HW3

5/29/2020

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

Exponential smoothing can be applied to many everyday events. For example if the Hotel Franchise owner wants to look at monthly/yearly sales of his Atlanta Hotels. He sees that there is a spike in the sales during the last week of month becaue there was a conference in the town and lots of people few into the town from all over the country. Here they can use exponential smoothing model to smooth out the spike in the sale. I would keep the value of alpha to close to 0 as the historic records are more relevant Which will smooth out the spike caused by current event. Similarly, if they see spike in the sales due to few hotels in the neighborhood closing down. I will keep the alpha value close to 1, to heavily weight the most recent data.

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.

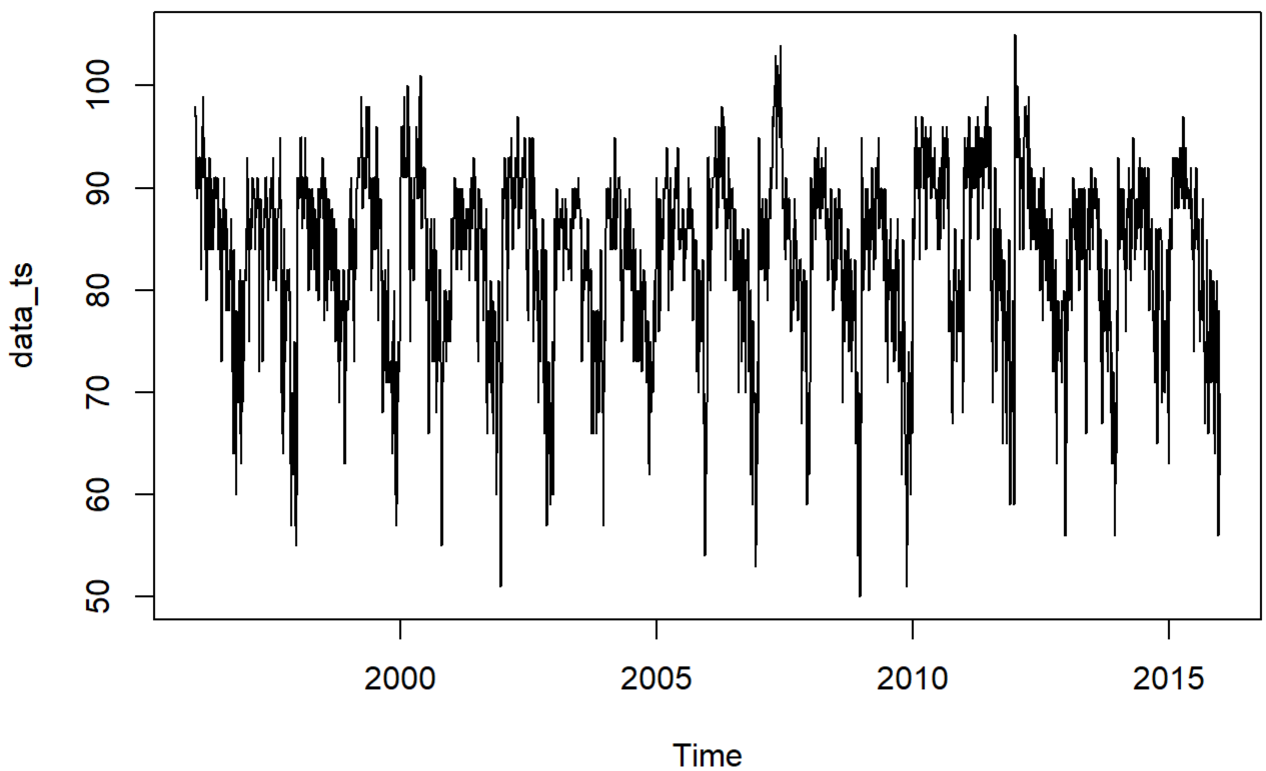
Let's load the data first…

```
data <- read.table("/Users/chintan/Downloads/6501/temps.txt",stringsAsFactors = FALSE, he
  ader = TRUE)
head(data)
```

```
##     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
```

Let's convert this data in to vector and then into a time series as need time series data for HoltWInters function.

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

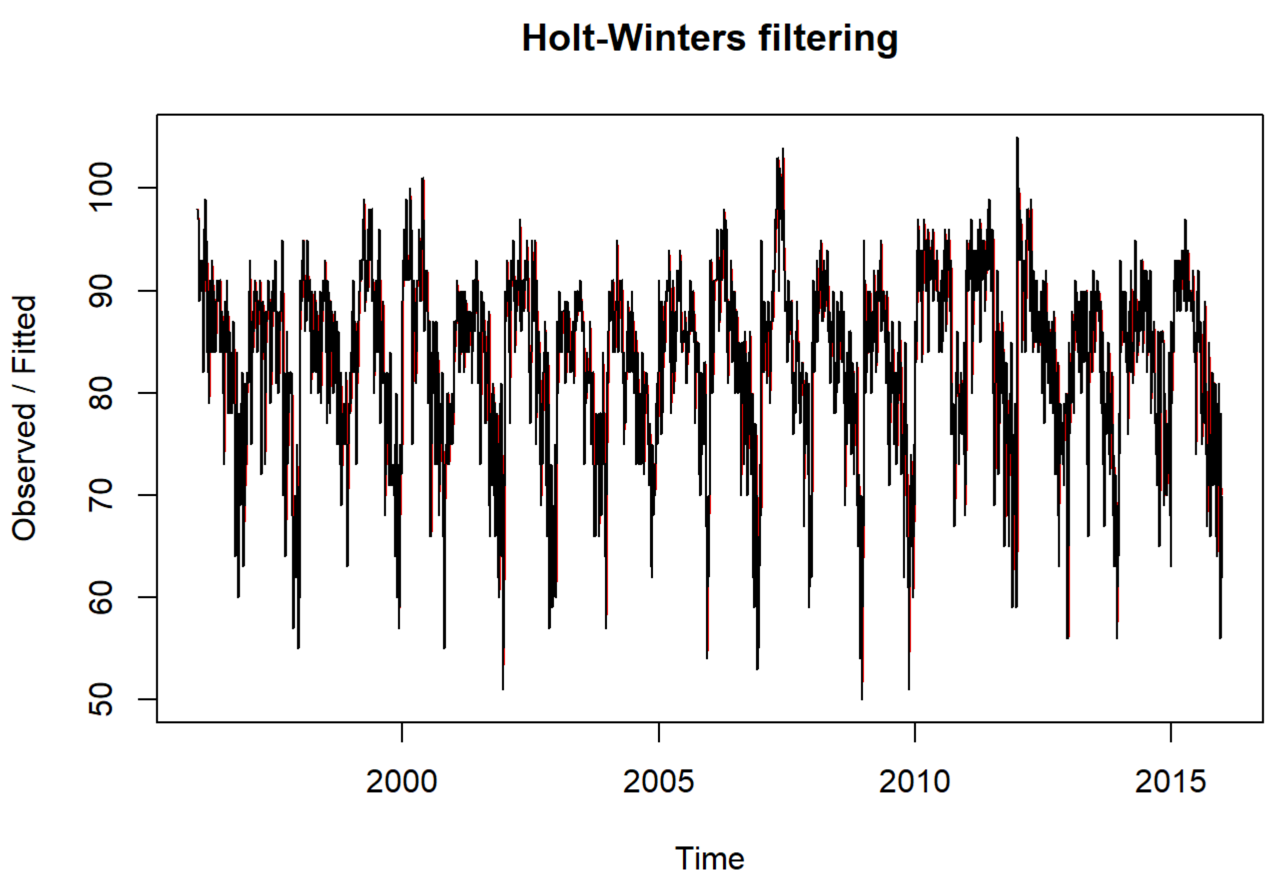



```
data_alpha <- HoltWinters(data_ts, beta= FALSE, gamma = FALSE)
data_alpha
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = data_ts, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.8388021
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##      [,1]
## a 63.30952
```

When we ran the HoltWinters function on our data set without trend and seasonal component, it returns the value of alpha as 0.8388 very close one. That means that somewhat more weight is placed on the recent data.

```
plot(data_alpha)
```



We can apply HoltWinters function for trend and observe the result...

```
data_beta <- HoltWinters(data_ts, gamma = FALSE)
data_beta
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = data_ts, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.8445729
##  beta : 0.003720884
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 63.2530022
## b -0.0729933
```

The above result suggests that there is no trend as value of b and value of beta is very close to zero.

Now we can also check for triple exonentail smoothing(multiplicative seansonality)

```
data_gamma <- HoltWinters(data_ts,alpha = NULL, beta = NULL, gamma =NULL,seasonal = "mult
  iplicative")
data_gamma
```

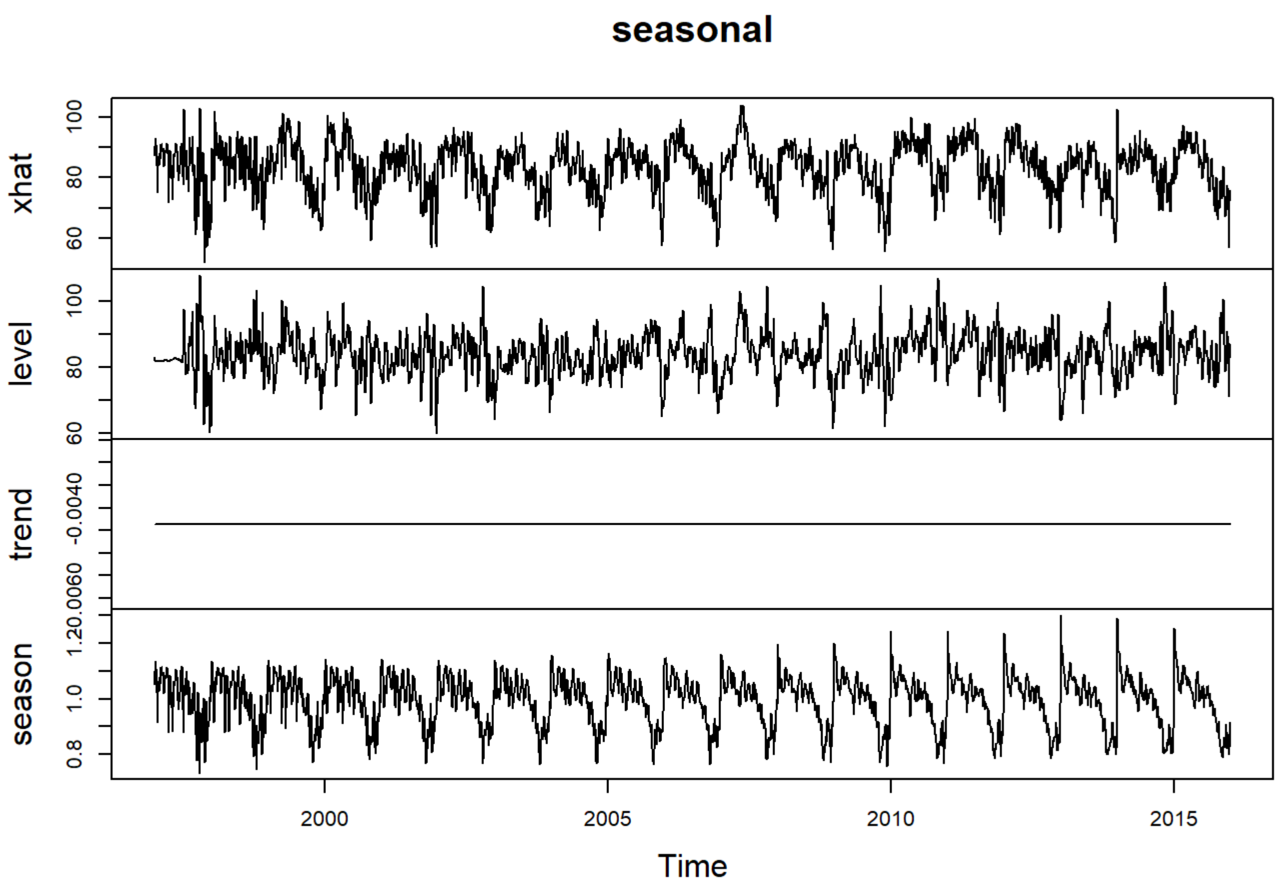
```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = data_ts, alpha = NULL, beta = NULL, gamma = NULL,     seasonal = "mult
iplicative")
##
## Smoothing parameters:
##  alpha: 0.615003
##  beta : 0
##  gamma: 0.5495256
##
## Coefficients:
##              [,1]
## a    73.679517064
## b    -0.004362918
## s1    1.239022317
## s2    1.234344062
## s3    1.159509551
## s4    1.175247483
## s5    1.171344196
## s6    1.151038408
## s7    1.139383104
## s8    1.130484528
## s9    1.110487514
## s10   1.076242879
## s11   1.041044609
## s12   1.058139281
## s13   1.032496529
## s14   1.036257448
## s15   1.019348815
## s16   1.026754142
## s17   1.071170378
## s18   1.054819556
## s19   1.084397734
## s20   1.064605879
## s21   1.109827336
## s22   1.112670130
## s23   1.103970506
## s24   1.102771209
## s25   1.091264692
## s26   1.084518342
## s27   1.077914660
## s28   1.077696145
## s29   1.053788854
## s30   1.079454300
## s31   1.053481186
## s32   1.054023885
## s33   1.078221405
```

```
## s34   1.070145761
## s35   1.054891375
## s36   1.044587771
## s37   1.023285461
## s38   1.025836722
## s39   1.031075732
## s40   1.031419152
## s41   1.021827552
## s42   0.998177248
## s43   0.996049257
## s44   0.981570825
## s45   0.976510542
## s46   0.967977608
## s47   0.985788411
## s48   1.004748195
## s49   1.050965934
## s50   1.072515008
## s51   1.086532279
## s52   1.098357400
## s53   1.097158461
## s54   1.054827180
## s55   1.022866587
## s56   0.987259326
## s57   1.016923524
## s58   1.016604903
## s59   1.004320951
## s60   1.019102781
## s61   0.983848662
## s62   1.055888360
## s63   1.056122844
## s64   1.043478958
## s65   1.039475693
## s66   0.991019224
## s67   1.001437488
## s68   1.002221759
## s69   1.003949213
## s70   0.999566344
## s71   1.018636837
## s72   1.026490773
## s73   1.042507768
## s74   1.022500795
## s75   1.002503740
## s76   1.004560984
## s77   1.025536556
## s78   1.015357769
## s79   0.992176558
## s80   0.979377825
## s81   0.998058079
## s82   1.002553395
```

```
## s83   0.955429116
## s84   0.970970220
## s85   0.975543504
## s86   0.931515830
## s87   0.926764603
## s88   0.958565273
## s89   0.963250387
## s90   0.951644060
## s91   0.937362688
## s92   0.954257999
## s93   0.892485444
## s94   0.879537700
## s95   0.879946892
## s96   0.890633648
## s97   0.917134959
## s98   0.925991769
## s99   0.884247686
## s100  0.846648167
## s101  0.833696369
## s102  0.800001437
## s103  0.807934782
## s104  0.819343668
## s105  0.828571029
## s106  0.795608740
## s107  0.796609993
## s108  0.815503509
## s109  0.830111282
## s110  0.829086181
## s111  0.818367239
## s112  0.863958784
## s113  0.912057203
## s114  0.898308248
## s115  0.878723779
## s116  0.848971946
## s117  0.813891909
## s118  0.846821392
## s119  0.819121827
## s120  0.851036184
## s121  0.820416491
## s122  0.851581233
## s123  0.874038407
```

The above result of gamma = 0.549 indicates that there is some seasonal factors in the data.

```
seasonal  <- data_gamma$fitted
plot(seasonal)
```



If we look closer to the season in above graph, we can see at the end there is littlebit change in the pattern.It is getting smoother. We can export this data and perform cusum to confrim this change.

```
seasonal_m <- matrix(seasonal[,4], nrow =123)
```

Below is the code to export the data into excel file…

```
#library(xlsx)
#file <- "HW4.xlsx"
#wb <- loadWorkbook("C:/Users/chintan/Downloads/6501/Hw4.xlsx")
#sheets <- getSheets(wb)
#sheet <- sheets[[1]]
#addDataFrame(seasonal_m, sheet, col.names = FALSE, row.names = FALSE, startRow = 2, star
  tColumn =2)
#saveWorkbook(wb,file)
```

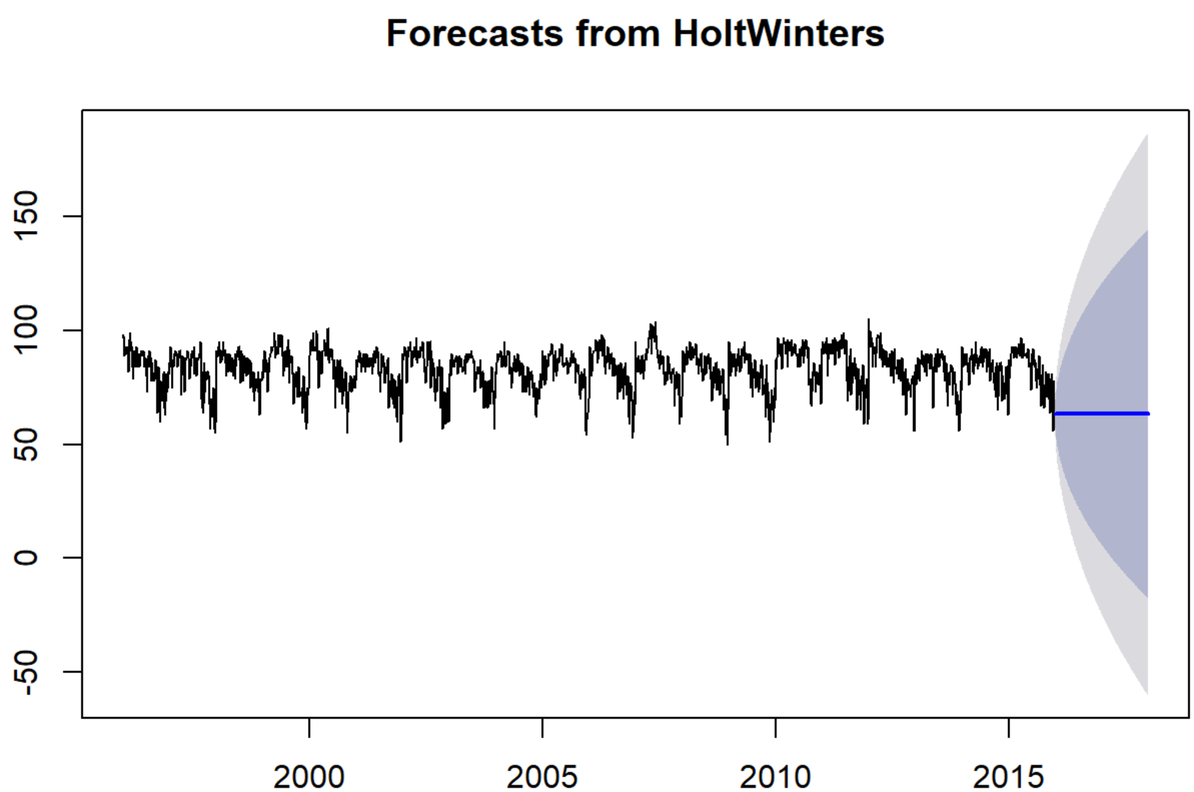
Please find the CUSUM analysis in the attached excel. The graphs in the sheet1 of excel shows the comparisons between early years (1997 to 2000) vs recent years(2012 to 2015). We can clearly see that the recent years graph's tail is showing smoothness. When I apply the CUSUM function it was very clear that in the recent years summers days are reduces. in 1997 summer days were 91 while moving towards more recent years the summer day got reduced to 67. See sheet2 and sheet3 of the attached excel file.

Below are the some forecast results…….

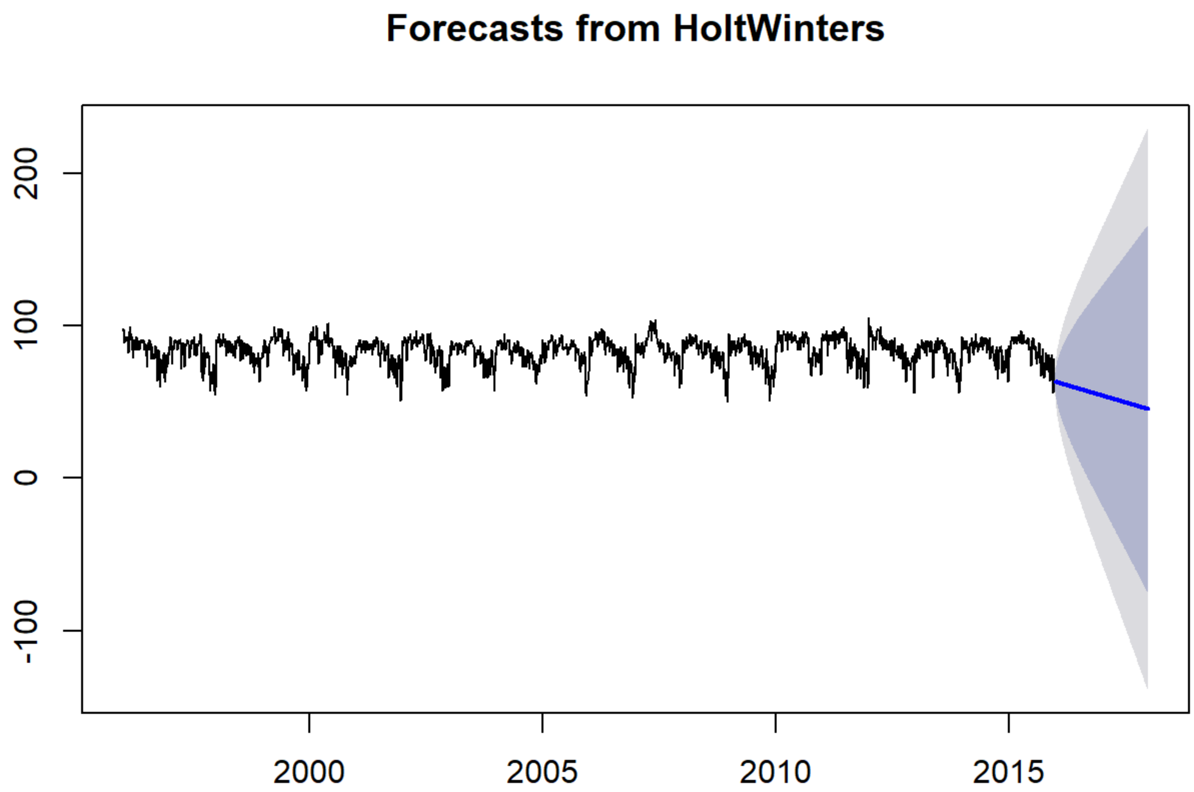
```
library("forecast")
```

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

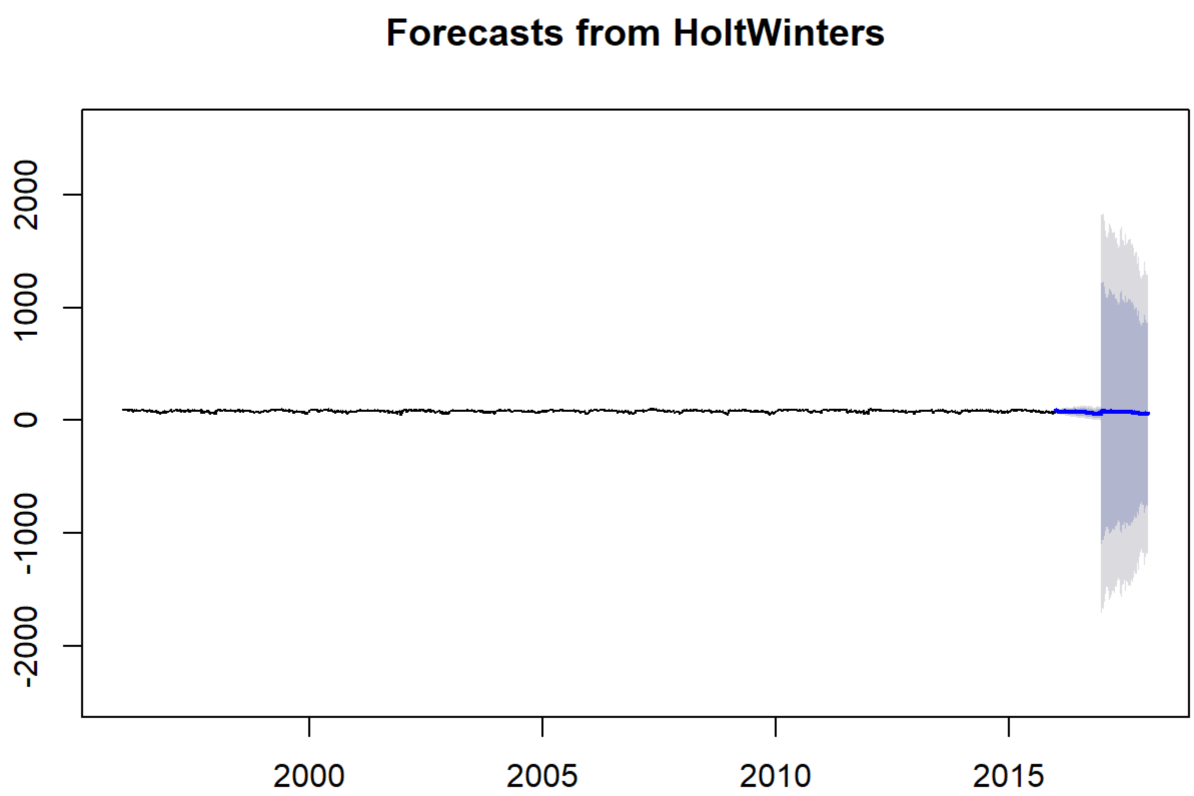
```
plot(forecast(data_alpha))
```




```
plot(forecast(data_beta))
```




```
plot(forecast(data_gamma))
```



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.

Example of Linear regression can be found in everywhere. For example if you have car and you have been recording how much mileage that car has given over the years. You can predict much will it cost(for gas) you to make a road trip from Atlanta to New York. Another example is if you own the hotel franchise and spending certain amount of money on marketing every year and you have sales data. You can predict if you spend certain amount money on marketing how much of your sales will be for that year. You can add more predictors like type of marketing. Online marketing, coupons in local newspaper, coupons in travel booklets, purpose of the visit (vacation vs business) etc. All predictors can be used to predict the sales of the given year or month or quarter .

Question 8.2

Using crime data from http://www.statsci.org/data/general/uscrime.txt (http://www.statsci.org/data/general/uscrime.txt) (file uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html (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:

Let’s load the data first…

```
cr_data <- read.table("/Users/chintan/Downloads/6501/crimedata.txt",stringsAsFactors = FA
  LSE, header = TRUE)
head(cr_data)
```

```
##      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
```

First we need to build the regression model using lm function. Then we will use predict function with the given parameters to predict the crime rate.

We are using Crime column as the target/response variable and using rest of the columns as predictors.

```
lm_model <- lm(Crime~.,data = cr_data)
summary(lm_model)
```

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

Let's analyze the output..... I have searched online to understand the output and find out few interesting things about it. I will use this interpretation to analyze the prediction.

Let's create the data points with given values in the question.

```
given_dp <- 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)
```

```
cr_predict<- predict(lm_model, given_dp)
cr_predict
```

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

Our linear regression model for the given data points predict the crime rate of 155.43. Let's just cross check that. When we look at our data set the Max crime rate is 1993 and min crime rate is 342 but our model predicts the crime rate of 155 which is less than our min crime rate. That seems little bit odd. There must be something wrong.

One reason could be the outliers as seen in previous HW, that our data consists some outliers. Let's try removing them and see the predictions…

```
library(outliers)
```

```
cr_data1 <- cr_data[-which.max(cr_data$Crime),]
cr_data2 <- cr_data1[-which.max(cr_data1$Crime),]
cr_data3 <- cr_data2[-which.max(cr_data2$Crime),]
cr_data4 <- cr_data3[-which.max(cr_data3$Crime),]
cr_data5 <- cr_data4[-which.max(cr_data4$Crime),]

cr_test <- cr_data5[,16]
cr_test
```

```
##  [1]  791  578 1234  682  963  856  705  849  511  664  798  946  539  929  750
## [16] 1225  742  439 1216  968  523  342 1216 1043  696  373  754 1072  923  653
## [31] 1272  831  566  826 1151  880  542  823 1030  455  508  849
```

```
gb_test <- grubbs.test(cr_test,type =10)
gb_test
```

```
## 
##  Grubbs test for one outlier
## 
## data:  cr_test
## G = 1.87133, U = 0.91251, p-value = 1
## alternative hypothesis: highest value 1272 is an outlier
```

```
new_lmmodel <- lm(Crime~.,data = cr_data5)
new_predict <- predict(new_lmmodel, given_dp)
new_predict
```

```
##       1
## 87.2166
```

Well, this didn't solve the problem.

Let's try something else..... Below is the summary of our data set. We can compare the value of given data points against this summary and check that all the values of given data point is within Max and Min value of each column. After comparing I confirm that is nothing wrong with the given data point.

```
summary(cr_data)
```

```
##        M               So               Ed              Po1
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80
##  Mean   :13.86   Mean   :0.3404   Mean   :10.56   Mean   : 8.50
##  3rd Qu.:14.60   3rd Qu.:1.0000   3rd Qu.:11.45   3rd Qu.:10.45
##  Max.   :17.70   Max.   :1.0000   Max.   :12.20   Max.   :16.60
##       Po2              LF              M.F              Pop
##  Min.   : 4.100   Min.   :0.4800   Min.   : 93.40   Min.   :  3.00
##  1st Qu.: 5.850   1st Qu.:0.5305   1st Qu.: 96.45   1st Qu.: 10.00
##  Median : 7.300   Median :0.5600   Median : 97.70   Median : 25.00
##  Mean   : 8.023   Mean   :0.5612   Mean   : 98.30   Mean   : 36.62
##  3rd Qu.: 9.700   3rd Qu.:0.5930   3rd Qu.: 99.20   3rd Qu.: 41.50
##  Max.   :15.700   Max.   :0.6410   Max.   :107.10   Max.   :168.00
##        NW              U1                U2             Wealth
##  Min.   : 0.20   Min.   :0.07000   Min.   :2.000   Min.   :2880
##  1st Qu.: 2.40   1st Qu.:0.08050   1st Qu.:2.750   1st Qu.:4595
##  Median : 7.60   Median :0.09200   Median :3.400   Median :5370
##  Mean   :10.11   Mean   :0.09547   Mean   :3.398   Mean   :5254
##  3rd Qu.:13.25   3rd Qu.:0.10400   3rd Qu.:3.850   3rd Qu.:5915
##  Max.   :42.30   Max.   :0.14200   Max.   :5.800   Max.   :6890
##       Ineq            Prob              Time            Crime
##  Min.   :12.60   Min.   :0.00690   Min.   :12.20   Min.   : 342.0
##  1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 658.5
##  Median :17.60   Median :0.04210   Median :25.80   Median : 831.0
##  Mean   :19.40   Mean   :0.04709   Mean   :26.60   Mean   : 905.1
##  3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:1057.5
##  Max.   :27.60   Max.   :0.11980   Max.   :44.00   Max.   :1993.0
```

One reason could be that the numbers of predictors we have used, are causing this issue. Now, how can we find out which predictors are important and which are not. Here, we can use the summary of our "lm" model's output.

In our output under the coefficients there are few columns but we are interested in only two. One is "t value" and the other is "Pr>|t|" As explained in lecture video 8.6, we can remove the predictors which P value is greater than 0.09… However, the lecture also suggets that we need to take judgment call on removing predictors sometime they have very high P value but they could be imp. predictors for analyzing the data.

The lm_test1 model looks at the predictors with P<0.09…

```
lm_test1 <- lm(Crime ~ M+Ed + Po1 +U2+ Wealth+ Ineq+ Prob,data = cr_data)
summary(lm_test1)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Wealth + Ineq + Prob,
##     data = cr_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -424.63  -85.83  -26.18  100.57  504.95
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.734e+03  1.058e+03  -5.421 3.29e-06 ***
## M            1.122e+02  3.360e+01   3.338 0.001863 **
## Ed           1.821e+02  4.599e+01   3.960 0.000309 ***
## Po1          1.030e+02  1.681e+01   6.130 3.41e-07 ***
## U2           8.352e+01  4.093e+01   2.041 0.048078 *
## Wealth       1.134e-01  9.244e-02   1.227 0.227063
## Ineq         8.058e+01  1.740e+01   4.631 3.98e-05 ***
## Prob        -3.357e+03  1.561e+03  -2.151 0.037762 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 199.4 on 39 degrees of freedom
## Multiple R-squared:  0.7746, Adjusted R-squared:  0.7341
## F-statistic: 19.14 on 7 and 39 DF,  p-value: 8.28e-11
```

```
cr_predict_test1<- predict(lm_test1, given_dp)
cr_predict_test1
```

```
##        1
## 1043.069
```

The lecture also points that sometimes we need to look at other factores as well.So I have created one model using “t-value” What I have found out is that the coefficient t-value is a measure of how many standard deviations our coefficient estimate is far away from 0. We want it to be far away from zero as this could declare a relationship between predictors and response variable exist. In our case the t-values vary from -3.6 to + 3.0. For let’s consider all the predictors with positive t-value.

```
lm_test2 <- lm(Crime ~ M+Ed + Po1 +M.F+NW+U2+ Wealth+ Ineq,data = cr_data)
summary(lm_test2)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + NW + U2 + Wealth +
##     Ineq, data = cr_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -375.67 -100.20  -16.81  100.51  549.25
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.620e+03  1.325e+03  -4.997 1.34e-05 ***
## M            1.102e+02  3.990e+01   2.763 0.008790 **
## Ed           1.611e+02  5.820e+01   2.768 0.008672 **
## Po1          1.087e+02  2.007e+01   5.418 3.58e-06 ***
## M.F          7.536e+00  1.345e+01   0.560 0.578631
## NW          -1.200e+00  5.312e+00  -0.226 0.822498
## U2           7.616e+01  4.547e+01   1.675 0.102175
## Wealth       1.524e-01  9.716e-02   1.569 0.124964
## Ineq         8.160e+01  1.953e+01   4.177 0.000166 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 212.4 on 38 degrees of freedom
## Multiple R-squared:  0.7508, Adjusted R-squared:  0.6983
## F-statistic: 14.31 on 8 and 38 DF,  p-value: 2.353e-09
```

```
cr_predict_test2<- predict(lm_test2, given_dp)
cr_predict_test2
```

```
##        1
## 948.3808
```

As we can see that once we remove the less important predictors from the model, the prediction of crime rate looks little better.

Let's try the "glm" function for the same predictors...

```
glm_model  <- glm(Crime ~ M+Ed + Po1 +M.F+NW+U2+ Wealth+ Ineq,data = cr_data, family="gau
  ssian")
summary(glm_model)
```

```
##
## Call:
## glm(formula = Crime ~ M + Ed + Po1 + M.F + NW + U2 + Wealth +
##     Ineq, family = "gaussian", data = cr_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -375.67  -100.20   -16.81   100.51   549.25
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.620e+03  1.325e+03  -4.997 1.34e-05 ***
## M            1.102e+02  3.990e+01   2.763 0.008790 **
## Ed           1.611e+02  5.820e+01   2.768 0.008672 **
## Po1          1.087e+02  2.007e+01   5.418 3.58e-06 ***
## M.F          7.536e+00  1.345e+01   0.560 0.578631
## NW          -1.200e+00  5.312e+00  -0.226 0.822498
## U2           7.616e+01  4.547e+01   1.675 0.102175
## Wealth       1.524e-01  9.716e-02   1.569 0.124964
## Ineq         8.160e+01  1.953e+01   4.177 0.000166 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 45132.15)
##
##     Null deviance: 6880928  on 46  degrees of freedom
## Residual deviance: 1715022  on 38  degrees of freedom
## AIC: 647.11
##
## Number of Fisher Scoring iterations: 2
```

```
cr_predict_glm<- predict(glm_model, given_dp)
cr_predict_glm
```

```
##        1
## 948.3808
```

Well, glm function also predicts the same crime rate. Overall its looks like we need to understand the importance of predicators that we choose to build the model.