

```
# Getting Data and plotting
setwd("C:/Users/coffee/Downloads/Software/R/Cowpertwait/IF")
data = read.csv("data.csv", header = FALSE)
data.ts = ts(data, start = 2001, frequency = 12)
data.ts
plot.ts(data.ts)

# time series decomposition (additive), into TREND, SEASON and RANDOM ERRORS

a = decompose(data.ts, type="additive")
#par(mfrow = c(4,1))
plot(a$x)
plot(a$trend)
plot(a$seasonal)
plot(a$random)
plot(a)

#par(mfrow = c(1,1))
ts.plot(cbind(a$trend, a$trend + a$seasonal), lty = 1:2)

# time series decomposition (multiplicative), into TREND, SEASON and RANDOM ERRORS

b = decompose(data.ts, type="mult")
##par(mfrow = c(4,1))
plot(b$x)
plot(b$trend)
plot(b$seasonal)
plot(b$random)
plot(b)

#par(mfrow = c(1,1))
ts.plot(cbind(b$trend, b$trend * b$seasonal), lty = 1:2)

# Compare additive and multiplicative

#par(mfrow = c(2,1))
ts.plot(cbind(a$trend, a$trend + a$seasonal), lty = 1:2)
ts.plot(cbind(b$trend, b$trend * b$seasonal), lty = 1:2)

# AUTOCORRELATION with lag 1 and 2 of original data
#par(mfrow = c(1,1))
acf(data)
acf(data)$acf[2]
acf(data)$acf[3]

# Lag 1 plot for autocorrelation
#par(mfrow = c(2,1))
plot(data[1:180,], data[2:181,])
# Lag 2 plot for autocorrelation
plot(data[1:179,], data[3:181,])

# Correlogram of Random component of decomposed series
#par(mfrow = c(1,1))
acf(a$random[7:175])
# Autoregressive model of order 2 ( further covered)
# Seasonal adjustment is effective by seeing decrease in Standard deviation
# SD of data
sd(data[7:175,])
# SD of data - trend
sd(data[7:175,] - a$trend[7:175])
#SD of random
sd(a$random[7:175])

# Exponential Smoothing - assumes no seasonal effects and trend
```

```
data.hw1 = HoltWinters(data.ts, beta=0, gamma = 0)
data.hw1
plot(data.hw1)
# One Step ahead minimum error and corresponding alpha chosen by R
data.hw1$SSE
data.hw1$alpha

# chosen alpha value = 1/n
p = 1/181
data.hw2 = HoltWinters(data.ts, alpha = p, beta=0, gamma = 0)
data.hw2
plot(data.hw2)
# One Step ahead minimum error and corresponding alpha chosen
data.hw2$SSE
data.hw2$alpha

# chosen alpha value = 0.2
data.hw3 = HoltWinters(data.ts, alpha = 0.2, beta=0, gamma = 0)
data.hw3
plot(data.hw3)
# One Step ahead minimum error and corresponding alpha chosen
data.hw3$SSE
data.hw3$alpha

# HOLT WINTERS - including level, slope and seasonal effects - ADDITIVE

data.hw4 = HoltWinters(data.ts, seasonal = "additive")
data.hw4
data.hw4$coef
data.hw4$SSE
plot(data.hw4$fitted)
plot(data.hw4)

# HOLT WINTERS - including level, slope and seasonal effects - MULTIPLICATIVE

data.hw5 = HoltWinters(data.ts, seasonal = "mult")
data.hw5
data.hw5$coef
data.hw5$SSE
plot(data.hw5$fitted)
plot(data.hw5)

# HOLT WINTERS - including level, slope and seasonal effects - ADDITIVE

data.hw6 = HoltWinters(data.ts, alpha = 0.2, beta = 0.2, gamma = 0.2, seasonal = "mult")
data.hw6
data.hw6$coef
data.hw6$SSE
plot(data.hw6$fitted)
plot(data.hw6)

# HOLT WINTERS - including level, slope and seasonal effects - ADDITIVE

data.hw7 = HoltWinters(data.ts, alpha = 0.2, beta = 0.2, gamma = 0.2, seasonal =
"additive")
data.hw7
data.hw7$coef
data.hw7$SSE
plot(data.hw7$fitted)
plot(data.hw7)
acf(residuals(data.hw7))

# ADDITIVE gives least SSE, PREDICTION using ADDITIVE
data.hw4 = HoltWinters(data.ts, seasonal = "additive")
plot(data.hw4)
```

```

data.predict1 = predict(data.hw4, n.ahead = 6*12)
ts.plot(data.ts, data.predict1, lty = 1:2)
acf(residuals(data.hw4))
# data.predict2 = predict(data.hw5, n.ahead = 6*12)
# ts.plot(data.ts, data.predict2, lty = 1:2)
# data.predict3 = predict(data.hw6, n.ahead = 6*12)
# ts.plot(data.ts, data.predict3, lty = 1:2)
# data.predict4 = predict(data.hw7, n.ahead = 6*12)
# ts.plot(data.ts, data.predict4, lty = 1:2)

# Calculating if residual series (data.ts - data.hw4$fitted[,1]) is white noise or not
residual_series = data.ts - data.hw4$fitted[,1]
mean(residual_series)
var(residual_series)
acf(residual_series)
# variance = 1.5, mean = 0, and correlogram shows it resembles white noise.
acf(diff(a$random[7:175]))
# first order difference of random walk are white noise
# shows that random errors follows random walk
# head(a$random[7:175])
# head(diff(a$random[7:175]))
acf(diff(data.ts))
# first order differences also shows no significant value at lag 1 but other significant
values shows the model needs some extension.
data.hw4_2 = HoltWinters(data.ts, gamma=0)
acf(resid(data.hw4_2))
# better fit using gamma =0

# Fitted AR
layout(1:1)
data.ar = ar(data.ts)
mean(data.ts)
data.ar$order
data.ar$ar
data.ar$ar + c(-1.96,1.96)*sqrt(data.ar$asy.var)
acf(data.ar$resid[-(1:data.ar$order)])
pacf(data.ar$resid[-(1:data.ar$order)])

#method = MLE, only take vector i.e first column
data.ar2 = ar(data.ts[,1], method = "mle")
mean(data.ts)
data.ar2$order
data.ar2$ar
acf(data.ar2$resid[-(1:data.ar2$order)])
pacf(data.ar2$resid[-(1:data.ar2$order)])

# fitting moving aveergae
data.ma = arima(data.ts, order = c(0,0,1))
data.ma
acf(data.ma$res[-1])
# not a satisfactory fit
# com#parison of ARMA, AR, and MA
data.ARvsARMA = arima(data.ts, order = c(1,0,0))
data.MAvsARMA = arima(data.ts, order = c(0,0,1))
data.ARMA = arima(data.ts, order = c(1,0,1))
AIC(data.ARvsARMA)
AIC(data.MAvsARMA)
AIC(data.ARMA)
data.ARMA
acf(resid(data.ARMA))
acf(resid(data.ARvsARMA))
acf(resid(data.MAvsARMA))
# both AR and ARMA are good approximate
predict.arma = predict(data.ARMA, n.ahead = 48)
ts.plot(cbind(data.ts, predict.arma$pred+predict.arma$se), lty = 1:2)

```

```
# difference
plot(diff(data.ts))
plot(diff(data.ts, d = 2))
plot(diff(data.ts, d = 3))
plot(diff(data.ts, d = 4))
plot(diff(data.ts, d = 12))
#ARIMA
data.ima = arima(data.ts, order = c(0,1,1))
data.ima
acf(resid(data.ima))
pacf(resid(data.ima))

data.ima_predict = predict(data.ima, n.ahead = 48)
data.ima_predict

ts.plot(cbind(data.ts, data.ima_predict$pred+data.ima_predict$se), lty = 1:2)
library(forecast)
data.ima_forecast = forecast(data.ima)
data.ima_forecast
plot(data.ima_forecast)
ts.plot(data.ts,data.ima_forecast$fitted, lty=1:2)
```