05 0.003687297 0.002711013 -0.001472833 0.1325645  
## MASE ACF1  
## Training set 0.999373 0.0004279761

checkresiduals(m.ets)



##   
## Ljung-Box test  
##   
## data: Residuals from ETS(M,N,N)  
## Q\* = 27.231, df = 8, p-value = 0.0006449  
##   
## Model df: 2. Total lags used: 10

residuals do not look good. we’ll not consider this model

f.ets=forecast(m.ets,h=28)  
error.ets=InvBoxCox(f.ets$mean,lambda=lambda1)-test[,1]  
unique(InvBoxCox(f.ets$mean,lambda=lambda1)) #only gives 1 value in the preidction

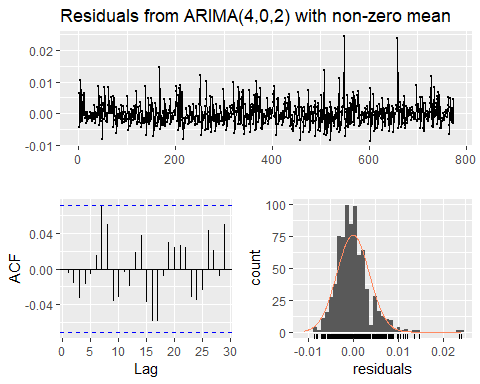
## [1] 6017.425

## auto.arima

# fit auto.arima on en  
m.auto=auto.arima(train[,1],lambda=lambda1)  
summary(m.auto)

## Series: train[, 1]   
## ARIMA(4,0,2) with non-zero mean   
## Box Cox transformation: lambda= -0.4826064   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2 mean  
## -0.7326 0.7675 0.7293 0.0042 1.6715 0.8172 2.0428  
## s.e. 0.1684 0.0685 0.1372 0.0473 0.1647 0.1291 0.0019  
##   
## sigma^2 estimated as 1.308e-05: log likelihood=3259.97  
## AIC=-6503.93 AICc=-6503.75 BIC=-6466.71  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 3185.942 30567.34 7144.078 -3.723988 22.4529 0.7655936  
## ACF1  
## Training set -0.1083828

checkresiduals(m.auto)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,0,2) with non-zero mean  
## Q\* = 9.2117, df = 3, p-value = 0.0266  
##   
## Model df: 7. Total lags used: 10

residuals look okay, pass ljung-box test if we lower the significance level.

f.auto <- forecast(m.auto,h=28)  
error.auto <- f.auto$mean-test[,1]  
#plot(forecast(m.auto,h=28),include=100)

## arima

eacf(transformed.en)

## AR/MA  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
## 0 x x x x x x x x x x x x x x   
## 1 o o x o x o o x o o o o o o   
## 2 x o x o o x o x o o o o o o   
## 3 x o o o o o o x o o o o o o   
## 4 x x o o o o o x o o o o o o   
## 5 x x o x o o o x o o o o o o   
## 6 x x x x x o o x o o o o o o   
## 7 x x x x o x o x o o o o o o

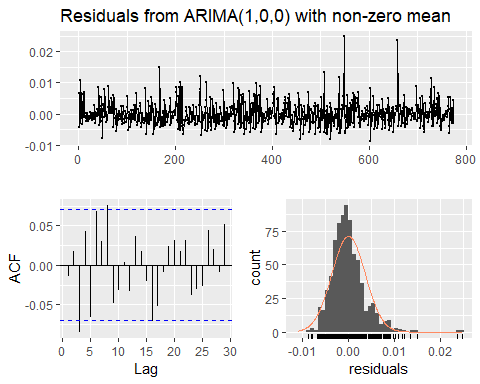
m.1.0 <- Arima(train[,1], order=c(1,0,0), lambda=lambda1)  
m.1.1 <- Arima(train[,1], order=c(1,0,1),lambda=lambda1)  
summary(m.1.0) # better

## Series: train[, 1]   
## ARIMA(1,0,0) with non-zero mean   
## Box Cox transformation: lambda= -0.4826064   
##   
## Coefficients:  
## ar1 mean  
## 0.9327 2.0428  
## s.e. 0.0129 0.0019  
##   
## sigma^2 estimated as 1.322e-05: log likelihood=3253.29  
## AIC=-6500.58 AICc=-6500.55 BIC=-6486.62  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 3147.97 31090.47 7381.518 -3.807397 22.60383 0.7910387  
## ACF1  
## Training set -0.1917853

summary(m.1.1)

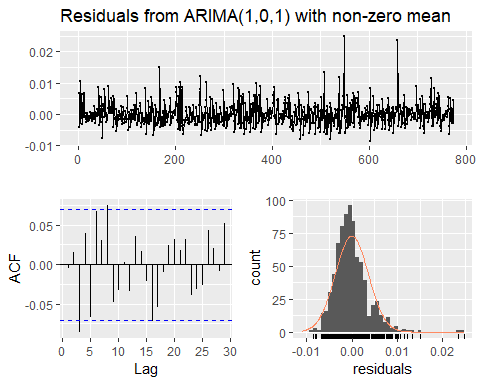
## Series: train[, 1]   
## ARIMA(1,0,1) with non-zero mean   
## Box Cox transformation: lambda= -0.4826064   
##   
## Coefficients:  
## ar1 ma1 mean  
## 0.9344 -0.0117 2.0427  
## s.e. 0.0137 0.0381 0.0019  
##   
## sigma^2 estimated as 1.324e-05: log likelihood=3253.33  
## AIC=-6498.67 AICc=-6498.62 BIC=-6480.06  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 3177.725 31051.9 7366.58 -3.747261 22.58665 0.7894379  
## ACF1  
## Training set -0.1812352

checkresiduals(m.1.0)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,0) with non-zero mean  
## Q\* = 22.304, df = 8, p-value = 0.004383  
##   
## Model df: 2. Total lags used: 10

checkresiduals(m.1.1)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,1) with non-zero mean  
## Q\* = 21.892, df = 7, p-value = 0.002652  
##   
## Model df: 3. Total lags used: 10

AIC from arima(1,0,0) is not much different than auto.arima, but this model is simpler. However the redisual p-value is far from passing the ljung-box test, so we’ll not consider this model.

f.arima.100 <- forecast(m.1.0,h=28)  
error.arima.100 <- f.arima.100$mean-test[,1]

## sarima

# reconstruct ts with 365, assume annual seasonality   
### not good!   
#annual.en=ts(transformed.en,frequency=365)  
# fit auto.arima on en  
#m2=auto.arima(annual.en,seasonal=T, D=1)  
#summary(m2)  
  
# reconstruct ts with 30, assume monthly seasonality   
### weekly is better.  
#monthly.en=ts(transformed.en,frequency=30)  
# fit auto.arima on en  
#m.monthly=auto.arima(monthly.en,seasonal=T, D=1)  
#summary(m.monthly)  
#checkresiduals(m.monthly)

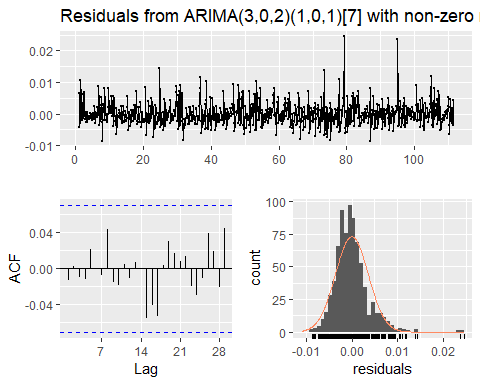
## weekly

# fit auto.arima on weekly ts   
m.weekly=auto.arima(train.weekly[,1],seasonal=T,lambda=lambda1)  
summary(m.weekly)

## Series: train.weekly[, 1]   
## ARIMA(3,0,2)(1,0,1)[7] with non-zero mean   
## Box Cox transformation: lambda= -0.4826064   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 ma2 sar1 sma1 mean  
## -0.7796 0.7492 0.7451 1.7197 0.8654 0.5562 -0.4573 2.0428  
## s.e. 0.0734 0.0352 0.0653 0.0598 0.0543 0.2532 0.2709 0.0019  
##   
## sigma^2 estimated as 1.296e-05: log likelihood=3263.86  
## AIC=-6509.72 AICc=-6509.48 BIC=-6467.84  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 3126.547 30712.47 7157.469 -3.735335 22.26676 0.4569681  
## ACF1  
## Training set -0.1147682

AIC is lower and loglikelyhood is better.

checkresiduals(m.weekly)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,0,2)(1,0,1)[7] with non-zero mean  
## Q\* = 2.8275, df = 6, p-value = 0.8302  
##   
## Model df: 8. Total lags used: 14

the residuals also look like white noise and pass ljung-box test.

f.sarima <- forecast(m.weekly,h=28)  
error.sarima <- f.sarima$mean-test.weekly[,1]

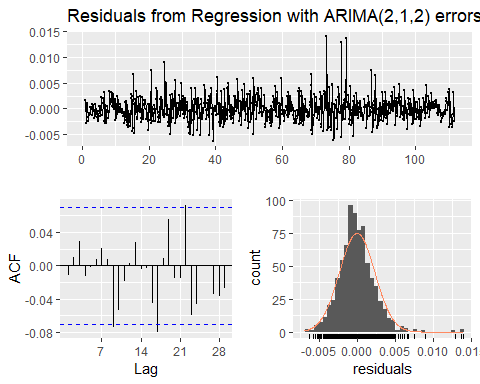
## regression with arima error

adding the influence from es, fr, and de pages.

xreg =cbind(transformed.es, transformed.fr, transformed.de)  
m.reg <- auto.arima(train.weekly[,1],xreg =xreg,seasonal=T, lambda=lambda1)  
summary(m.reg)

## Series: train.weekly[, 1]   
## Regression with ARIMA(2,1,2) errors   
## Box Cox transformation: lambda= -0.4826064   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 transformed.es transformed.fr  
## -0.3616 0.4353 0.0044 -0.5897 0.1276 0.1076  
## s.e. 0.1661 0.0908 0.1616 0.0829 0.0101 0.0092  
## transformed.de  
## 0.0198  
## s.e. 0.0025  
##   
## sigma^2 estimated as 5.481e-06: log likelihood=3593.54  
## AIC=-7171.08 AICc=-7170.89 BIC=-7133.86  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 311.7833 12379.86 3344.256 -1.655122 13.51809 0.2135138  
## ACF1  
## Training set 0.05532041

checkresiduals(m.reg)



##   
## Ljung-Box test  
##   
## data: Residuals from Regression with ARIMA(2,1,2) errors  
## Q\* = 8.9192, df = 7, p-value = 0.2585  
##   
## Model df: 7. Total lags used: 14

further lowers AIC, residuals look good.

transformed.test=cbind(transformed.es=BoxCox(test[,2],lambda=lambda2),  
 transformed.fr=BoxCox(test[,3],lambda=lambda3),  
 transformed.de=BoxCox(test[,4],lambda=lambda4))  
f.xreg <- forecast(m.reg,xreg=transformed.test,h=28)  
f.xreg

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 111.7143 5581.775 4641.536 6834.673 4236.183 7673.672  
## 111.8571 7041.603 5528.769 9260.124 4916.071 10878.886  
## 112.0000 8210.939 6143.488 11505.951 5348.616 14104.925  
## 112.1429 7187.416 5386.288 10049.932 4692.617 12301.734  
## 112.2857 4993.897 3853.751 6716.046 3400.554 8007.401  
## 112.4286 5350.171 4053.546 7371.882 3548.261 8933.681  
## 112.5714 6414.408 4685.966 9288.177 4038.259 11649.344  
## 112.7143 5637.390 4153.099 8068.564 3591.848 10037.250  
## 112.8571 4735.933 3535.321 6657.165 3074.862 8178.420  
## 113.0000 4438.961 3315.458 6235.064 2884.317 7655.925  
## 113.1429 3789.482 2864.470 5237.585 2504.895 6360.527  
## 113.2857 3949.395 2948.636 5550.383 2564.761 6817.735  
## 113.4286 4373.223 3195.106 6331.674 2753.580 7940.525  
## 113.5714 3969.953 2919.482 5696.009 2523.029 7098.003  
## 113.7143 7037.116 4692.145 11652.879 3894.790 16156.149  
## 113.8571 9821.003 6089.711 18298.229 4914.308 27988.087  
## 114.0000 9448.562 5849.500 17651.839 4717.630 27063.438  
## 114.1429 9658.919 5899.881 18466.722 4734.214 28900.513  
## 114.2857 15827.328 8525.937 38385.960 6538.697 76354.153  
## 114.4286 20612.615 10222.765 59976.105 7624.986 149753.125  
## 114.5714 19221.831 9617.378 54791.792 7194.232 133031.460  
## 114.7143 12242.244 6834.647 27629.209 5307.509 50807.547  
## 114.8571 11417.869 6426.725 25361.057 5005.219 45866.812  
## 115.0000 17397.613 8686.614 49831.100 6493.542 121775.918  
## 115.1429 14479.461 7536.582 37770.111 5712.521 81844.039  
## 115.2857 19140.651 9115.717 61453.875 6710.729 177179.689  
## 115.4286 13576.543 7067.705 35402.903 5357.405 76683.012  
## 115.5714 13639.730 7033.365 36317.283 5314.314 80715.379

error.xreg <- f.xreg$mean-test.weekly[,1]

## VAR

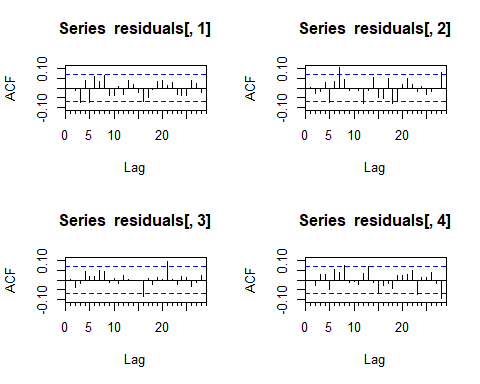
VARselect(transformed)

## $selection  
## AIC(n) HQ(n) SC(n) FPE(n)   
## 4 2 2 4   
##   
## $criteria  
## 1 2 3 4  
## AIC(n) -3.726759e+01 -3.736971e+01 -3.738896e+01 -3.739569e+01  
## HQ(n) -3.722089e+01 -3.728565e+01 -3.726754e+01 -3.723692e+01  
## SC(n) -3.714628e+01 -3.715136e+01 -3.707357e+01 -3.698326e+01  
## FPE(n) 6.529701e-17 5.895814e-17 5.783466e-17 5.744728e-17  
## 5 6 7 8  
## AIC(n) -3.736920e+01 -3.736076e+01 -3.736741e+01 -3.734996e+01  
## HQ(n) -3.717307e+01 -3.712727e+01 -3.709656e+01 -3.704175e+01  
## SC(n) -3.685973e+01 -3.675424e+01 -3.666385e+01 -3.654935e+01  
## FPE(n) 5.899093e-17 5.949320e-17 5.910225e-17 6.014687e-17  
## 9 10  
## AIC(n) -3.734614e+01 -3.732560e+01  
## HQ(n) -3.700057e+01 -3.694268e+01  
## SC(n) -3.644849e+01 -3.633091e+01  
## FPE(n) 6.038239e-17 6.164173e-17

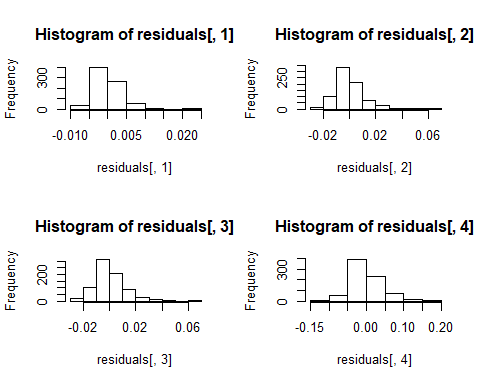
# 4 selected by aic, 2 by bic  
m.var <- VAR(transformed,p=2,type = "both")  
summary(m.var)

##   
## VAR Estimation Results:  
## =========================   
## Endogenous variables: en, es, fr, de   
## Deterministic variables: both   
## Sample size: 773   
## Log Likelihood: 10094.231   
## Roots of the characteristic polynomial:  
## 0.9297 0.8535 0.7926 0.7374 0.3012 0.2113 0.2079 0.02871  
## Call:  
## VAR(y = transformed, p = 2, type = "both")  
##   
##   
## Estimation results for equation en:   
## ===================================   
## en = en.l1 + es.l1 + fr.l1 + de.l1 + en.l2 + es.l2 + fr.l2 + de.l2 + const + trend   
##   
## Estimate Std. Error t value Pr(>|t|)   
## en.l1 8.441e-01 5.455e-02 15.474 <2e-16 \*\*\*  
## es.l1 2.052e-02 1.709e-02 1.201 0.2303   
## fr.l1 -8.983e-03 1.563e-02 -0.575 0.5658   
## de.l1 8.289e-03 4.101e-03 2.021 0.0436 \*   
## en.l2 1.186e-01 5.467e-02 2.169 0.0304 \*   
## es.l2 -1.159e-02 1.711e-02 -0.677 0.4983   
## fr.l2 -8.865e-03 1.564e-02 -0.567 0.5709   
## de.l2 -1.012e-02 4.091e-03 -2.475 0.0135 \*   
## const 9.706e-02 4.932e-02 1.968 0.0494 \*   
## trend 1.018e-07 6.034e-07 0.169 0.8660   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## Residual standard error: 0.00362 on 763 degrees of freedom  
## Multiple R-Squared: 0.8707, Adjusted R-squared: 0.8692   
## F-statistic: 570.9 on 9 and 763 DF, p-value: < 2.2e-16   
##   
##   
## Estimation results for equation es:   
## ===================================   
## es = en.l1 + es.l1 + fr.l1 + de.l1 + en.l2 + es.l2 + fr.l2 + de.l2 + const + trend   
##   
## Estimate Std. Error t value Pr(>|t|)   
## en.l1 7.738e-01 1.618e-01 4.783 2.07e-06 \*\*\*  
## es.l1 5.889e-01 5.069e-02 11.618 < 2e-16 \*\*\*  
## fr.l1 1.394e-02 4.637e-02 0.301 0.76381   
## de.l1 2.834e-03 1.216e-02 0.233 0.81582   
## en.l2 -4.624e-01 1.621e-01 -2.852 0.00447 \*\*   
## es.l2 1.830e-01 5.074e-02 3.607 0.00033 \*\*\*  
## fr.l2 8.825e-03 4.637e-02 0.190 0.84913   
## de.l2 -7.790e-03 1.213e-02 -0.642 0.52103   
## const -2.362e-01 1.463e-01 -1.615 0.10680   
## trend -1.664e-06 1.790e-06 -0.930 0.35282   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## Residual standard error: 0.01074 on 763 degrees of freedom  
## Multiple R-Squared: 0.7945, Adjusted R-squared: 0.7921   
## F-statistic: 327.8 on 9 and 763 DF, p-value: < 2.2e-16   
##   
##   
## Estimation results for equation fr:   
## ===================================   
## fr = en.l1 + es.l1 + fr.l1 + de.l1 + en.l2 + es.l2 + fr.l2 + de.l2 + const + trend   
##   
## Estimate Std. Error t value Pr(>|t|)   
## en.l1 7.945e-01 1.700e-01 4.673 3.51e-06 \*\*\*  
## es.l1 4.412e-02 5.327e-02 0.828 0.40786   
## fr.l1 5.631e-01 4.873e-02 11.556 < 2e-16 \*\*\*  
## de.l1 1.424e-02 1.278e-02 1.114 0.26551   
## en.l2 -4.896e-01 1.704e-01 -2.873 0.00418 \*\*   
## es.l2 3.614e-02 5.332e-02 0.678 0.49815   
## fr.l2 1.487e-01 4.873e-02 3.051 0.00236 \*\*   
## de.l2 -1.453e-02 1.275e-02 -1.140 0.25483   
## const -2.370e-01 1.537e-01 -1.542 0.12358   
## trend -2.493e-06 1.881e-06 -1.326 0.18530   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## Residual standard error: 0.01128 on 763 degrees of freedom  
## Multiple R-Squared: 0.7865, Adjusted R-squared: 0.784   
## F-statistic: 312.3 on 9 and 763 DF, p-value: < 2.2e-16   
##   
##   
## Estimation results for equation de:   
## ===================================   
## de = en.l1 + es.l1 + fr.l1 + de.l1 + en.l2 + es.l2 + fr.l2 + de.l2 + const + trend   
##   
## Estimate Std. Error t value Pr(>|t|)   
## en.l1 2.194e+00 6.096e-01 3.599 0.000341 \*\*\*  
## es.l1 2.723e-01 1.910e-01 1.426 0.154333   
## fr.l1 1.428e-03 1.747e-01 0.008 0.993480   
## de.l1 6.254e-01 4.583e-02 13.645 < 2e-16 \*\*\*  
## en.l2 -1.061e+00 6.110e-01 -1.737 0.082766 .   
## es.l2 -1.227e-01 1.912e-01 -0.642 0.521070   
## fr.l2 4.959e-02 1.747e-01 0.284 0.776621   
## de.l2 1.098e-01 4.572e-02 2.401 0.016592 \*   
## const -2.039e+00 5.511e-01 -3.700 0.000231 \*\*\*  
## trend 3.038e-07 6.743e-06 0.045 0.964082   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## Residual standard error: 0.04045 on 763 degrees of freedom  
## Multiple R-Squared: 0.7889, Adjusted R-squared: 0.7864   
## F-statistic: 316.8 on 9 and 763 DF, p-value: < 2.2e-16   
##   
##   
##   
## Covariance matrix of residuals:  
## en es fr de  
## en 1.310e-05 2.648e-05 2.643e-05 8.462e-05  
## es 2.648e-05 1.152e-04 6.883e-05 2.347e-04  
## fr 2.643e-05 6.883e-05 1.273e-04 2.339e-04  
## de 8.462e-05 2.347e-04 2.339e-04 1.636e-03  
##   
## Correlation matrix of residuals:  
## en es fr de  
## en 1.0000 0.6814 0.6472 0.5779  
## es 0.6814 1.0000 0.5683 0.5406  
## fr 0.6472 0.5683 1.0000 0.5126  
## de 0.5779 0.5406 0.5126 1.0000

residuals=residuals(m.var)  
par(mfrow=c(2,2))  
Acf(residuals[,1])  
Acf(residuals[,2])  
Acf(residuals[,3])  
Acf(residuals[,4])



par(mfrow=c(2,2))  
hist(residuals[,1])  
hist(residuals[,2])  
hist(residuals[,3])  
hist(residuals[,4])



Box.test(residuals[,1],type="Ljung-Box",lag=12)

##   
## Box-Ljung test  
##   
## data: residuals[, 1]  
## X-squared = 20.233, df = 12, p-value = 0.0628

Box.test(residuals[,2],type="Ljung-Box",lag=12)

##   
## Box-Ljung test  
##   
## data: residuals[, 2]  
## X-squared = 22.809, df = 12, p-value = 0.02939

Box.test(residuals[,3],type="Ljung-Box",lag=12)

##   
## Box-Ljung test  
##   
## data: residuals[, 3]  
## X-squared = 7.7243, df = 12, p-value = 0.8063

Box.test(residuals[,4],type="Ljung-Box",lag=12)

##   
## Box-Ljung test  
##   
## data: residuals[, 4]  
## X-squared = 13.273, df = 12, p-value = 0.3495

all the residuals look okay.

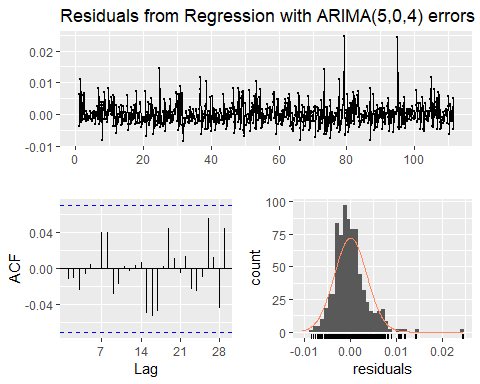
f.var=forecast(m.var,h=28)  
f.var.en=f.var$forecast$en$mean  
error.var=InvBoxCox(f.var.en, lambda=lambda1)-test[,1]  
#error.var = sqrt(mean(InvBoxCox(forecast(m.var, h=28)$forecast$transformed.en$mean, lambda=lambda1)-test[,1])^2)

## fourier transformation - dynamic harmonic regression

transformed.weekly.en=ts(transformed.en,frequency = 7)  
harmonic <- fourier(transformed.weekly.en,K=3)  
m.fourier <- auto.arima(transformed.weekly.en,xreg=harmonic, seasonal=F)  
summary(m.fourier)

## Series: transformed.weekly.en   
## Regression with ARIMA(5,0,4) errors   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ma1 ma2 ma3  
## -1.3313 0.1642 1.0876 0.5685 0.1222 2.2728 1.9899 0.7598  
## s.e. 0.4157 NaN 0.6251 NaN NaN 0.3921 NaN NaN  
## ma4 intercept S1-7 C1-7 S2-7 C2-7 S3-7 C3-7  
## 0.1373 2.0429 -4e-04 8e-04 -2e-04 -3e-04 1e-04 1e-04  
## s.e. NaN 0.0020 2e-04 2e-04 1e-04 1e-04 1e-04 1e-04  
##   
## sigma^2 estimated as 1.275e-05: log likelihood=3274.47  
## AIC=-6514.94 AICc=-6514.13 BIC=-6435.84  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE  
## Training set -3.823015e-05 0.003533254 0.002577319 -0.0021633 0.1260188  
## MASE ACF1  
## Training set 0.413209 -0.01226784

checkresiduals(m.fourier)



##   
## Ljung-Box test  
##   
## data: Residuals from Regression with ARIMA(5,0,4) errors  
## Q\* = 11.817, df = 3, p-value = 0.008039  
##   
## Model df: 16. Total lags used: 19

residuals do not pass the Ljung-box test. couple of outliers in the residuals.

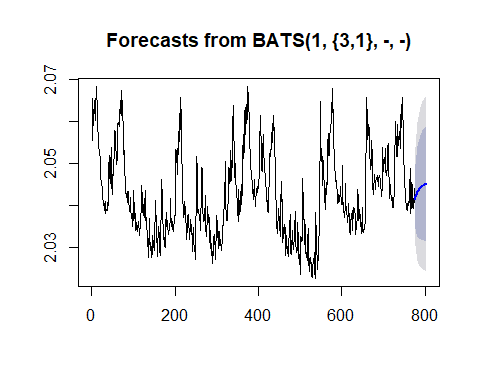
new.harmonic <- fourier(transformed.weekly.en,K=3,h=28)  
f.fourier <- forecast(m.fourier,xreg=new.harmonic, h=28)  
error.fourier <- InvBoxCox(f.fourier$mean, lambda=lambda1)-as.numeric(test[,1])

## TBATS

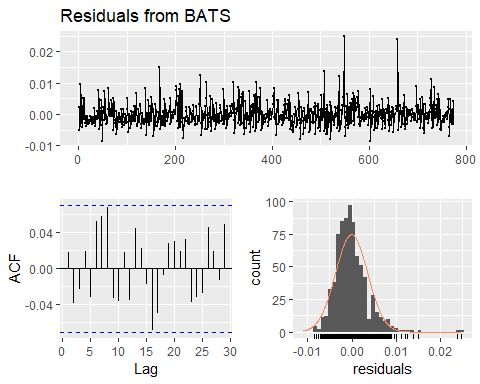
m.tbats <- tbats(transformed.en)  
print(m.tbats)

## BATS(1, {3,1}, -, -)  
##   
## Call: tbats(y = transformed.en)  
##   
## Parameters  
## Alpha: 0.01801944  
## AR coefficients: 0.198666 0.700155 -0.047834  
## MA coefficients: 0.704137  
##   
## Seed States:  
## [,1]  
## [1,] 2.058643  
## [2,] 0.000000  
## [3,] 0.000000  
## [4,] 0.000000  
## [5,] 0.000000  
##   
## Sigma: 0.003629454  
## AIC: -3532.974

plot(forecast(m.tbats, h=28))



checkresiduals(m.tbats)

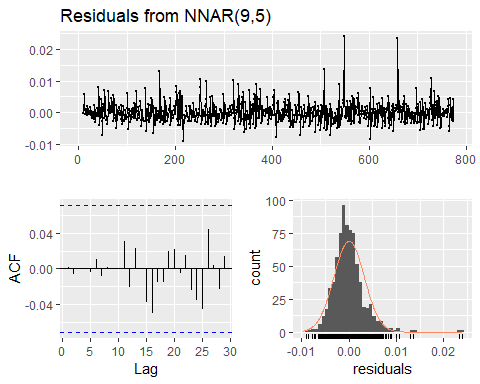


##   
## Ljung-Box test  
##   
## data: Residuals from BATS  
## Q\* = 16.11, df = 3, p-value = 0.001077  
##   
## Model df: 10. Total lags used: 13

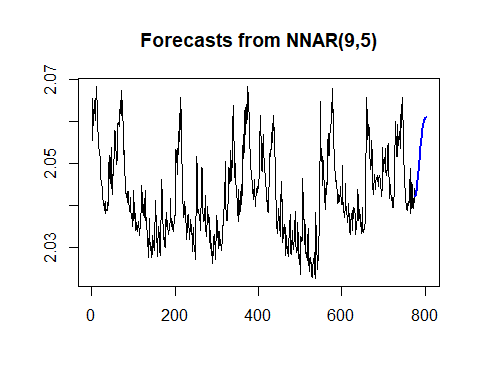
residuals do not pass ljung-box test.

## nnetar

nn <-nnetar(transformed.en)  
checkresiduals(nn)



f.nn=forecast(nn,h=28)  
error.nn=InvBoxCox(f.nn$mean, lambda=lambda1)-(test[,1])  
rmse.nn<-sqrt(mean(error.nn^2,na.rm=TRUE))  
plot(forecast(nn, h=28))



## model comparison

aicc <- cbind(ets=m.ets$aicc, auto.arima=m.auto$aicc, arima.100=m.1.0$aicc,   
 sarima=m.weekly$aicc, xreg=m.reg$aicc, varma.AIC =AIC(m.var), fourier = m.fourier$aicc)   
aicc

## ets auto.arima arima.100 sarima xreg varma.AIC fourier  
## [1,] -3523.251 -6503.746 -6500.547 -6509.48 -7170.887 -20108.46 -6514.131

sort(aicc)

## [1] -20108.461 -7170.887 -6514.131 -6509.480 -6503.746 -6500.547  
## [7] -3523.251

bic <- cbind(ets=m.ets$bic, auto.arima=m.auto$bic, arima.100=m.1.0$bic,   
 sarima=m.weekly$bic, xreg=m.reg$bic, varma =BIC(m.var), fourier = m.fourier$bic)   
bic

## ets auto.arima arima.100 sarima xreg varma fourier  
## [1,] -3509.324 -6466.711 -6486.62 -6467.84 -7133.863 -19922.45 -6435.841

ll <- cbind(ets=m.ets$loglik, auto.arima=m.auto$loglik, arima.100=m.1.0$loglik,   
 sarima=m.weekly$loglik, xreg=m.reg$loglik, varma =logLik(m.var), fourier = m.fourier$loglik)   
ll

## ets auto.arima arima.100 sarima xreg varma fourier  
## [1,] 1764.641 3259.967 3253.289 3263.858 3593.538 10094.23 3274.47

rmse.ets <- sqrt(mean(error.ets^2,na.rm=TRUE))  
rmse.auto <- sqrt(mean(error.auto^2,na.rm=TRUE))  
rmse.arima.100 <- sqrt(mean(error.arima.100^2,na.rm=TRUE))  
rmse.weekly <- sqrt(mean(error.sarima^2,na.rm=TRUE))  
rmse.xreg <- sqrt(mean(error.xreg^2,na.rm=TRUE))  
rmse.varma <- sqrt(mean(error.var^2,na.rm=TRUE))  
rmse.fourier <- sqrt(mean(error.fourier^2,na.rm=TRUE))  
  
rmse <- c(ets=rmse.ets, auto.arima = rmse.auto, arima.100=rmse.arima.100,  
 sarima=rmse.weekly, xreg=rmse.xreg, varma=rmse.varma, fourier=rmse.fourier, nnetar=rmse.nn)  
rmse

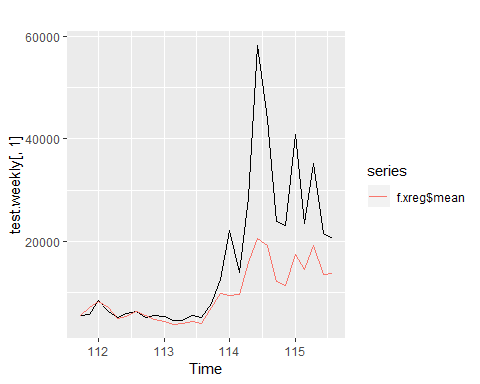
## ets auto.arima arima.100 sarima xreg varma   
## 17392.61 17044.36 17040.78 17080.81 11406.77 16764.97   
## fourier nnetar   
## 17132.87 13251.60

data <- rbind(aicc,bic,ll,rmse)  
comparison <- as.data.frame(t(data),row.name=c('ets','auto.arima','arima(1,0,0)','sarima(weekly)', 'regression with arima error', 'varma', 'fourier'))  
colnames(comparison)<-c('aic', 'bic', 'loglik', 'rmse')  
comparison[order(comparison$rmse),]

## aic bic loglik rmse  
## regression with arima error -7170.887 -7133.863 3593.538 11406.77  
## varma -20108.461 -19922.450 10094.231 16764.97  
## arima(1,0,0) -6500.547 -6486.620 3253.289 17040.78  
## auto.arima -6503.746 -6466.711 3259.967 17044.36  
## sarima(weekly) -6509.480 -6467.840 3263.858 17080.81  
## fourier -6514.131 -6435.841 3274.470 17132.87  
## ets -3523.251 -3509.324 1764.641 17392.61

var(2) model has a much lower aic and bic then other models, and way bigger loglikelihood. However, this is computed from all 4 equations. could not compare it with other 1 equation models. But we can compare the RMSE on the reserved test set. regression with arima error model seems to be the best fit.

autoplot(test.weekly[,1])+autolayer(f.xreg$mean)

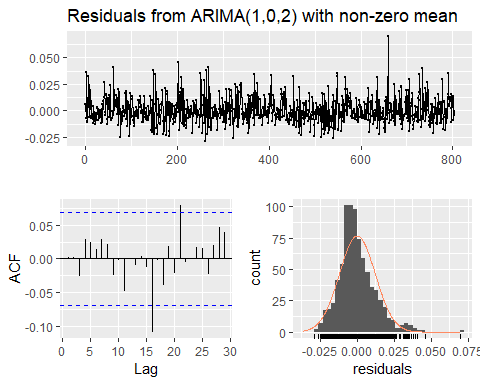
 error is bigger for the later time. it captures the general trend but underestimates the peak.

## forecasting 1 week out

## using arima to predict es, fr and de  
m.es <- auto.arima(tennis[,2],lambda='auto')  
summary(m.es)

## Series: tennis[, 2]   
## ARIMA(1,0,2) with non-zero mean   
## Box Cox transformation: lambda= -0.5131604   
##   
## Coefficients:  
## ar1 ma1 ma2 mean  
## 0.9126 -0.1117 -0.0553 1.8757  
## s.e. 0.0182 0.0402 0.0378 0.0040  
##   
## sigma^2 estimated as 0.0001467: log likelihood=2406  
## AIC=-4802 AICc=-4801.92 BIC=-4778.56  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 335.7386 2436.105 579.4839 -3.460336 26.76442 0.7509494  
## ACF1  
## Training set 0.06577733

checkresiduals(m.es)

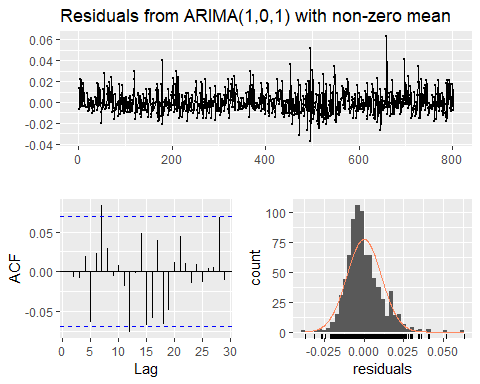


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,2) with non-zero mean  
## Q\* = 3.4277, df = 6, p-value = 0.7536  
##   
## Model df: 4. Total lags used: 10

m.fr <- auto.arima(tennis[,3],lambda='auto')  
summary(m.fr)

## Series: tennis[, 3]   
## ARIMA(1,0,1) with non-zero mean   
## Box Cox transformation: lambda= -0.512842   
##   
## Coefficients:  
## ar1 ma1 mean  
## 0.9100 -0.1209 1.8787  
## s.e. 0.0165 0.0400 0.0037  
##   
## sigma^2 estimated as 0.000118: log likelihood=2492.87  
## AIC=-4977.74 AICc=-4977.69 BIC=-4958.98  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 356.7594 3139.861 601.973 -3.052422 24.55874 0.7394736  
## ACF1  
## Training set 0.1578683

checkresiduals(m.fr)

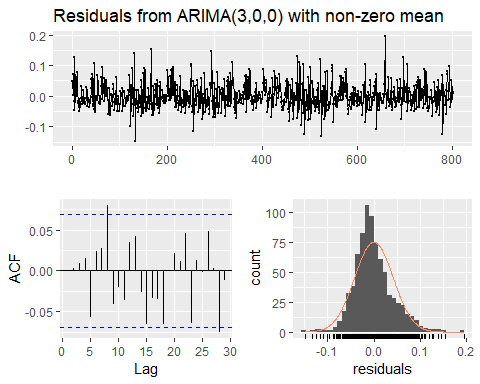


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,1) with non-zero mean  
## Q\* = 10.589, df = 7, p-value = 0.1576  
##   
## Model df: 3. Total lags used: 10

m.de <- auto.arima(tennis[,4],lambda = 'auto')  
summary(m.de)

## Series: tennis[, 4]   
## ARIMA(3,0,0) with non-zero mean   
## Box Cox transformation: lambda= -0.3625696   
##   
## Coefficients:  
## ar1 ar2 ar3 mean  
## 0.8102 0.0396 0.0436 2.4546  
## s.e. 0.0352 0.0454 0.0353 0.0136  
##   
## sigma^2 estimated as 0.001724: log likelihood=1416.72  
## AIC=-2823.44 AICc=-2823.37 BIC=-2800  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 389.3231 4031.202 595.0305 -5.682306 30.50874 0.7014538  
## ACF1  
## Training set 0.05038729

checkresiduals(m.de)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,0,0) with non-zero mean  
## Q\* = 10.787, df = 6, p-value = 0.09517  
##   
## Model df: 4. Total lags used: 10

# use predicted values to construct a new xreg to predict 1 more week of en  
new.es <- forecast(m.es,h=7)$mean  
new.fr <- forecast(m.fr,h=7)$mean  
new.de <- forecast(m.de,h=7)$mean  
# append the next 7 days prediction to the test period  
new.xreg=cbind(transformed.es=BoxCox(append(test[,2],new.es),lambda=lambda2),  
 transformed.fr=BoxCox(append(test[,3],new.fr),lambda=lambda3),  
 transformed.de=BoxCox(append(test[,4],new.de),lambda=lambda4))  
autoplot(forecast(m.reg,h=28+7, xreg=new.xreg),inlcude=50)

