What the Holt?!

Joshua Bela

December 2, 2019

1. For the time series data set with seasonality – start with the raw data before differencing here please!!!

# import libraries  
library(lubridate) #month() year()

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

library(tseries)#kpss.test()

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(dplyr)#select()

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':  
##   
## intersect, setdiff, union

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# library(aTSA)#adf.test(); arch.test()  
# library(forecast)#arima(); forecast.Arima()  
# library(stats)#Box.test(x, lag = 1, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)  
library(rucm)

## Loading required package: KFAS

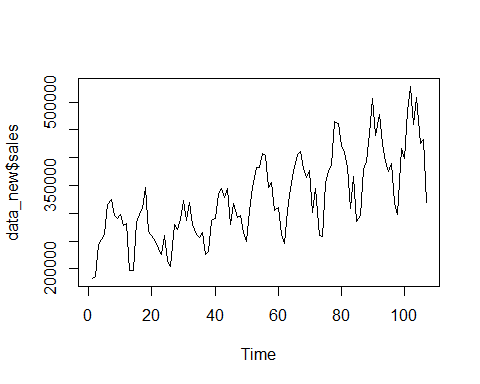
# import data  
data = read.csv("data/State\_time\_series.csv")  
# filter variables  
data\_sub = data %>% select(Date, Sale\_Counts)  
# filter NA values  
data\_sub = data\_sub[!is.na(data\_sub$Date) & !is.na(data\_sub$Sale\_Counts),]  
# write.csv(data\_sub, "data/output.csv")

Monthify Data

monthify = function(data){  
 month = year = NULL  
 sales = NULL  
 for(i in 1:nrow(data)){  
 month\_local = data$Date[i] %>% month  
 year\_local = data$Date[i] %>% year()  
   
 if(sum(  
 c(month\_local, year\_local) == c(month, year)  
 ) == 2){  
 sales[length(sales)] = sales[length(sales)] + data$Sale\_Counts[i]  
 }  
 else{  
 sales[length(sales) + 1] = data$Sale\_Counts[i]  
 month = month\_local  
 year = year\_local  
 }  
 }  
 sales %>% return()  
}  
  
data\_new = monthify(data\_sub)  
data\_new = data\_new[-length(data\_new)]  
data\_new = data\_new[11:length(data\_new)]  
data\_new = data.frame(  
 sales = data\_new,  
 time = 1:length(data\_new)  
)

1. Plot out the time series and

# convert to time series object  
ts1 = ts(data\_new$sales, frequency = 12)  
  
# plot time series  
plot.ts(data\_new$sales)

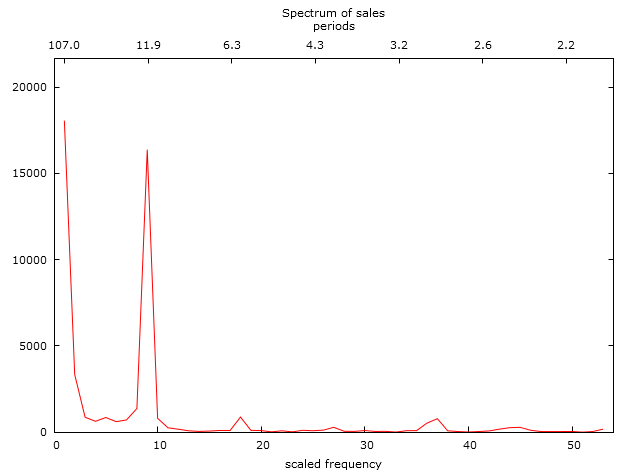


suggest whether a type 1, type 2 or type 3 Holt Winter model should be applied and why. **Because trend and seasonality apply, we should use type 3.**

1. Eyeball the size of the period. What do you think it might be? Why is that?

**There appear to be 9 periods in the plot above. As expected, this suggests annual seasonality. Why are these prices seasonal? Prices of many various goods and services have annual seasonality. I’m not sure why. It could have something to do with the psychology of New Year celebration or with Christmas consumerism.**

1. Use GRETL to do a periodogram for the data. What does the periodogram suggest might be the period length for the data?



**The period length is 12 months.**

1. If a type 2 or type 3 model, then apply a KPSS or ADF test to test for trend.

kpss.test(ts1)

## Warning in kpss.test(ts1): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ts1  
## KPSS Level = 1.6114, Truncation lag parameter = 4, p-value = 0.01

**Based on the p-value of the KPSS test, there is a trend.**

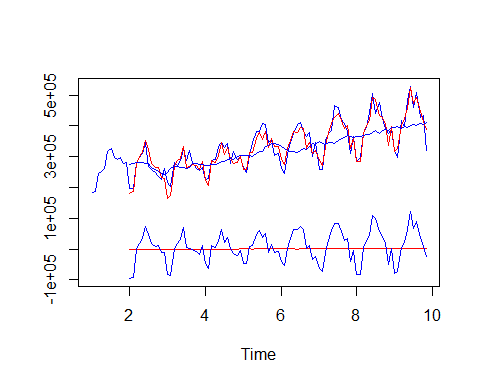
1. Decide the weights you will use for the three components of Winter Holt smoothing – constant, trend and seasonality (gamma) – why these values? If you are using SAS then read about these weights in the proc docs, otherwise fish around in the R or Python docs.

model = HoltWinters(ts1)

**In the HoltWinters() function for R, leaving NULL values for these components allows the learning algorithm to automatically optimize them, so I let the algorithm determine the optimal values.**

1. Run the Holt Winter model and then, using sgplot or your other favorite plotting poison, plot the actual data and the fitted/forecast data on the same graph.

model$x %>% ts.plot(  
 model$fitted,  
 col = c("blue", "red")  
)



How did the Holt-Winter model do in terms of forecasting?

**It looks fantastic! Great job on the learning algorithm to optimize the parameters for trend and seasonality. Quite robust function.**

1. Next run the same data set using the Unobserved Components Model time series analytic technique. Interpret the significance analysis of components table (based on final components in terms of trend, irregular (ARMA) and seasonality components in the data set – that is, which components are statistically significant?

temp = data\_new  
temp$sales = temp$sales/1000  
# write.csv(temp, "sales\_monthly.csv")  
  
model2 <- ucm(formula = sales ~ 0, data = temp, level = T, irregular = T,  
 slope = T, season = T, season.length = 12)  
model2

## Call:  
## ucm(formula = sales ~ 0, data = temp, irregular = T, level = T,   
## slope = T, season = T, season.length = 12)  
##   
## Parameter estimates:  
## NULL  
##   
## Estimated variance:  
## Irregular\_Variance Level\_Variance Slope\_Variance Season\_Variance   
## 0.0060 202.6457 0.0002 89.6489

**Based on the infinitessimal p-values, trend, irregularity, and seasonality are all significant.**

1. Have the UCM model produce fitted values for the existing data and forward 12 periods into the future and plot the original time series as well as the fitted/forecast data as well.

fitted = rep(NA, 107)  
predictions = predict(model2$model, n.ahead = 12\*12)  
  
NEW = c(fitted, predictions)  
ORIG = NEW  
ORIG[1:nrow(temp)] = temp$sales  
  
x = 1:length(NEW)  
plot(x, ORIG,  
main="Sales, Monthly",  
ylab="",  
type="l",  
col="blue")  
lines(x,NEW, col="red")  
legend("topleft",  
c("Original","Forecast"),  
fill=c("blue","red")  
)

