Week 3 - Homework

Alessio Benedetti

04 june 2018

Question 7.1

Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of ?? (the first smoothing parameter) to be closer to 0 or 1, and why?

We can immagine to apply the exponential smoothing method in the elections field. By using the historical votes receivied over time by a particular party, we can build a model to predict the future evolution of votes for that party. In such an example I would immagine high randomness, due to voters intentions, so an alpha parameter near to 0.

Question 7.2

Using the 20 years of daily high temperature data for Atlanta (July through October) from Question 6.2 (file temps.txt), build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years.

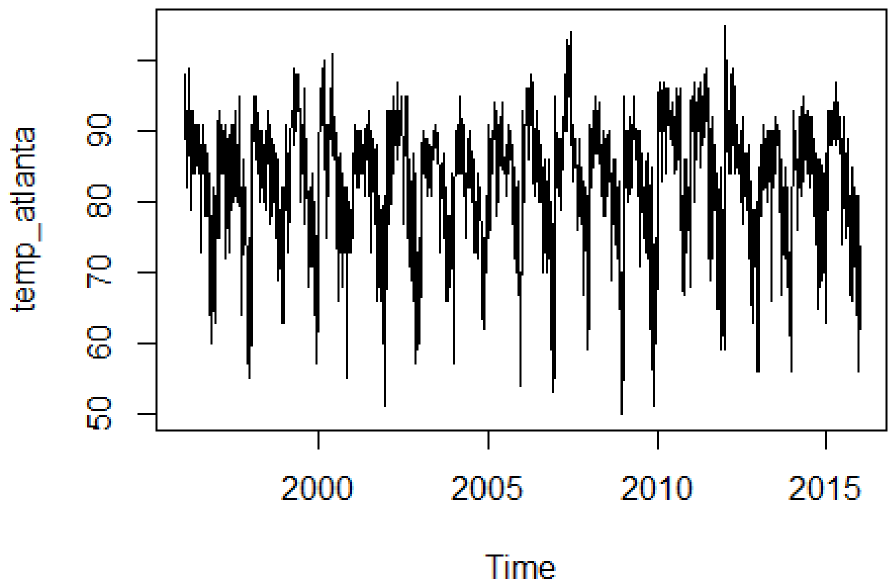
First we need to load the libraries and the data from the temp *txt* file.

```
#install.packages('tseries')
library(tseries)
#install.packages('forecast')
library(forecast)
raw_data <- read.table('7.2tempsSummer2018.txt', header=TRUE)
head(raw_data) #view top  rows of dataset
```

```
##     DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006
## 1 1-Jul    98    86    91    84    89    84    90    73    82    91    93
## 2 2-Jul    97    90    88    82    91    87    90    81    81    89    93
## 3 3-Jul    97    93    91    87    93    87    87    87    86    86    93
## 4 4-Jul    90    91    91    88    95    84    89    86    88    86    91
## 5 5-Jul    89    84    91    90    96    86    93    80    90    89    90
## 6 6-Jul    93    84    89    91    96    87    93    84    90    82    81
##   X2007 X2008 X2009 X2010 X2011 X2012 X2013 X2014 X2015
## 1    95    85    95    87    92   105    82    90    85
## 2    85    87    90    84    94    93    85    93    87
## 3    82    91    89    83    95    99    76    87    79
## 4    86    90    91    85    92    98    77    84    85
## 5    88    88    80    88    90   100    83    86    84
## 6    87    82    87    89    90    98    83    87    84
```

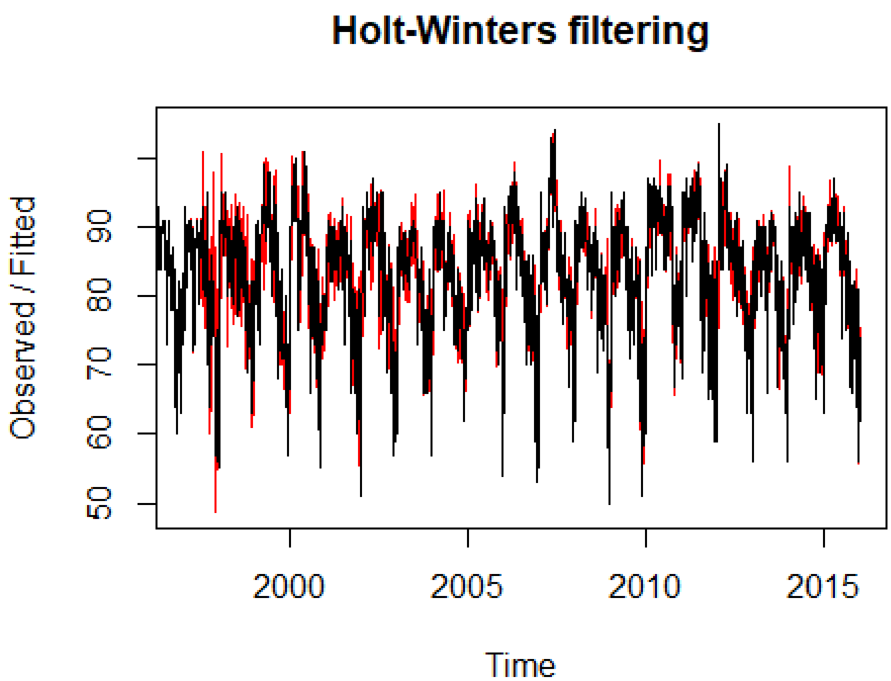
We can now plot the data.

```r
temp_atlanta <- as.vector(unlist(raw_data[,2:21]))
temp_atlanta <- ts( temp_atlanta, start = 1996, frequency = 123 )
plot.ts(temp_atlanta)
```



Now we can apply the exponential smoothing model via the Holt Winters algorithm.

```r
temp_HW_model <- HoltWinters(temp_atlanta)
plot(temp_HW_model)
```

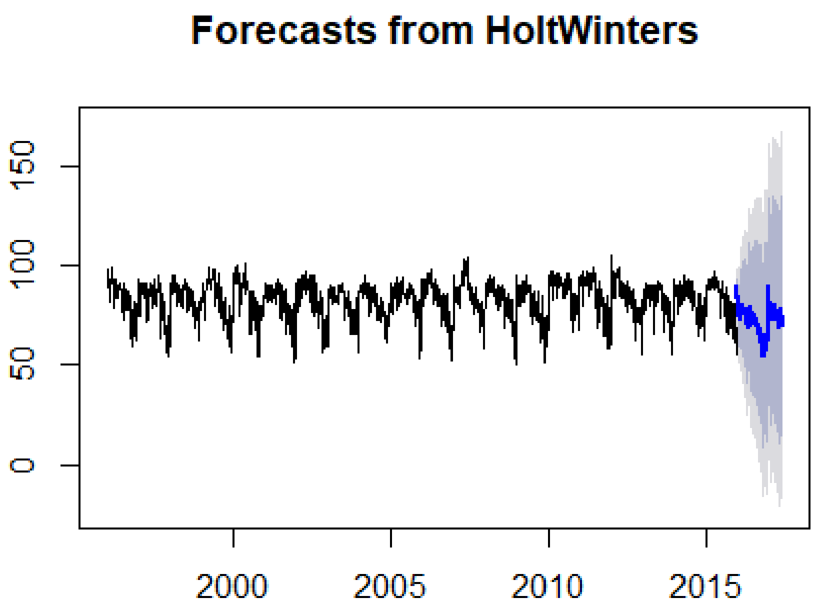


By typing *temp_HW_model* we're able to see the informations about the smoothing parameters:
Smoothing parameters: alpha: 0.6610618 beta : 0 gamma: 0.6248076

Since our alpha is 0,66 and is closer to 1 rather than 0, we can understand that there were not much randomness in the system and therefore the historical temperatures observations "weight" more in the model.

We can now use the model to predict the future progress of the temperatures by using the forecast function (blue in the next figure) as well as the 80% and 90% confidence intervals.

```
temp_atlanta_forecast <- predict(temp_HW_model, n.ahead = 90, prediction.interval
= TRUE)
plot(forecast(temp_HW_model, h=180))
```



Question 8.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

We can apply the linear regression to evaluate the fuel consumption of a car. As predictors we can consider the type of transmission, the type of tires, the type of fuel, the weight of the car and its drag coefficient.

Question 8.2

Using crime data from file uscrime.txt, use regression (a useful R function is lm or glm) to predict the observed crime rate in a city with the following data: 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. Show your model (factors used and their coefficients), the software output, and the quality of fit.

First we need to load the libraries and the data from the temp *txt* file.

```
#install.packages('caret')
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2
```

```r
raw_data <- read.table('8.2uscrimeSummer2018.txt', header=TRUE)
head(raw_data) #view top  rows of dataset
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6
##       Prob    Time Crime
## 1 0.084602 26.2011   791
## 2 0.029599 25.2999  1635
## 3 0.083401 24.3006   578
## 4 0.015801 29.9012  1969
## 5 0.041399 21.2998  1234
## 6 0.034201 20.9995   682
```

We can start by building an initial model where we use all of the predictors, in order to evaluate (based on their respective p values) the ones that can be removed from the model.

```r
model_all_base <- lm(Crime ~ ., raw_data)
summary(model_all_base)
```

```
## 
## Call:
## lm(formula = Crime ~ ., data = raw_data)
## 
## 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
```

Based on the output of the first regression we can evaluate other models by playing on diferent combinations of attributes. To do so we'll use the caret package and the k fold cross validation algorithm with an *svmLinear* method.

```r
# Define train control for k fold cross validation
train_control <- trainControl(method="cv", number=10)
# Fit the models
model_1 <- train(Crime~ M + Ed + Po1 + U2 + Ineq + Prob, data=raw_data,
trControl=train_control, method="svmLinear")
model_2 <- train(Crime~ M + Ed + U2 + Ineq + Prob, data=raw_data,
trControl=train_control, method="svmLinear")
model_3 <- train(Crime~ M + Ed + Po1 + Ineq + Prob, data=raw_data,
trControl=train_control, method="svmLinear")
model_4 <- train(Crime~ M + Ed + Ineq + Prob, data=raw_data,
trControl=train_control, method="svmLinear")
model_5 <- train(Crime~ M + Ed + Ineq, data=raw_data, trControl=train_control,
method="svmLinear")
# Summarise Results
print(model_1)

## Support Vector Machines with Linear Kernel
##
## 47 samples
##  6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 44, 42, 42, 42, 42, 41, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   197.9349  0.7327533  165.2376
##
## Tuning parameter 'C' was held constant at a value of 1

print(model_2)

## Support Vector Machines with Linear Kernel
##
## 47 samples
##  5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42, 42, 42, 42, 43, 43, ...
## Resampling results:
##
```

```
##   RMSE      Rsquared   MAE     
##   351.9147  0.3175334  271.6161
## 
## Tuning parameter 'C' was held constant at a value of 1
```

```r
print(model_3)
```

```
## Support Vector Machines with Linear Kernel 
## 
## 47 samples
##  5 predictor
## 
## No pre-processing
## Resampling: Cross-Validated (10 fold) 
## Summary of sample sizes: 42, 43, 42, 42, 43, 42, ... 
## Resampling results:
## 
##   RMSE      Rsquared   MAE     
##   219.5367  0.6812571  178.2163
## 
## Tuning parameter 'C' was held constant at a value of 1
```

```r
print(model_4)
```

```
## Support Vector Machines with Linear Kernel 
## 
## 47 samples
##  4 predictor
## 
## No pre-processing
## Resampling: Cross-Validated (10 fold) 
## Summary of sample sizes: 41, 42, 43, 44, 42, 43, ... 
## Resampling results:
## 
##   RMSE      Rsquared   MAE     
##   345.2311  0.3470286  293.0245
## 
## Tuning parameter 'C' was held constant at a value of 1
```

```r
print(model_5)
```

```
## Support Vector Machines with Linear Kernel 
## 
## 47 samples
##  3 predictor
## 
## No pre-processing
## Resampling: Cross-Validated (10 fold) 
## Summary of sample sizes: 44, 42, 42, 41, 42, 41, ... 
## Resampling results:
## 
##   RMSE      Rsquared   MAE
```

```
##   374.4052  0.3645608  285.0746
##
## Tuning parameter 'C' was held constant at a value of 1
```

By examing the Rsquared parameter we see that the highest value is obtained with the *model_1* where the predictors are M, Ed, Po1, U2, Ineq and Prob.

FInally we can use our model to predict the observed crime rate in a city where the data are:

```
new_city_data <- data.frame(M = 14.0, Ed = 10.0, Po1 = 12.0, U2 = 3.6, Ineq =
20.1, Prob = 0.04)
predict(model_1, new_city_data)
```

```
##        1
## 1301.432
```