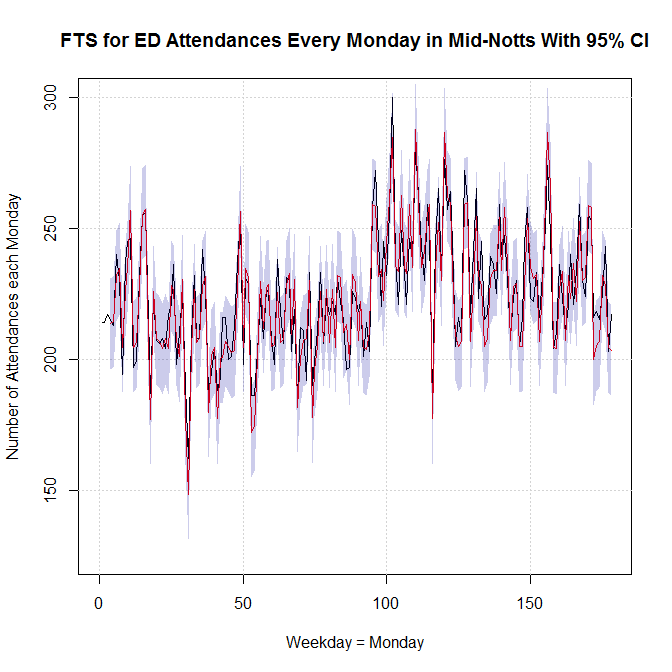
**THIS IS UPDATED VERSION**



**R code for getting the predictions and plotting**

**# Required libraries**

library('forecastHybrid');

library(hybridModels);

library('forecast');

library(TTR);

library(MASS);

library(mgcv);

library(nlme);

library(TSA);

library(AnalyzeTS);

library(dplyr);

library(lubridate)

library('forecast');

library(MASS);

**# Read the data. Set the folder.**

library(readxl);setwd('C:/Users/msztaj/Desktop/EMAHSN- Projects/ED Predictor/Pui-Shan data')

PS\_data <- read\_excel("PSdata.xlsx", sheet = "sheetname");

attach(PS\_data); names(PS\_data)

**# Getting monthly data from daily values**

PS\_monthly = tapply(PS\_data$RK5BCED, week(PS\_data$date), sum)

monthwise <- PS\_monthly[1:52]

monthwise = t(monthwise); # converting row vector to column vector

PS\_yearly = tapply(PS\_data$RK5BCED, year(PS\_data$date), sum)

monthwise = as.data.frame(monthwise)

monthwise =as.ts(monthwise[1:52,])

# As each year the admissions are different from the others, so taking proportion of total is better

#boxplot(RK5BCED ~ PS\_data$Month, main="Month Wise ED Attendances in All Commissioners", xlab="Months",ylab= "Number of ED Attdendances")

#=================================================================================

**# Converting numeric to time series object**

a <- as.ts(monthwise); ts.plot(a) # checking the time series

LMon <- length(monthwise); print(LMon)

**# How many weekdays to predict. Best is 1, 2 or 3. Or number of months to predict**

Forecast4 = 6

**# Splitting into training and testing dataset.**

LL\_trainMon <- LMon - Forecast4; LL\_testMon <- LL\_trainMon + 1 monthwise\_train <- monthwise[1:LL\_trainMon]; length(monthwise\_train)

monthwise\_test <- monthwise[LL\_testMon:LMon]; length(monthwise\_test);

sdMondays <- round(sd(monthwise\_train))

#=================================================================================**# Monday - Fuzzy Time Series Modelling**

#=================================================================================

ts\_mon =time

ts\_mon <- as.ts(ts\_mon); ts\_mon = round(ts\_mon,0); # Remove decimals

D1\_mon = min(ts\_mon); D2\_mon = max(ts\_mon); summary(ts\_mon);

singh\_Mon <- fuzzy.ts1(series,n = 50, D1 = D1\_mon, D2= D2\_mon, type= "Singh", plot=FALSE, trace =TRUE)

rbind(singh\_Mon$accuracy)

singh\_Mon$accuracy

E\_mon = round(monthwise\_train - singh\_Mon$table2$forecast,0);

FTS\_MondayModel <- cbind(ts\_mon,singh\_Mon$table2$forecast,E\_mon);

write.csv(FTS\_MondayModel, 'C:/Users/msztaj/Desktop/EMAHSN- Projects/ED Predictor/EXCEL Files/Monthwise\_MidNotts.csv')

#=================================================================================

# Monday - Next Day Prediction

#=================================================================================

# Alow is lower limit of each fuzzy set

Alow\_Mon <- t(singh\_Mon$table1$dow); Alow\_Mon <-as.data.frame(Alow\_Mon)

# Alow is upper limit of each fuzzy set

Ahigh\_Mon <- t(singh\_Mon$table1$up); Ahigh\_Mon <-as.data.frame(Ahigh\_Mon)

# A is the set of fuzzy numbers with both lower and upper values.

AMon <- rbind(Alow\_Mon,Ahigh\_Mon); UMon <-length(AMon)

# Forecast4 how many weeks. This variable was defined in the start.

Lagg = 6

a = LL\_trainMon - Lagg; b = LL\_trainMon; k=1; c = 0; d = 0;

for (i in a:b) {

for (j in 1:UMon) {

c[k] = which(monthwise\_train[i] >= AMon[1,] & (monthwise\_train[i] < AMon[2,]))

}

k = k + 1

}

FjsMon <- singh\_Mon$table1$mid[c] ; wtsMon <- singh\_Mon$table1$num[c]

Mon\_nextday\_prediction <- sum(FjsMon \*wtsMon)/sum(wtsMon) # next Monday's prediction

e1Mon <- round(Mon\_nextday\_prediction - monthwise\_test[1]); print(e1Mon) # Error in prediction

#=================================================================================# Monday - Next to next Day Prediction

#=================================================================================

Next2NextMondays <- c(monthwise\_train, Mon\_nextday\_prediction) # Including the first week prediction into original data for 2 weeks ahead prediction.

L2Mon <-length(Next2NextMondays); print(L2Mon)

a2Mon = L2Mon - Forecast4 ; b2Mon = L2Mon ; k = 1; c = 0 ; d2Mon = 0;

for (i in a2Mon:b2Mon) {

for (j in 1:UMon) {

c[k] = which(Next2NextMondays[i] >= AMon[1,] & (Next2NextMondays[i] < AMon[2,]))

}

k = k+1

}

Fjs2 <- singh\_Mon$table1$mid[c]; wts2 <- singh\_Mon$table1$num[c]

nex2next\_Monday\_prediction <- sum(Fjs2 \*wts2)/sum(wts2); print(nex2next\_Monday\_prediction)

e2Mon <- round(nex2next\_Monday\_prediction - monthwise\_test[2]); print(e2Mon)

#=================================================================================

# Monday - Next to Next Day Prediction - lag = 3

#=================================================================================

Third2NextMonday <- c(Next2NextMondays,nex2next\_Monday\_prediction)

L3Mon <-length(Third2NextMonday); print(L3Mon)

a = L3Mon - Lagg; b = L3Mon ; k = 1; c = 0 ; d = 0;

for (i in a:b) {

for (j in 1:UMon) {

c[k] = which(Third2NextMonday[i] >= AMon[1,] & (Third2NextMonday[i] < AMon[2,]))

}

K = k +1

}

Fjs2 <- singh\_Mon$table1$mid[c]; wts2 <- singh\_Mon$table1$num[c]

Third2next\_Monday\_prediction <- sum(Fjs2 \*wts2)/sum(wts2); print(Third2next\_Monday\_prediction)

e3Mon <- round(Third2next\_Monday\_prediction - monthwise\_test[3]); print(e3Mon)

errorsMon <- round(c(e1Mon,e2Mon,e3Mon));

print('Errors in Prediction for next 3 days'); print(errorsMon, Quote =TRUE)

ForecastedMon <- errorsMon+monthwise\_test

# ERRROS saved into one file

# @@@@@@@@@@@@@@@@@@@@@@@@@@

errorsMon <- round(c(e1Mon,e2Mon,e3Mon)); print(errorsMon);

errors\_3weeks <- cbind(errorsMon)

write.csv(errors\_3weeks,'C:/Users/msztaj/Desktop/EMAHSN- Projects/ED Predictor/EXCEL Files/FTSnext 3 week day forecasts.csv')

AllForecast\_errors <- cbind(ForecastedMon,errorsMon)

write.csv(AllForecast\_errors,'C:/Users/msztaj/Desktop/EMAHSN- Projects/ED Predictor/EXCEL Files/next 3 week day forecasts with errors.csv')

# @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

# MONDAY - Plotting Time Series with 95% Confidence Intervals and 13 week predictions with 95% #Prediction Intervals

# @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

jpeg("H9\_monthwisegOOD.png");

dev.new(); plot.new();

monthwise\_train\_date <- monthwise$date[1:LL\_trainMon] #Adjusting length of date column

#length(monthwise\_train\_date)

plot(singh\_Mon$table2$forecast, main = "Fuzzy Time Series for A&E Attendances on Every Monday \n With 95% Confidence Interval"

,xlab = "Monday number", ylab = "Number of Attendances",

xlim = c(0,LL\_trainMon), ylim = c(0, max(ts\_mon)),

grid(10,10, lty = 6, col = "cornsilk2"))

grid()

lines(ts\_mon); lines(singh\_Mon$table2$forecast, col='red')

upper <- singh\_Mon$table2$forecast + 1.96\*sqrt(singh\_Mon$accuracy[5]) # singh\_Mon$accuracy = 36.55

lower <- singh\_Mon$table2$forecast - 1.96\*sqrt(singh\_Mon$accuracy[5]) # fit\_Monday$sigma2 = 36.55

polygon(c(time(monthwise\_train\_date),rev(time(monthwise\_train\_date))), c(upper,rev(lower)),

col=rgb(0,0,0.6,0.2), border=FALSE)

out <- (ts\_mon < lower | ts\_mon > upper)