week 3 homework

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 a(the first smoothing parameter) to be closer to 0 or 1, and why?

Exponential smoothing is a forecasting method for short-term forecasts. For example, it can be used in retail sales, to predict product demand over time, using data such as volume of product sales over a certain time period. In such a case where we are predicting something based in human behavior, the value of α would be closer to 0. This is to accommodate the high randomness associated with human behavior, and assigns a lower weightage to the current observation x_t.

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. (Part of the point of this assignment is for you to think about how you might use exponential smoothing to answer this question. Feel free to combine it with other models if you'd like to. There's certainly more than one reasonable approach.)

Note: in R, you can use either `HoltWinters` (simpler to use) or the `smooth` package's `es` function (harder to use, but more general). If you use `es`, the Holt-Winters model uses `model="AAM"` in the function call (the first and second constants are used "A"dditively, and the third (seasonality) is used "M"ultiplicatively; the documentation doesn't make that clear).

We are using the `HoltWinters` function which is located in the `stats` package in R.

```
library(stats)
```

```
set.seed(42)
```

```
# load data
temp <- read.delim("../week 3 data-summer/data 7.2/temps.txt")
head(temp)
```

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

First we will attempt to use single exponential smoothing with the `HoltWinters` function (https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/HoltWinters). Before we can use the `HoltWinters` function, we are required to create a time-series object of type `ts` (https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/ts) which we can do easily using the `ts` function.

Per `HoltWinters` documentation: "The function tries to find the optimal values of α, β and γ by minimizing [RMSE] if they are `NULL` (the default)". We will leave `alpha` to the default value of `NULL` and set `beta` and `gamma` to `False` as these are coefficients that represent trend and seasonality, which are not present in single exponential smoothing.

```
# single exponential smoothing

# convert temp data into time-series object
temp_vector <- as.vector(unlist(temp[2:21]))
temp_ts <- ts(temp_vector, start=1996, frequency=nrow(temp)) # beginning in 1996, nrorw(temp) observati

# create simple exponential smoothing model
model1 <- HoltWinters(temp_ts, # data
                      beta=F,  # no trend
                      gamma=F) # no seasonality (data is summer only)
```

```
model1$alpha
```

```
## [1] 0.8388021
```

```
model1$beta
```

```
## [1] FALSE
```

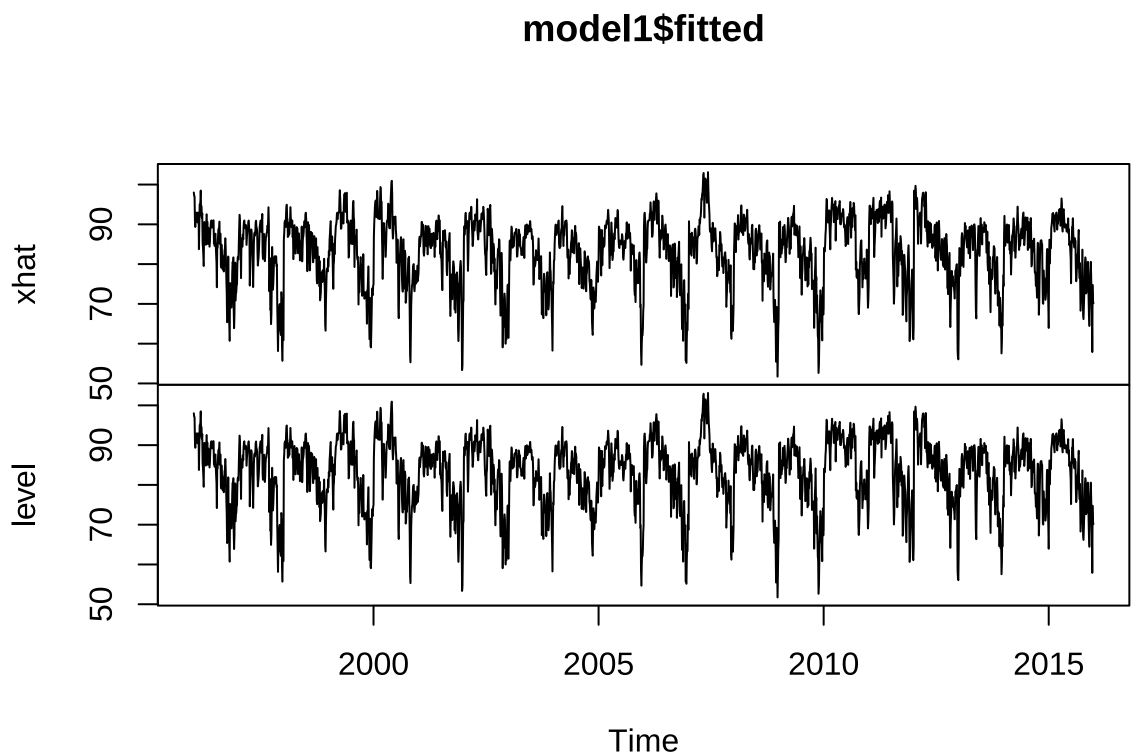
```
model1$gamma
```

```
## [1] FALSE
```

The model has determined a high value of `alpha`, 0.8388021. This represents a higher weightage to observed data and less to the previous baseline and indicates that the randomness in our data is relatively low.

Next, plot the time-series model:

```
plot(model1$fitted)
```



Plotting the model, we can see the data has quite noticeable variations and there might be trends or seasonality hidden within the data. Trends and seasonality are represented by the `beta` and `gamma` parameters. This time, we will leave these two parameters as `NULL`, together with the `alpha` parameter so that the function will try to find the optimal values for all 3 coefficients. We also set the `seasonal` parameter to `"multiplicative"`.

```
# double and triple exponential smoothing

# convert temp data into time-series object
temp_ts2 <- ts(temp_vector, start=1996, frequency=nrow(temp))

# create general exponential smoothing model
model2 <- HoltWinters(temp_ts2,
                      seasonal="multiplicative")
```

Let's see our coefficients:

```
model2$alpha
```

```
##    alpha
## 0.615003
```

```
model2$beta
```

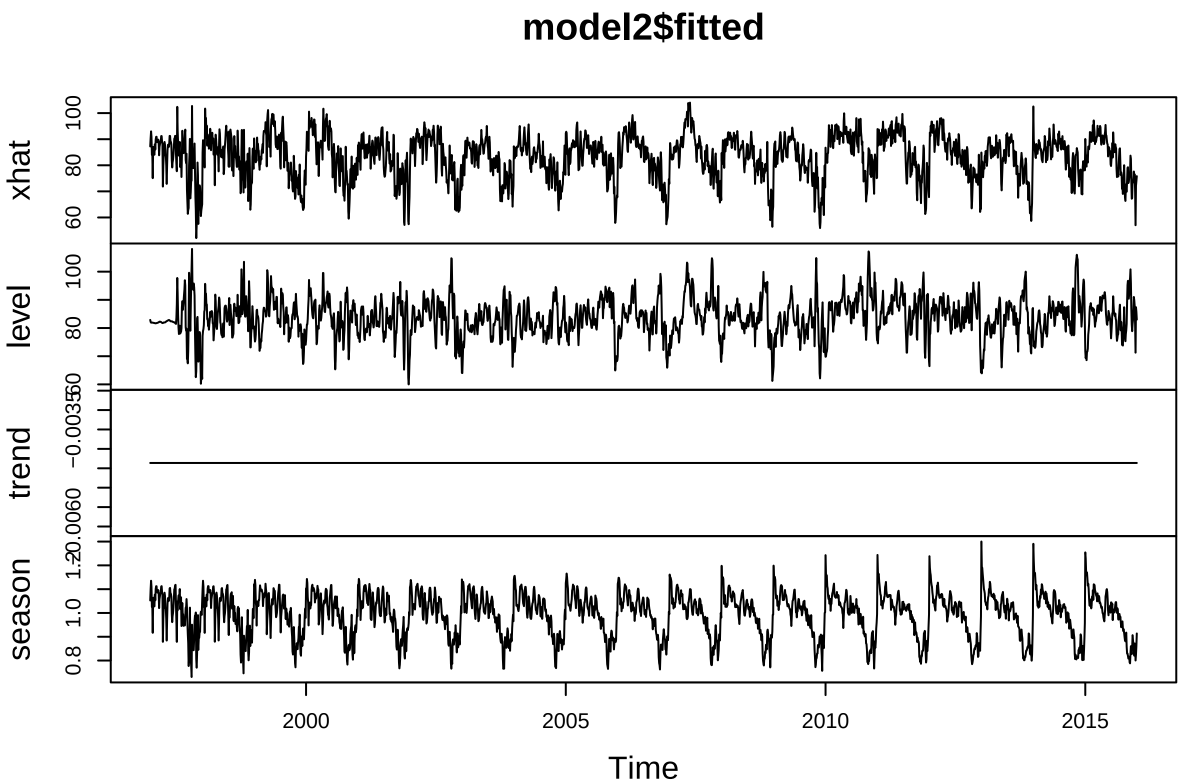
```
## beta
##    0
```

```r
model2$gamma
```

```
##     gamma
## 0.5495256
```

Our more general Holt-Winters model has indicated more randomness than the previous model ($\alpha = 0.615003$) and the presence of seasonality ($\gamma = 0.549256$), but no trending ($\beta = 0$).

```r
plot(model2$fitted)
```



The decomposed plot of our Holt-Winters model also clearly shows the flat trend in the data.

Overall, there is no discernible increasing or decreasing trend over the past 20 years. We can conclude that there is no statistical evidence to suggest an increase in summer temperatures, which corresponds with longer summers, from 1996 to 2015.

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.

Linear regressions can be used in businesses to understand the relationship between their business practices and revenue.

Some possible predictors could include advertisement spending, number of outlets their product is available at and online store traffic.

Question 8.2

Using crime data from http://www.statsci.org/data/general/uscrime.txt (file `uscrime.txt`, description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is `lm` or `glm`) to predict the observed crime rate in a city with the following [data:\\](data:){.uri}
`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.

Note that because there are only 47 data points and 15 predictors, you'll probably notice some overfitting. We'll see ways of dealing with this sort of problem later in the course.

We are using the `lm` function which is found in the `stats` package.

```
library(stats)
```

```
set.seed(42)
```

```
crime <- read.delim("../week 3 data-summer/data 8.2/uscrime.txt")
head(crime)
```

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

```r
# create dataframe from sample input from question
sample <- data.frame(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)
```

Here we create a simple linear regression model. Train it on the base `uscrime` dataset, without any scaling or normalization of data. Run the baseline model, and observe our model summary and model error.

```r
# train linear regression model
model3 <- lm(Crime~.,
             data=crime)

# get summary of baseline model
summary(model3)
```

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

```r
# get RMSE of baseline model
sigma(model3)
```

```
## [1] 209.0644
```

Predict a crime rate based on the sample data.

```r
# regression prediction based on sample
test <- predict(model3, sample)
test
```

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

```r
# reponses of training dataset
summary(crime$Crime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   342.0   658.5   831.0   905.1  1057.5  1993.0
```

Our model has predicted a crime rate of 155.43. Even simply eyeballing at the spread of responses in the training dataset, we can see that our model's prediction is extremely low. It's even lower than the minimum response in the training dataset which is 342.

A possible reason is that our model was not trained properly, and all the predictors were used regardless of significance to the model. This could have resulted in overfitting on predictors that were not statistically significant.

Let's try a new model using only significant predictors. We will select predictors with a p-value < 0.05.

```r
# train new model with only statistically significant predictors
model4 <- lm(Crime~M+Ed+Po1+U2+Ineq+Prob,
             data=crime)

# get summary of new model
summary(model4)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 ** 
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *  
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307 
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

```r
# get RMSE of baseline model
sigma(model4)
```

```
## [1] 200.6899
```

This model is a better model than our previous model, with a lower RMSE of 200 instead of 209. Additionally, the p-value, the R-squared value and F-statistic are still reasonable with our new coefficients. We can be more confident about this model than the previous one.

And we can say that our final linear equation is:

`Crime = -5040.50 + 105.02M + 196.47Ed + 115.02Pol + 89.37U2 + 67.65Ineq - 3801.84Prob`

Finally, generate a prediction:

```r
# new regression prediction from sample data
test2 <- predict(model4, sample)
test2
```

```
##        1 
## 1304.245
```

Thus we have predicted an observed crime rate based on the provided sample data to be 1304.245.