Homework 3

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

My answer:

Problem: daily stock price for the company XXX

Data needed: the stock price over the last 2 or 5 years,the longer the better

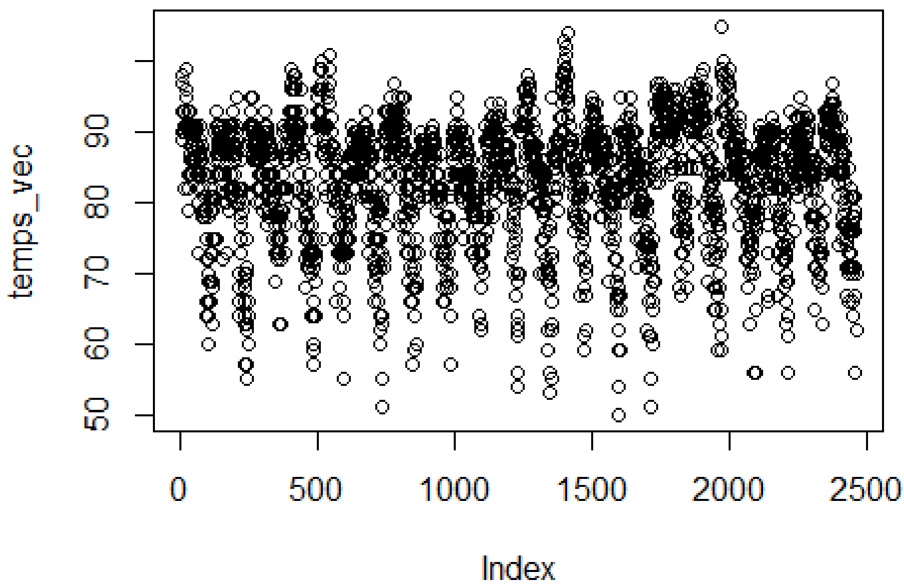
I would expect the alpha to be closer to 1, which means there is not much randomness in the system. The stock price of the company should reflect the actual value of the company, when there is not many market uncertainties (trade war etc.). The price stays within an acceptable range of fluctuation. If we observed a fluctuation today, it probably means today's baseline is close to the observed 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.

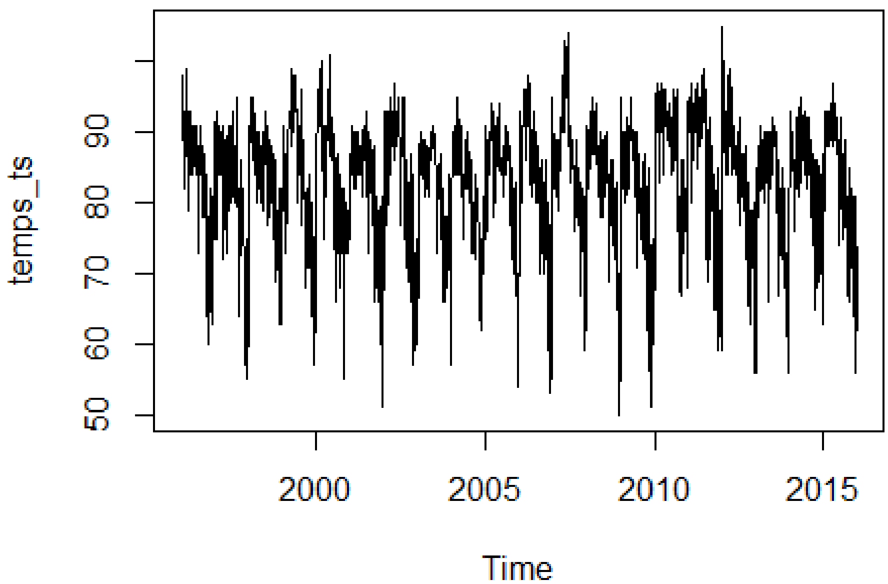
Read in the data and plot it to see what the distribution looks like

```
temps<-read.table("temps.txt",header=TRUE)
temps_vec<-as.vector(unlist(temps[,2:21]))

plot(temps_vec)
```

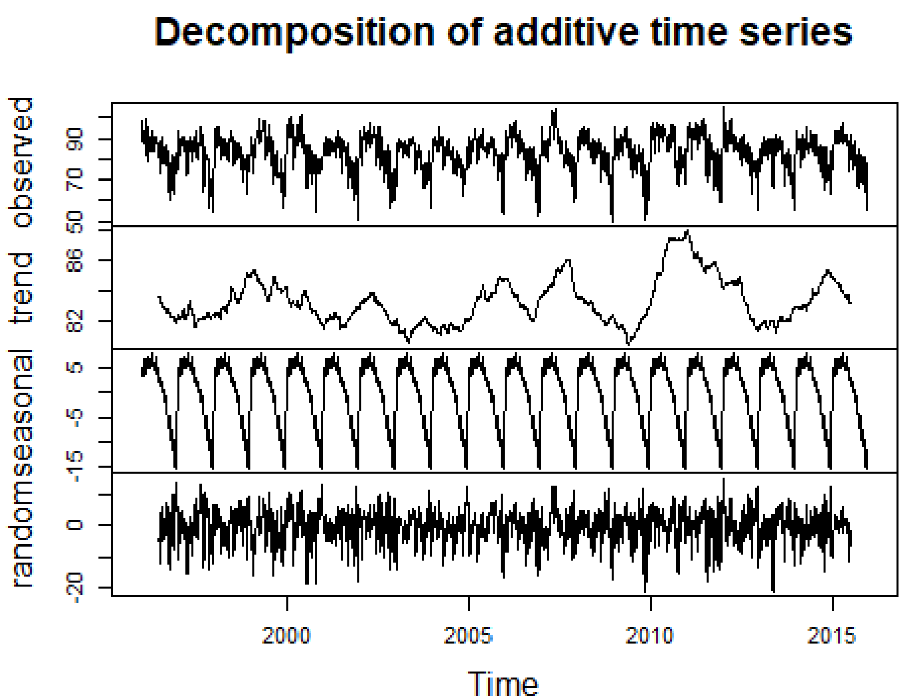


Convert it to time series data and check the distribution

```
temps_ts<-ts(temps_vec,start=1996,frequency = 123)
temps_ts

plot(temps_ts)
```




```
plot(decompose(temps_ts))
```



Apply the HoltWinters function to get the predicted value

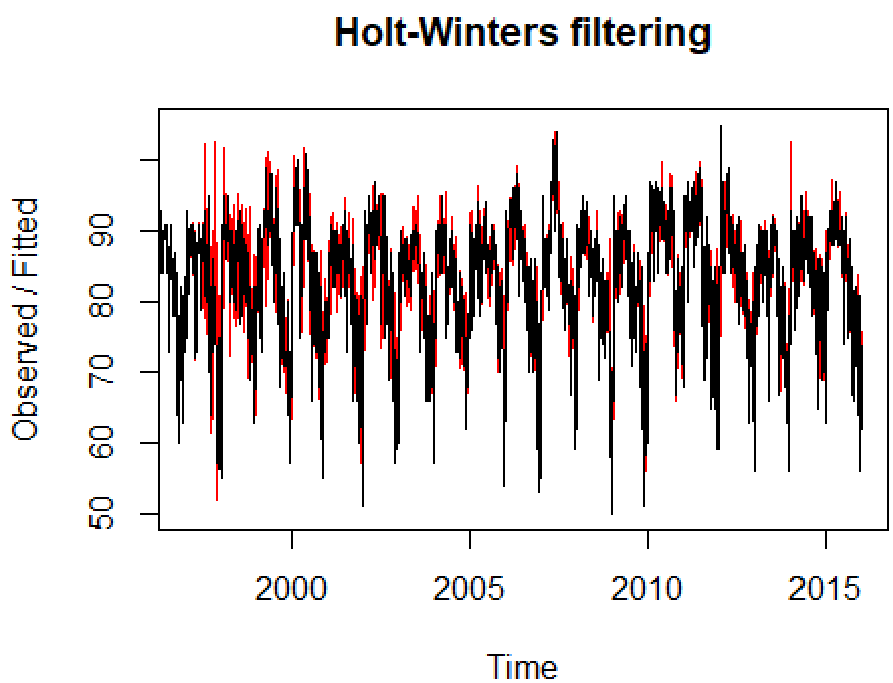
```
temps_hw<-HoltWinters(temps_ts,alpha = NULL, beta= NULL, gamma =
NULL,seasonal = "multiplicative")
temps_hw
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal
component.
##
## Call:
## HoltWinters(x = temps_ts, alpha = NULL, beta = NULL, gamma = NULL,
seasonal = "multiplicative")
##
## 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
```

```r
summary(temps_hw)

##              Length Class  Mode
## fitted       9348   mts    numeric
## x            2460   ts     numeric
## alpha           1   -none- numeric
## beta            1   -none- numeric
## gamma           1   -none- numeric
## coefficients  125   -none- numeric
## seasonal        1   -none- character
## SSE             1   -none- numeric
## call            6   -none- call

plot(temps_hw)
```



We can see from the plot that the predicted values (red line) agree pretty well with the observed values (black). The prediction begins at about year 1997, that's because 1996's data were used for prediction. The model works better with more data available. So it looks like the red line and black line align better at the later year.

Now let's take a look at the seasonal factors:

```r
head(temps_hw$fitted)

##          xhat    level        trend   season
## [1,] 87.23653 82.87739 -0.004362918 1.052653
## [2,] 90.42182 82.15059 -0.004362918 1.100742
## [3,] 92.99734 81.91055 -0.004362918 1.135413
```

```
## [4,] 90.94030 81.90763 -0.004362918 1.110338
## [5,] 83.99917 81.93634 -0.004362918 1.025231
## [6,] 84.04496 81.93247 -0.004362918 1.025838
```

```
temps_hw_sf<-matrix(temps_hw$fitted[,4],nrow=123)
head(temps_hw_sf)
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,] 1.052653 1.049468 1.120607 1.103336 1.118390 1.108172 1.140906
## [2,] 1.100742 1.099653 1.108025 1.098323 1.110184 1.116213 1.126827
## [3,] 1.135413 1.135420 1.139096 1.142831 1.143201 1.138495 1.129678
## [4,] 1.110338 1.110492 1.117079 1.125774 1.134539 1.126117 1.130758
## [5,] 1.025231 1.025233 1.044684 1.067291 1.084725 1.097239 1.115055
## [6,] 1.025838 1.025722 1.028169 1.042340 1.053954 1.067494 1.080203
##          [,8]     [,9]    [,10]    [,11]    [,12]    [,13]    [,14]
## [1,] 1.140574 1.125438 1.122063 1.161415 1.198102 1.198910 1.243012
## [2,] 1.154074 1.142187 1.131889 1.144549 1.134661 1.153433 1.165431
## [3,] 1.156092 1.165657 1.147982 1.149459 1.135756 1.153310 1.155197
## [4,] 1.137722 1.150639 1.146992 1.142497 1.150162 1.151169 1.157751
## [5,] 1.103877 1.120818 1.133733 1.132167 1.142714 1.139244 1.112909
## [6,] 1.094312 1.102680 1.092178 1.075766 1.088547 1.082185 1.103092
##         [,15]    [,16]    [,17]    [,18]    [,19]
## [1,] 1.243781 1.238435 1.300204 1.290647 1.254521
## [2,] 1.172935 1.190735 1.191956 1.219190 1.228826
## [3,] 1.157286 1.169773 1.189915 1.172309 1.169045
## [4,] 1.163844 1.159343 1.166605 1.167993 1.158956
## [5,] 1.132435 1.132045 1.145230 1.168161 1.170449
## [6,] 1.115071 1.118575 1.121598 1.134962 1.145475
```

```
temps_hw_smoothed<-matrix(temps_hw$fitted[,1],nrow=123)
```

I am using the predicted value (Xhat) and the observed value (data from original temps.txt) to see if summer has gotten later over the 20 years. When the difference between the predicted value and the observed value reached max, which means predicted value is much higher than the observed value, among the whole period, I assume that day is the summer end date (the model thought the temperature was still high, while the actual temperature already went down). Then I compare the date from year 1997 to year 2015 to see how the date varies.

```
fit<-temps_hw$fitted

y1997<- data.frame(fit = fit[1:123,1], obs = temps[,2])
y1997$dif <-y1997$fit - y1997$obs
head(y1997)
```

```
##        fit obs         dif
## 1 87.23653  98 -10.7634671
## 2 90.42182  97  -6.5781811
## 3 92.99734  97  -4.0026586
## 4 90.94030  90   0.9402979
```

```
## 5 83.99917  89  -5.0008320
## 6 84.04496  93  -8.9550380
```

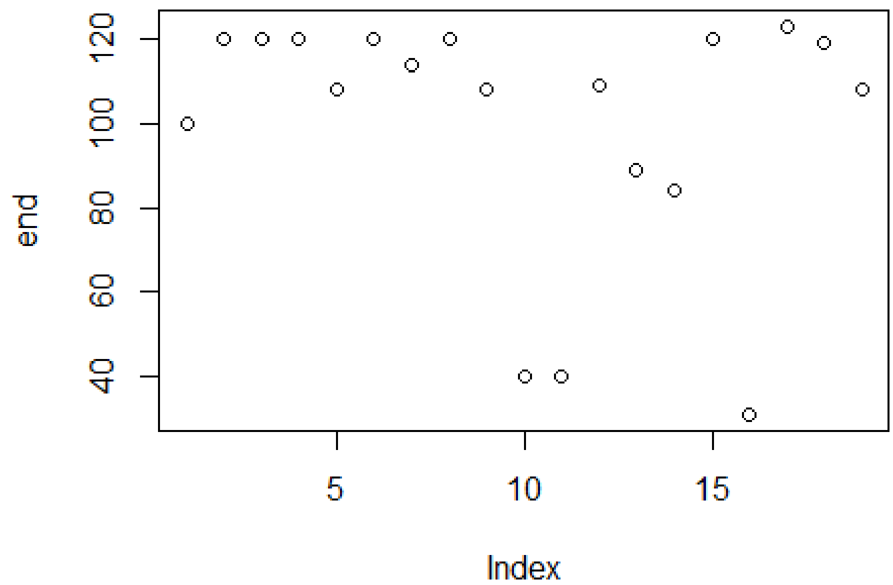
```r
a<-which.max(y1997$dif)
y1997[which.max(y1997$dif),]
```

```
##          fit obs      dif
## 100 102.7329  78 24.73294
```

The 100th day is the day where predicted tempererature is much higher than the actual temperature.

```r
y1998<- data.frame(fit = fit[124:246,1], obs = temps[,3])
y1998$dif <-y1998$fit - y1998$obs
b<-which.max(y1998$dif)

y1999<- data.frame(fit = fit[247:369,1], obs = temps[,3])
y1999$dif <-y1999$fit - y1999$obs
c<-which.max(y1999$dif)

y2000<- data.frame(fit = fit[370:492,1], obs = temps[,3])
y2000$dif <-y2000$fit - y2000$obs
d<-which.max(y2000$dif)

y2001<- data.frame(fit = fit[493:615,1], obs = temps[,3])
y2001$dif <-y2001$fit - y2001$obs
e<-which.max(y2001$dif)

y2002<- data.frame(fit = fit[616:738,1], obs = temps[,3])
y2002$dif <-y2002$fit - y2002$obs
f<-which.max(y2002$dif)

y2003<- data.frame(fit = fit[739:861,1], obs = temps[,3])
y2003$dif <-y2003$fit - y2003$obs
g<-which.max(y2003$dif)

y2004<- data.frame(fit = fit[862:984,1], obs = temps[,3])
y2004$dif <-y2004$fit - y2004$obs
h<-which.max(y2004$dif)

y2005<- data.frame(fit = fit[985:1107,1], obs = temps[,3])
y2005$dif <-y2005$fit - y2005$obs
i<-which.max(y2005$dif)

y2006<- data.frame(fit = fit[1108:1230,1], obs = temps[,3])
y2006$dif <-y2006$fit - y2006$obs
j<-which.max(y2006$dif)

y2007<- data.frame(fit = fit[1231:1353,1], obs = temps[,3])
y2007$dif <-y2007$fit - y2007$obs
```

```r
k<-which.max(y2007$dif)

y2008<- data.frame(fit = fit[1354:1476,1], obs = temps[,3])
y2008$dif <-y2008$fit - y2008$obs
l<-which.max(y2008$dif)

y2009<- data.frame(fit = fit[1477:1599,1], obs = temps[,3])
y2009$dif <-y2009$fit - y2009$obs
m<-which.max(y2009$dif)

y2010<- data.frame(fit = fit[1600:1722,1], obs = temps[,3])
y2010$dif <-y2010$fit - y2010$obs
n<-which.max(y2010$dif)

y2011<- data.frame(fit = fit[1723:1845,1], obs = temps[,3])
y2011$dif <-y2011$fit - y2011$obs
o<-which.max(y2011$dif)

y2012<- data.frame(fit = fit[1846:1968,1], obs = temps[,3])
y2012$dif <-y2012$fit - y2012$obs
p<-which.max(y2012$dif)

y2013<- data.frame(fit = fit[1969:2091,1], obs = temps[,3])
y2013$dif <-y2013$fit - y2013$obs
q<-which.max(y2013$dif)

y2014<- data.frame(fit = fit[2092:2214,1], obs = temps[,3])
y2014$dif <-y2014$fit - y2014$obs
r<-which.max(y2014$dif)

y2015<- data.frame(fit = fit[2215:2337,1], obs = temps[,3])
y2015$dif <-y2015$fit - y2015$obs
s<-which.max(y2015$dif)

end<-c(a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s)
plot(end)
```



The plot shows the end days stays within a range of 100 -120 days from July 1st. So I would say the summer does not get later over the 20 years.

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.

My answer:

The height of an adult

predictors:

mother's height

father's height

gender

calories intake each week

height at age ten

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: 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.

```
crime<-
read.table("http://www.statsci.org/data/general/uscrime.txt",header=TRUE)
head(crime)

##      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

model1<-lm(Crime~.,crime)
summary(model1)

##
## 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
```

I am using 0.05 as the threshold. Based on the p-value, I will keep M,Ed,Ineq,Prob,to fit a new model.

```
model2<-lm(Crime~M+Ed+Ineq+Prob,crime)
summary(model2)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Ineq + Prob, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -532.97 -254.03  -55.72  137.80  960.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1339.35    1247.01  -1.074  0.28893
## M              35.97      53.39   0.674  0.50417
## Ed            148.61      71.92   2.066  0.04499 *
## Ineq           26.87      22.77   1.180  0.24458
## Prob        -7331.92    2560.27  -2.864  0.00651 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 347.5 on 42 degrees of freedom
## Multiple R-squared:  0.2629, Adjusted R-squared:  0.1927
## F-statistic: 3.745 on 4 and 42 DF,  p-value: 0.01077
```

Now only the Ed and Prob still remains significant. The adjusted R-squred dropped from 0.7078 to 0.1927. I will fit a new model to see what happens.

```
model3<-lm(Crime~Ed+Prob,crime)
summary(model3)
```

```
##
## Call:
## lm(formula = Crime ~ Ed + Prob, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -650.98 -279.57  -14.06  198.00  957.48
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   517.30     588.48   0.879   0.3842
## Ed             63.67      50.26   1.267   0.2119
## Prob        -6049.00    2472.93  -2.446   0.0185 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 351.2 on 44 degrees of freedom
## Multiple R-squared:  0.2115, Adjusted R-squared:  0.1756
## F-statistic: 5.899 on 2 and 44 DF,  p-value: 0.005373
```

Now only Prob remains significant. The adjusted R-squred dropped from 0.1927. to 0.1756. So the first model with all the variables included truns out to be the best model so far. Next, I will try to use every combination to find the "best" model.

Create a NULL vector called model so we have something to add our layers to

```
model<-NULL
```

Create a vector of the dataframe column names used to build the formula.

```
vars <-names(crime)
```

Remove the response variable (it's in the 16th column)

```
vars <-vars[-16]
```

The combn function will run every different combination of variables and then run the lm.

```
for(i in 1:length(vars)){
                xx = combn(vars,i)
                if(is.null(dim(xx))){
                        fla = paste("Crime ~", paste(xx, collapse="+"))

model[[length(model)+1]]=lm(as.formula(fla),data=crime)
```

```
                } else {
                        for(j in 1:dim(xx)[2]){
                                fla = paste("Crime ~",
paste(xx[1:dim(xx)[1],j], collapse="+"))

model[[length(model)+1]]=lm(as.formula(fla),data=crime)
                        }
                }
        }
```

See how many models were build using the loop above.

```
length(model)

## [1] 32767
```

Create a vector to extract AIC and BIC values from the model variable.

```
AICs <- NULL
BICs <- NULL
        for(i in 1:length(model)){
                AICs[i] = AIC(model[[i]])
                BICs[i] = BIC(model[[i]])
        }
```

See which models were chosen as best by AIC and BIC.

```
which(AICs==min(AICs))

## [1] 18494

which(BICs==min(BICs))

## [1] 5817
```

See which variables are in those models, and the corresponding adjusted R-squared.

```
summary(model[[18494]])

##
## Call:
## lm(formula = as.formula(fla), data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -444.70 -111.07    3.03  122.15  483.30
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6426.10    1194.61  -5.379 4.04e-06 ***
## M              93.32      33.50   2.786  0.00828 **
## Ed            180.12      52.75   3.414  0.00153 **
## Po1           102.65      15.52   6.613 8.26e-08 ***
```

```
## M.F            22.34      13.60   1.642  0.10874
## U1          -6086.63    3339.27  -1.823  0.07622 .
## U2            187.35      72.48   2.585  0.01371 *
## Ineq           61.33      13.96   4.394 8.63e-05 ***
## Prob        -3796.03    1490.65  -2.547  0.01505 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.5 on 38 degrees of freedom
## Multiple R-squared:  0.7888, Adjusted R-squared:  0.7444
## F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

```
summary(model[[24966]])
```

```
##
## Call:
## lm(formula = as.formula(fla), data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -457.83 -109.01   -4.51  125.53  495.77
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6280.671   1350.943  -4.649 4.15e-05 ***
## M              95.131     34.736   2.739 0.009431 **
## Ed            178.983     53.626   3.338 0.001935 **
## Po1           102.721     15.722   6.533 1.20e-07 ***
## M.F            21.191     14.569   1.454 0.154240
## U1          -6160.110   3395.044  -1.814 0.077724 .
## U2            189.483     73.930   2.563 0.014578 *
## Ineq           61.562     14.167   4.346 0.000104 ***
## Prob        -4066.106   1877.730  -2.165 0.036873 *
## Time           -1.431      5.920  -0.242 0.810259
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 198 on 37 degrees of freedom
## Multiple R-squared:  0.7892, Adjusted R-squared:  0.7379
## F-statistic: 15.39 on 9 and 37 DF,  p-value: 4.971e-10
```

```
summary(model[[5817]])
```

```
##
## Call:
## lm(formula = as.formula(fla), 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
summary(model[[11564]])
```

```
## 
## Call:
## lm(formula = as.formula(fla), data = crime)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -480.89  -89.12   -6.63  140.27  576.79 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -4911.094    960.729  -5.112 8.79e-06 ***
## M             106.659     33.877   3.148 0.003144 ** 
## Ed            189.408     48.288   3.922 0.000345 ***
## Po1           115.704     13.993   8.269 4.16e-10 ***
## U2             88.720     41.364   2.145 0.038249 *  
## Ineq           67.728     14.083   4.809 2.28e-05 ***
## Prob        -4249.756   1880.672  -2.260 0.029502 *  
## Time           -2.310      5.538  -0.417 0.678810    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 202.8 on 39 degrees of freedom
## Multiple R-squared:  0.7669, Adjusted R-squared:  0.7251 
## F-statistic: 18.33 on 7 and 39 DF,  p-value: 1.553e-10
```

From the output above, we can see the first two are the same model, and the last two are the same model. I will compare these two models with the model1, which has all the variables included.

```r
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.5.2
```

```
AIC(model1,model[[18494]],model[[5817]])

##                 df      AIC
## model1          17 650.0291
## model[[18494]] 10 639.3151
## model[[5817]]   8 640.1661

BIC(model1,model[[18494]],model[[5817]])

##                 df      BIC
## model1          17 681.4816
## model[[18494]] 10 657.8166
## model[[5817]]   8 654.9673

data.table(model1=0.7078,model18494=0.7444,model5817=0.7307)

##    model1 model18494 model5817
## 1: 0.7078     0.7444    0.7307
```

In conclusion, I would say model18494 is the best model I can found. The equation of the model is:

crime= -6426.10+93.32*M+180.12*Ed+102.65*Po1+22.34*M.F-6086.63*U1+187.35*U2+61.33*Ineq-3796.03*Prob

The corresponding AIC is 639.3151, BIC is 657.8166, and the adjusted R-squared is 0.7444.

Use the seleted model to find the crime rate in the city with data provided:

```
crimerate=-6426.10+93.32*14.0+180.12*10.0+102.65*12.0+22.34*94.0-6086.63*0.120+187.35*3.6+61.33*20.1-3796.03*0.04
crimerate

## [1] 1038.296
```