HW3

Will

7 de junio de 2017

HW3

Question 1

A situation in my work where I might use exponential smoothing could be inflation forecast. We use CPI as a input for expected rate of return. Monthly CPI tend to looks seasonally with respect to its history. So I could use a algoritm to determinate the best smoothing parameter, choises the lower square error. If that not the case I think that I could use 0.5.

Question 2

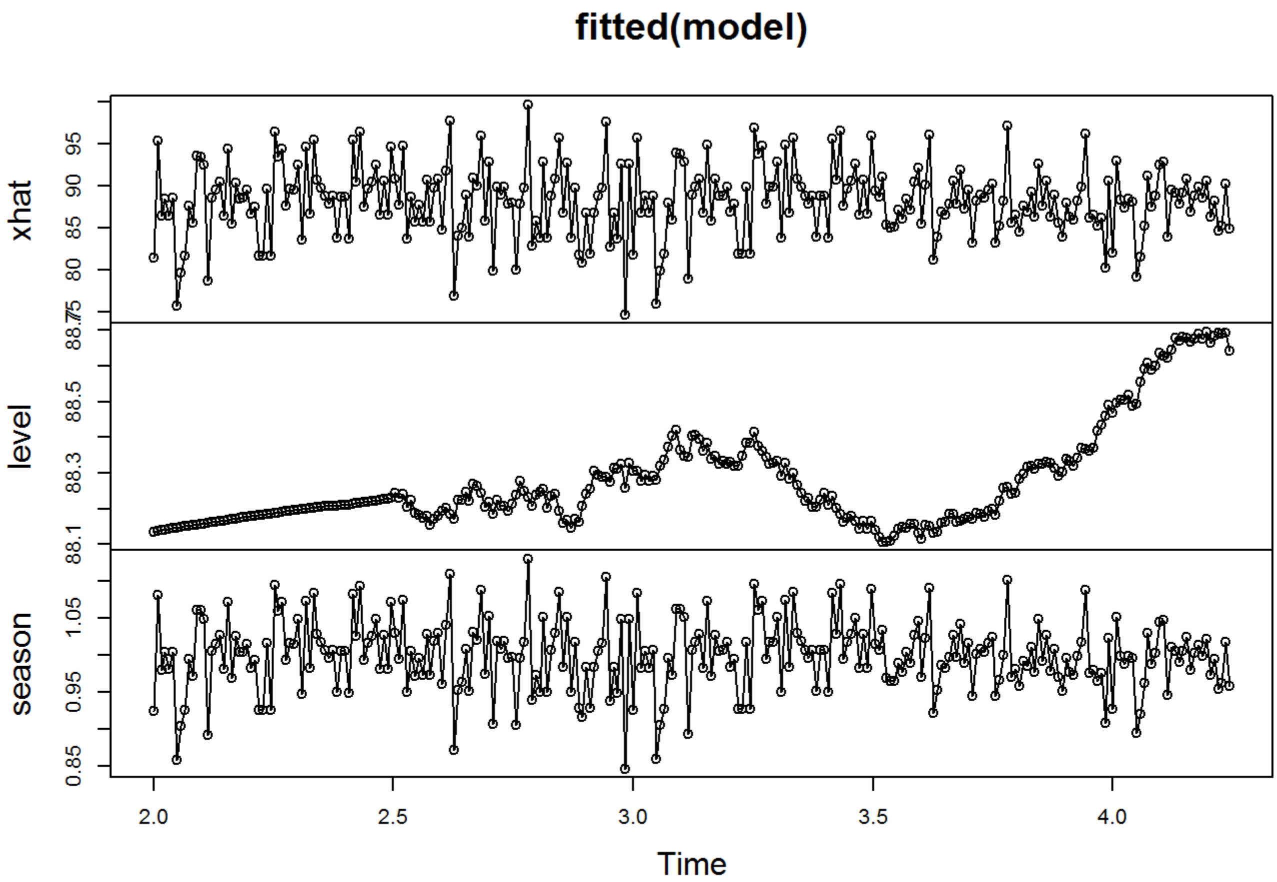
```
library(forecast)



setwd("C:/Users/ce02144/Documents/HW3")
#Reading data
weather <- read.table("temps2.txt", header = TRUE)

#Converting to timeserie
weather.ts <- ts(weather[,2:21],
                 frequency = 123)

#Create model
model = HoltWinters(x = weather.ts, beta = FALSE, seasonal = "multiplicative")

#Exploring results
model
```

```
## Holt-Winters exponential smoothing without trend and with multiplicative seasonal component.
##
## Call:
## HoltWinters(x = weather.ts, beta = FALSE, seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 0.003602305
##  beta : FALSE
##  gamma: 0.3349119
##
## Coefficients:
##            [,1]
## a    88.6737351
## s1    1.0511398
## s2    1.0471550
## s3    1.0550725
## s4    0.9724155
## s5    1.0218876
## s6    1.0215369
## s7    1.0062241
## s8    0.9838796
## s9    1.0256818
## s10   0.9992231
## s11   1.0479930
## s12   1.0034866
## s13   0.9964509
## s14   1.0153821
## s15   0.9702652
## s16   1.0078804
## s17   0.9702829
## s18   1.0263794
## s19   0.9729813
## s20   0.9731394
## s21   1.0477167
## s22   1.0103373
## s23   1.0669926
## s24   1.0069094
## s25   1.0221226
## s26   1.0140726
## s27   1.0254200
## s28   1.0066853
## s29   1.0034566
## s30   1.0068381
## s31   1.0586770
## s32   0.9942702
## s33   0.9927694
## s34   1.0331662
## s35   0.9715186
## s36   0.9802569
## s37   0.9839578
## s38   0.9957634
## s39   0.9728097
## s40   1.0168391
## s41   0.9880801
## s42   0.9983103
## s43   1.0299514
## s44   1.0095804
## s45   1.0181410
## s46   1.0675946
## s47   0.9244345
## s48   0.9790192
```

```
## s49   0.9860974
## s50   1.0063036
## s51   0.9970861
## s52   1.0023613
## s53   1.0011952
## s54   1.0464975
## s55   0.9959979
## s56   1.0097277
## s57   0.9621225
## s58   0.9959313
## s59   1.0010061
## s60   1.0211570
## s61   1.0211142
## s62   1.0076132
## s63   0.9807728
## s64   1.0028275
## s65   1.0025118
## s66   1.0777556
## s67   0.9750916
## s68   1.0204169
## s69   0.9708398
## s70   1.0125174
## s71   0.9885081
## s72   1.0025759
## s73   0.9910751
## s74   1.0502752
## s75   0.9974947
## s76   1.0239587
## s77   0.9611695
## s78   0.9848963
## s79   0.9827427
## s80   0.9849916
## s81   0.9921234
## s82   0.9610377
## s83   0.9956130
## s84   1.0282449
## s85   1.0139620
## s86   1.0840655
## s87   0.9826990
## s88   1.0306210
## s89   0.9820241
## s90   1.0005204
## s91   0.9364654
## s92   0.9987285
## s93   0.9532717
## s94   1.0624530
## s95   0.9971866
## s96   1.0011559
## s97   0.9671295
## s98   1.0026525
## s99   0.9499695
## s100  0.9559687
## s101  0.9796541
## s102  1.0060623
## s103  1.0009059
## s104  1.0408399
## s105  1.0345199
## s106  1.0408350
## s107  0.9694587
## s108  1.0458977
## s109  0.9977161
## s110  1.0027367
```

```
## s111  1.0014566
## s112  1.0140479
## s113  0.9875947
## s114  1.0176157
## s115  0.9990346
## s116  1.0194243
## s117  0.9891841
## s118  0.9907493
## s119  1.0055288
## s120  0.9516815
## s121  0.9641723
## s122  0.9641715
## s123  0.9880533
```

```
plot(fitted(model), plot.conf=FALSE, type="o", fcol="white", xlab="Time")
```



The level in fitted results shows that end of summer has been gotten later in recent years. However, taking into account the 20year history that trend is not very clear.

Question 3

A linear regression problem for could be *Colombian Currency Forecast*

As a predictors I could use:
CPI actual
CPI Forecast
Central Bank Rate actual
Central Bank Forecast
GDP actual
Consumer Sentiment
US vs Colombian bonds yield spread Price of insurance cost for colombian goverment debt

Question 4

```r
library(ggplot2)
library(MASS)
library(leaps)
library(relaimpo)
```

```
## Loading required package: boot
```

```
## Loading required package: survey
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:boot':
##
##     aml
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
## Loading required package: mitools
```

```
## This is the global version of package relaimpo.
```

```
## If you are a non-US user, a version with the interesting additional metric pmvd is available
```

```
## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

```r
setwd("C:/Users/ce02144/Documents/HW3")

#Reading data
uscrime <- read.table("uscrime.txt", header = TRUE)

#Creating model
model = lm(Crime~., data = uscrime)

#Analyzing model
summary(model)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

```
coefficients(model)
```

```
##   (Intercept)             M            So            Ed           Po1
## -5.984288e+03  8.783017e+01 -3.803450e+00  1.883243e+02  1.928043e+02
##           Po2            LF           M.F           Pop            NW
## -1.094219e+02 -6.638261e+02  1.740686e+01 -7.330081e-01  4.204461e+00
##            U1            U2        Wealth          Ineq          Prob
## -5.827103e+03  1.677997e+02  9.616624e-02  7.067210e+01 -4.855266e+03
##          Time
## -3.479018e+00
```

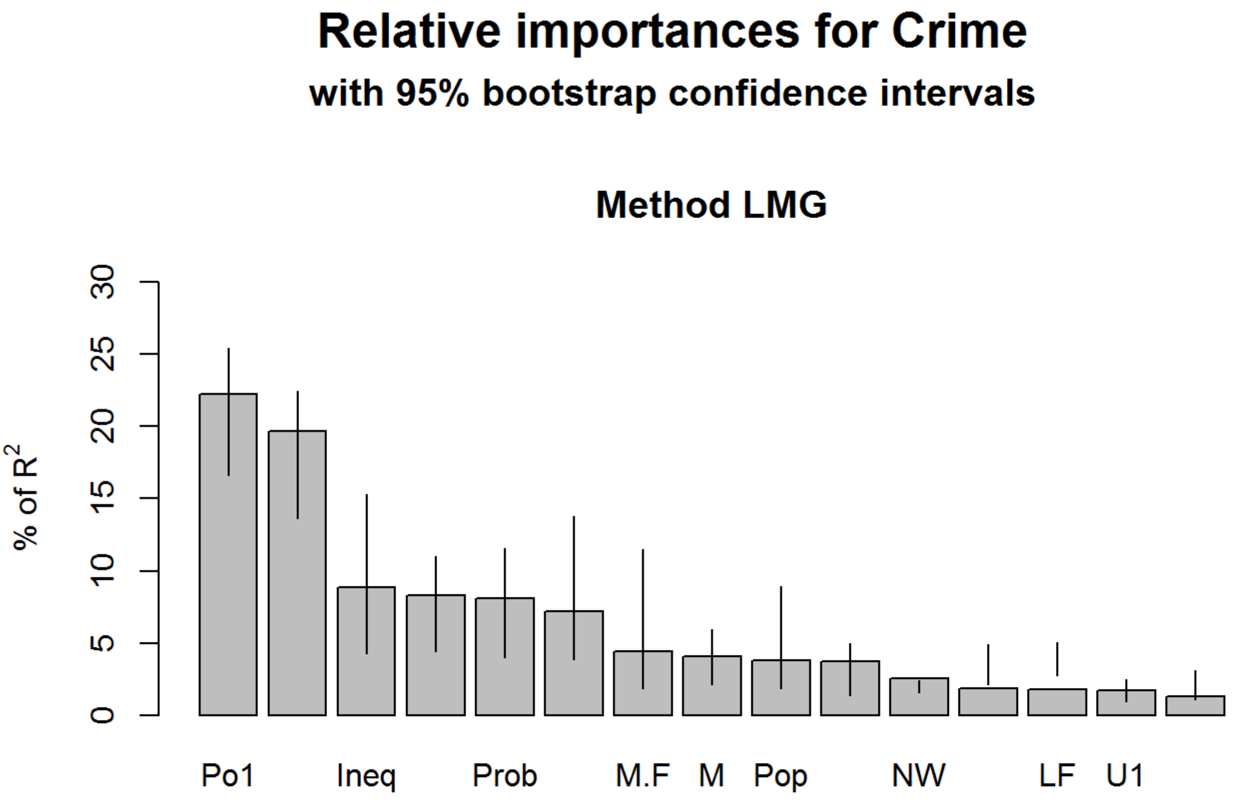
```
AIC(model)
```

```
## [1] 650.0291
```

```
BIC(model)
```

```
## [1] 681.4816
```

```
#Calculing the relative importance of predictors
# Bootstrap Measures of Relative Importance (5 samples)
boot <- boot.relimp(model, b = 5, type = c("lmg"), rela = TRUE)
plot(booteval.relimp(boot,sort = TRUE), las = 2, cex.axis = 0.5, srt = 60, adj = 1)
```

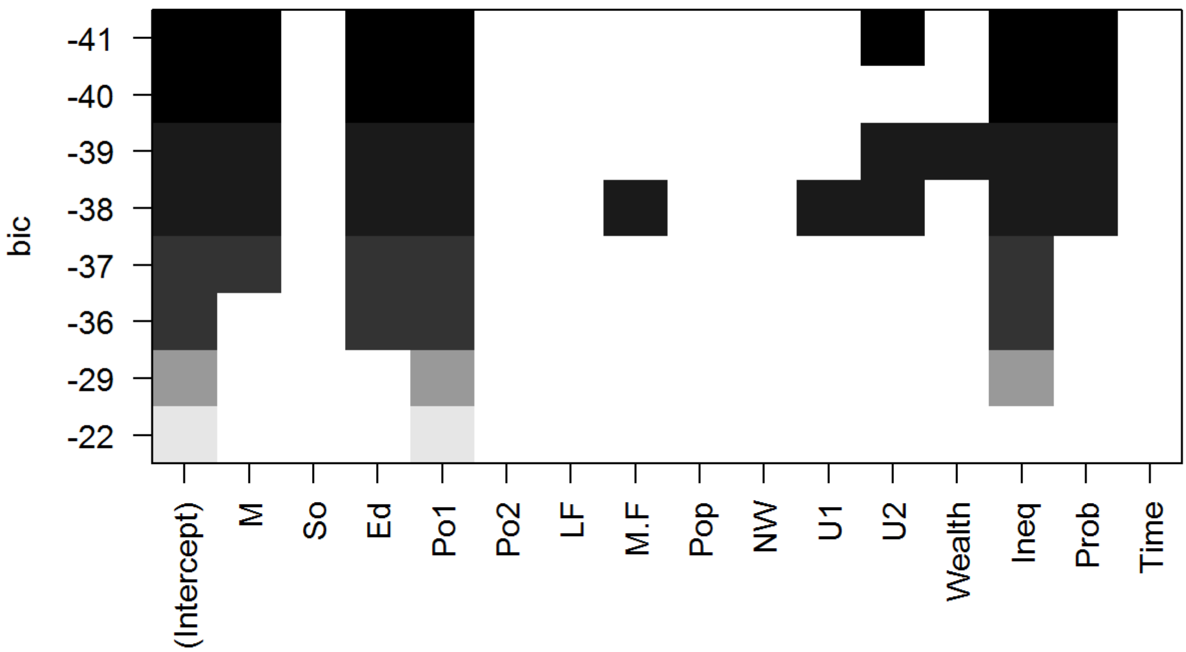


$R^2 = 80.31\%$, metrics are normalized to sum 100%.

```r
#Prediction test
M = 14.0
So = 0
Ed = 10.0
Po1 = 12.0
Po2 = 15.5
LF = 0.640
M.F = 94.0
Pop = 150
NW = 1.1
U1 = 0.120
U2 = 3.6
Wealth = 3200
Ineq = 20.1
Prob = 0.04
Time = 39.0

question4 <- data.frame(M,
                        So,
                        Ed,
                        Po1,
                        Po2,
                        LF,
                        M.F,
                        Pop,
                        NW,
                        U1,
                        U2,
                        Wealth,
                        Ineq,
                        Prob,
                        Time
                        )

#crime prediction
predict(model, newdata = question4,type = "response" )
```

```
##        1
## 155.4349
```

```r
#we could look for better model using leaps library
#nbest best model for each combination.

leaps <- regsubsets(Crime~., data = uscrime, nbest = 1)

# Plot alternative models by bic. There are same predictors but different intercepts

plot(leaps, scale = "bic")
```




```
# Plot alternative models by bic. There are same predictors but different intercepts
plot(leaps, scale = "r2")
```