Homework 3 Period Dataset

Pick a time series of your choosing. Then analyze using techniques you have learned in this course and others. Discuss at least three models and evaluate their performance.

**#R Code**

**#Load the data**

library(fpp)

setwd("~/R/PREDICT 413/Homework 3")

P<- read.csv("Period.csv")

summary(P)

Ptimeseries <- ts(P$Days, frequency = 12, start= c(2011,3))

Ptimeseries

**#Timeplot on Full Data**

plot(Ptimeseries, ylab= "Days", xlab= "Year", main= " Average Menstrual Cycle Days")

**#Split Data into Train/Test**

Ptimeseries\_train <-window(Ptimeseries, frequency = 12, start=c(2011, 3), end=c(2016, 8))

Ptimeseries\_train

Ptimeseries\_test <-window(Ptimeseries, frequency = 12, start=c(2016, 9), end=c(2018, 1))

Ptimeseries\_test

**#Model Selection**

**#ETS Model**

fit1 <- ets(Ptimeseries\_train)

summary(fit1)

**#Auto.Arima Pre-work**

tsdisplay(Ptimeseries\_train)

**#Unit Root Tests**

adf.test(Ptimeseries\_train, alternative = "stationary")

kpss.test(Ptimeseries\_train)

**#Number of Differencing Required**

nd <- ndiffs(Ptimeseries\_train)

if(nd > 0) {

xstar <- diff(Ptimeseries\_train,differences=nd)

}

nd

**#Differencing**

kpss.test(diff(Ptimeseries\_train))

tsdisplay(diff(Ptimeseries\_train))

**#Auto.Arima Model**

fit2 <- auto.arima(Ptimeseries\_train)

summary(fit2)

tsdisplay(residuals(fit2))

Box.test(residuals(fit2), fitdf=3, lag=10, type="Ljung")

tsdiag(fit2)

**#Neural Net Model**

fit3 <- nnetar(Ptimeseries\_train)

summary(fit3)

tsdisplay(residuals(fit3))

Box.test(residuals(fit3), fitdf=3, lag=10, type="Ljung")

**#Benchmark**

fit4\_meanf<-meanf(Ptimeseries\_train)

fit5\_naive<-naive(Ptimeseries\_train)

fit6\_rwfdrift<-rwf(Ptimeseries\_train, drift=TRUE)

**#Training Set Accuracy - Summary**

summary(fit1) # training set

summary(fit2) # training set

summary(fit3) # training set

summary(fit4\_meanf) # training set

summary(fit5\_naive) # training set

summary(fit6\_rwfdrift) # training set

**#Training Set Accuracy – Goodness-of-fit**

accuracy(fit1) # training set

accuracy(fit2) # training set

accuracy(fit3) # training set

accuracy (fit4\_meanf) # training set

accuracy (fit5\_naive) # training set

accuracy (fit6\_rwfdrift) # training set

**#Forecast on Test Set**

par(mfrow=c(2,2))

ETS\_ANN<-forecast(fit1, h=length(Ptimeseries\_test))

plot(ETS\_ANN, ylab="Days")

lines(Ptimeseries, col="red",ylab="Actual")

ETS\_ANN

Auto.ARIMA <-forecast(fit2, h=length(Ptimeseries\_test))

plot(Auto.ARIMA, ylab="Days")

lines(Ptimeseries, col="red",ylab="Actual")

Auto.ARIMA

NN <-forecast(fit3, h=length(Ptimeseries\_test))

plot(NN, ylab="Days")

lines(Ptimeseries, col="red",ylab="Actual")

NN

AVERAGE <-meanf(Ptimeseries\_train, h=length(Ptimeseries\_test))

plot(AVERAGE)

lines(Ptimeseries, col="red",ylab="Actual")

print(accuracy(ETS\_ANN, Ptimeseries\_test))

print(accuracy(Auto.ARIMA, Ptimeseries\_test))

print(accuracy(NN, Ptimeseries\_test))

print(accuracy(AVERAGE, Ptimeseries\_test))

par(mfrow=c(1,1))

**#Diagnostics**

**#Box-Ljung test A**

Box.test(ETS\_ANN$residuals, lag=25, type = "Ljung-Box")

par(mfrow=c(2,2))

acf(ETS\_ANN$residuals, lag.max=25)

plot(ETS\_ANN$residuals, ylab = "Residuals")

abline(h = 0, col = "red")

hist(ETS\_ANN$residuals, main = "", xlab = "Residuals")

par(mfrow=c(1,1))

**#Box-Ljung test B**

Box.test(Auto.ARIMA$residuals, lag=25, type = "Ljung-Box")

par(mfrow=c(2,2))

acf(Auto.ARIMA $residuals, lag.max=25)

plot(Auto.ARIMA $residuals, ylab = "Residuals")

abline(h = 0, col = "red")

hist(Auto.ARIMA $residuals, main = "", xlab = "Residuals")

par(mfrow=c(1,1))

**#Forecast of Next 3 Months**

par(mfrow=c(2,2))

fit7 <- ets (Ptimeseries, model ="ANN")

ETS\_ANN<-forecast(fit7, h=3)

ETS\_ANN

plot(ETS\_ANN, ylab="Days")

fit8 <- Arima(Ptimeseries, order=c(0,0,0))

Auto.ARIMA <-forecast(fit8, h=3)

plot(Auto.ARIMA, ylab="Days")

Auto.ARIMA

fit9 <- nnetar(Ptimeseries, order=c(1,1,2))

NN <-forecast(fit9, h=3)

plot(NN, ylab="Days")

NN

par(mfrow=c(1,1))

**Extra Code:**

library("forecast")  
library("fpp")  
library(dplyr)  
library(tidyr)  
sales\_df <- read.csv("data/advertising-and-sales-data-36-co.csv", stringsAsFactors=FALSE)  
  
sales\_df$MonthYear <- sales\_df$Month  
sales\_updated <- separate(sales\_df, col='Month', into = c('year', 'month' ), sep='-')  
  
sales\_series <- ts(sales\_df$Sales, deltat = 1/12, start = c(2001, 1))  
plot(sales\_series, main="Sales Data")  
  
fit\_arima <- auto.arima(sales\_series)  
summary(fit\_arima)  
fit\_ets <- ets(sales\_series)  
summary(fit\_ets)  
  
train = window(sales\_series, start=c(2001, 1), end=c(2003, 6))  
test = window(sales\_series, start=c(2003, 7), end=c(2003, 12))  
  
acf(train)  
  
fit\_arima\_train <- auto.arima(train)  
fit\_arima\_train  
fcast\_arima <- forecast(fit\_arima\_train, h=6)  
print("Arima Test Accuracy")  
arima\_accuracy <- accuracy(fcast\_arima, test)  
arima\_accuracy  
  
fit\_ets\_train <- ets(train)  
fit\_ets\_train  
fcast\_ets <- forecast(fit\_ets\_train, h=6)  
print("ETS Test Accuracy")  
ets\_accuracy <- accuracy(fcast\_ets, test)  
ets\_accuracy  
  
print("Combined forecast")  
forecast\_df <- data.frame(fcast\_arima=as.matrix(fcast\_arima$mean), fcast\_ets=as.matrix(fcast\_ets$mean), actual=as.matrix(test))  
forecast\_df$avg\_forecast <- (forecast\_df$fcast\_arima \* 2 + forecast\_df$fcast\_ets) / 3  
forecast\_df$avg\_diffs <-forecast\_df$avg\_forecast - forecast\_df$actual  
forecast\_df  
print(paste("ME", round(mean(forecast\_df$avg\_diffs),2)))  
print(paste("RMSE", round(sqrt(mean(forecast\_df$avg\_diffs^2)),2)))  
print(paste("MAE", round(mean(abs(forecast\_df$avg\_diffs)),2)))

**Next, I split the data, keeping the last 6 months out of the training.**

# Partition Data into Train and Test Sets

sales.train <- window(sales, end=c(1975,6), frequency = 12)

sales.test <- window(sales, start=c(1975,7), frequency = 12)

Putting two forecasts on one graph and highlighting a section

# Now hold out 6 months of data for a test set and try to forecast using the ETS and the auto.arima

fit.ets <- ets(sales.train)

fit.arima <- auto.arima(sales.train)

fcast.ets <- forecast(fit.ets, 6)

fcast.arima <- forecast(fit.arima, 6)

plot(sales, type='n', ylab='House Sales (in Thousands)',

main='6 mo. forecast for US House Sales, 1965-1975')

rect(1975.5,0,1976.0,80,col = 'lightgrey', border = NA)

lines(sales, col='black')

lines(fcast.ets$mean, col='red', lty=1, lwd=2)

lines(fcast.arima$mean, col='blue', lty=1, lwd=2)

legend('topleft',legend = c("Sales",fit.ets$method,

"ARIMA(1,1,0)(2,0,0)[12]"),

col=c("black","red","blue"),

lty=c(1,1,1), lwd=c(1,2,2))