#Load needed packages

library(forecast)

library(zoo)

library(ggplot2)

library(dplyr)

#Read data

dataCM <- read.csv("/Users/chrismcmanus/Desktop/DSBA 6211 - Advanced Business Analytics /AustralianWines.csv")

# Removing "null" values

dataCM <- na.omit(dataCM)

str(dataCM)

head(dataCM)

#Creating basic plot of dataset

xCM <- ts(dataCM$Red, start=c(1980, 1), frequency=12)

#Viewing data

xCM

#Basic plot of data

plot(xCM, ylab="Sales", xlab="Year")

########################################################################

#Regression-based model

Wine.lm <- tslm(xCM~trend)

summary(Wine.lm)

#Data partition for last two years

WineValidCM <- 24

WineTrainCM <- length(xCM) - WineValidCM

WineTrainCM.ts <- window(xCM, start=c(1980, 1), end=c(1980,WineTrainCM))

WineValidCM.ts <- window(xCM, start=c(1980, WineTrainCM+1), end=c(1980, WineTrainCM+WineValidCM))

########################################################################

#Model 1: Linear Trend

WineTrainCM.lm <- tslm(WineTrainCM.ts~trend)

summary(WineTrainCM.lm)

WineTrainCM.lm.pred <- forecast(WineTrainCM.lm, h=WineValidCM, level=0.95)

#Evaluating Model 1 performance

accuracy(WineTrainCM.lm.pred, WineValidCM.ts)

#Visualizing Model 1

par(mfrow=c(1,1))

plot(WineTrainCM.lm.pred, xlab="Years", ylab="Sales", main = "Model 1: Linear Trend")

lines(WineTrainCM.lm.pred$fitted, lwd=2, col="blue")

lines(WineValidCM.ts)

########################################################################

#Model 2: Seasonality

WineTrainCM.lm.season <- tslm(WineTrainCM.ts~season)

summary(WineTrainCM.lm.season)

WineTrainCM.lm.season.pred <- forecast(WineTrainCM.lm.season, h=WineValidCM, level=0.95)

accuracy(WineTrainCM.lm.season.pred, WineValidCM.ts)

#Visualizing Model 2

par(mfrow=c(1,1))

plot(WineTrainCM.lm.season.pred, xlab="Years", ylab="Sales", main = "Model 2: Seasonality")

lines(WineTrainCM.lm.season.pred$fitted, lwd=2, col="blue")

lines(WineValidCM.ts)

########################################################################

#Model 3: Linear Trend and Seasonality

WineTrainCM.lm.trend.season <- tslm(WineTrainCM.ts ~ + I(trend^2) + season)

summary(WineTrainCM.lm.trend.season)

WineTrainCM.lm.trend.season.pred <- forecast(WineTrainCM.lm.trend.season, h=WineValidCM, level=0.95)

accuracy(WineTrainCM.lm.trend.season.pred, WineValidCM.ts)

#Visualizing Model 3

par(mfrow=c(1,1))

plot(WineTrainCM.lm.trend.season.pred, xlab="Years", ylab="Sales", main = "Model 3: Linear Trend and Seasonality")

lines(WineTrainCM.lm.trend.season.pred$fitted, lwd=2, col="blue")

lines(WineValidCM.ts)

########################################################################

#Model 4: Simple Exponential Smoothing

sesCM <- ses(WineTrainCM.ts, alpha=0.2,h=36)

autoplot(sesCM, ylab="Sales", xlab="Year") +

autolayer(fitted(sesCM), series="Fitted") +

autolayer(WineValidCM.ts)

accuracy(sesCM, WineValidCM.ts)

#Identify optimal alpha parameter

alphaCM <- seq(.01, .99, by=.01)

MAPECM <- NA

for(i in seq\_along(alphaCM)) {

fit <- ses(WineTrainCM.ts, alpha = alphaCM[i], h=36)

MAPECM[i] <- accuracy(fit, WineValidCM.ts)[2,5]

}

# convert to a data frame and identify minimum alpha value

alphaCM.fit <- data\_frame(alphaCM, MAPECM)

alphaCM.min <- dplyr::filter(alphaCM.fit, MAPECM == min(MAPECM))

#Plot MAPE vs. Alpha

ggplot(alpha.fit, aes(alphaCM, MAPECM)) + geom\_line() + geom\_point(data=alphaCM.min, aes(alphaCM, MAPECM), size=2, color="blue")

#Model 4: Simple Exponential Smoothing with new alpha

sesCM <- ses(WineTrainCM.ts, alpha=0.14,h=36)

autoplot(sesCM, ylab="Sales", xlab="Year", main="Model 4: Simple Exponential Smoothing when Alpha=0.14") +

autolayer(fitted(sesCM), series="Fitted") +

autolayer(WineValidCM.ts)

accuracy(sesCM, WineValidCM.ts)