Hw-3

Jacob Lee

5 June 2017

Homework 3

Question 1

Our companies monthly revenues could be modeled via an exponential smoothing model. There is both a relationship to the previous level, the revenue trends and the seasonality (time of year) that determine the level of incomings.

Question 2

For this question I perfomed an analysis on average monthly data and on the daily data. I selected what I thought appropriate levels of alpha, beta and gamma given the levels and then plotted a one year forecast.

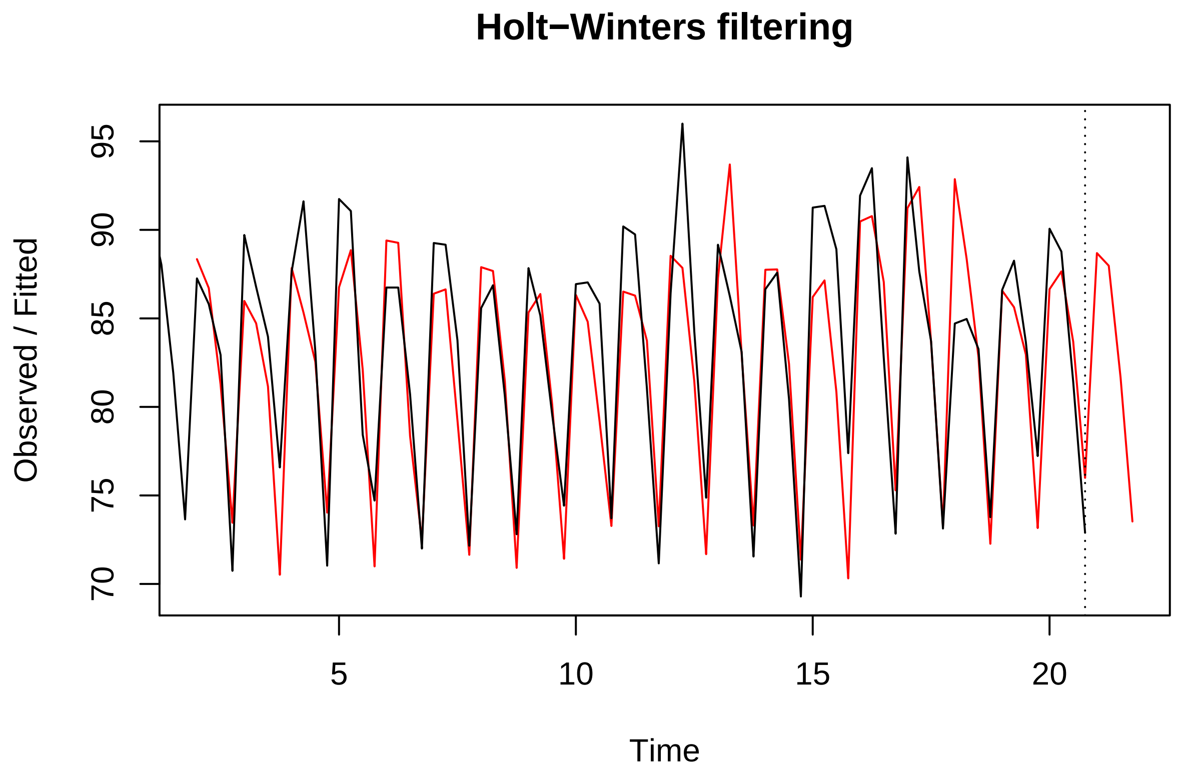
```r
library('smooth')
```

```
## This is package "smooth", v1.9.9
```

```r
temps = read.table('temps.txt', header = TRUE)
row.names(temps) <- temps$DAY
temps <- temps[2:21]

Monthly <- data.frame(M1=as.numeric(character()),
                      M2=as.numeric(character()),
                      M3=as.numeric(character()),
                      M4=as.numeric(character()))
mnths <- vector()
for(i in 1:20) {
  M1 <- mean(temps[1:31,i])
  M2 <- mean(temps[32:62,i])
  M3 <- mean(temps[63:92,i])
  M4 <- mean(temps[93:123,i])
  mnths <- c(mnths,M1,M2,M3,M4)
  Monthly <- rbind(Monthly,i=c(M1,M2,M3,M4))
}

monthly.ts <- ts(mnths,frequency = 4)
monthly.hw <- HoltWinters(monthly.ts, alpha=0.05, beta=0.1, gamma = 0.7)
preds <- predict(monthly.hw, n.ahead=(1*4))

series <- as.vector(temps[,1])
for(i in 2:20) {
  series<-c(series,temps[,i])
}
temps.ts <-ts(series, frequency = 123)
temps.hw <- HoltWinters(temps.ts, alpha = 0.2, beta = 0.0005, gamma = 0.5)
preds.days <- predict(temps.hw, n.ahead=(1*123))
```

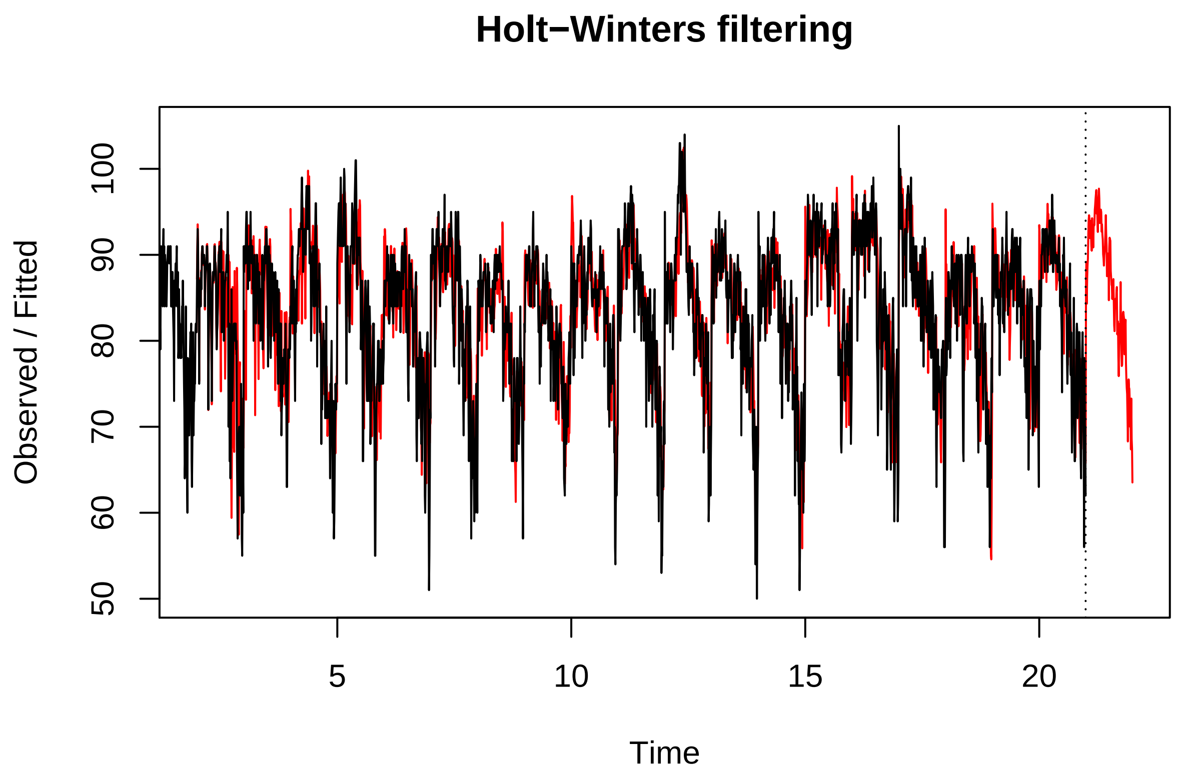
Given we now have two reasonable models we can look at the forecast to determine if the end of summer is coming later.

```r
plot(monthly.hw,preds)
```



If summer is ending later we would expect in our forecast to see a increasing temperatures in the 3rd and 4th months of our period. This is not observed.

```r
plot(temps.hw,preds.days)
```



This doesn't seem to me to be the best means of determining how long summer is, but lets compare our forecast year to previous results. If we define the end of summer to be the peak then summer peaks on the 35th day in our period. Previously we have seen the weather turn down after 20 days into the period, or as late as the 59th day in the period. Given this we cannot conclude summers are ending sooner.

Question 3

We can use regression to predict house prices in the area. Features we may use are: * average block size * median income * crime rate * distance to cbd * number of decent cafes

Question 4

I will build a model with linear regression and simply normalise all data points

```
crime = read.table('uscrime.txt', header = TRUE)
library('caret')
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
preprocessParams <- preProcess(crime[1:15], method=c("scale","center"))
transX <- predict(preprocessParams, crime[1:15])
transCrime <- crime
transCrime[1:15]<-predict(preprocessParams, crime[1:15])
# linear regression
model.lm <- lm(Crime~., data=transCrime)
```

```r
# predict provided results
features <- as.data.frame.numeric(c(14.0,0,10.0,12.0,15.5,
                                    0.640,94.0,150,1.1,0.120,3.6,3200,20.1,0.04,39.0),col.names = "sampl
features <- t(features)
cn <- colnames(crime)[1:15]
colnames(features)<-cn
features <- as.data.frame(predict(preprocessParams,features))
# prediction
results.lm = predict(model.lm, features)
results.lm
```

```
## c(14, 0, 10, 12, 15.5, 0.64, 94, 150, 1.1, 0.12, 3.6, 3200, 20.1, 0.04, 39)
##                                                                     155.4349
```

We can see a problem with this result, we would not expect to see a result of 155 on an example with a population to 150, this is a large population with an estimate below any we have seen in the actual dataset. We will need to try again, this time we will 1) do a log transform on the data - we identified this as possibly exlplaining the shape of the data in the previous example 2) remove highly correlated features and 3) perform cross-validation on our data.

```r
# Log transformation
logCrime<-crime
logCrime[3:15]<-log(logCrime[3:15])
logCrime[1]<-log(logCrime[1])

# Evaluate Algorithms: Feature Selection

# remove correlated attributes
# find attributes that are highly corrected
set.seed(42)
cutoff <- 0.70
correlations <- cor(logCrime[1:15])
highlyCorrelated <- findCorrelation(correlations, cutoff=cutoff)
# create a new dataset without highly corrected features
dataset_features <- logCrime[,-highlyCorrelated]

# Run algorithms using 10-fold cross validation
control <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"
# lm
set.seed(42)
fit.lm <- train(Crime~., data=dataset_features,
                method="lm", metric=metric, preProc=c("center", "scale"), trControl=control)

# Compare algorithms
summary(fit.lm)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -407.43 -128.99    6.72   99.48  387.44
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  905.085     34.105  26.538  < 2e-16 ***
## M             92.253     53.055   1.739   0.0906 .
## So            72.334     76.389   0.947   0.3500
## Po2          246.720     55.573   4.440 8.21e-05 ***
## LF            84.904     57.346   1.481   0.1474
## M.F           68.676     59.080   1.162   0.2527
## Pop           -8.784     59.296  -0.148   0.8831
## NW            96.017     64.554   1.487   0.1456
## U2            63.083     47.356   1.332   0.1912
## Prob        -124.041     66.450  -1.867   0.0701 .
## Time         -24.631     60.777  -0.405   0.6877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 233.8 on 36 degrees of freedom
## Multiple R-squared:  0.714,  Adjusted R-squared:  0.6345
## F-statistic: 8.986 on 10 and 36 DF,  p-value: 3.4e-07
```

```
fit.lm$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
## (Intercept)            M           So          Po2           LF
##     905.085       92.253       72.334      246.720       84.904
##         M.F          Pop           NW           U2         Prob
##      68.676       -8.784       96.017       63.083     -124.041
##        Time
##     -24.631
```

So we have built two models with the coefficents and goodness of fit as above. Now we can plug in the values in:

```
features <- as.data.frame.numeric(c(14.0,0,10.0,12.0,15.5,0.640,94.0,150,1.1,0.120,3.6,3200,20.1,0.04,3
features <- t(features)
cn <- colnames(crime)[1:15]
colnames(features)<-cn
features[1] <- log(features[1])
features[3:15] <- log(features[3:15])
features<-features[-highlyCorrelated]
features<-t(as.data.frame(features, row.names = colnames(dataset_features[1:10])))
result <- predict(fit.lm,features)
result
```

```
## features
## 1299.663
```

Here we end up with a estimate of 1300, from our descriptive analysis this is a lot close to what we would expect.

The next step in developing this model would be to look at removing values with a very low p-value. Note that we only have 3 values + the intercept passing a signifigance test at alpha = 0.1.