GTx: ISYE6501x - Homework 3

Muh Alif Ahsanul Islam

5/31/2019

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?

Answer:

When analyzing the trend of stock price of a company, it is often hard to determine the clear trend (up, down, or flat) because the price move up and down erratically. To solve this issue we can use exponential smoothing. The data needed is stock price.
Exponential smoothing formula:

$$S_t = \alpha x_t + (1 - \alpha) S_{t-1}$$

From formula above we can see by increasing alpha we give more weight to the current real data.
For big capitalization company I will choose alpha closer to one because I trust the current observation. Big company tend to have lower volatility and higher liquidity and thus make it a good reflection of underlying true value. Therefore, we don't need much smoothing to determine the trend (giving too many weight to past observation will obscure the most updated trend).
For small cap company I will use smaller alpha, giving less weight to current observation and more weight to previous values. Smaller company is very volatile so we can't really trust the current observation to determine the trend.

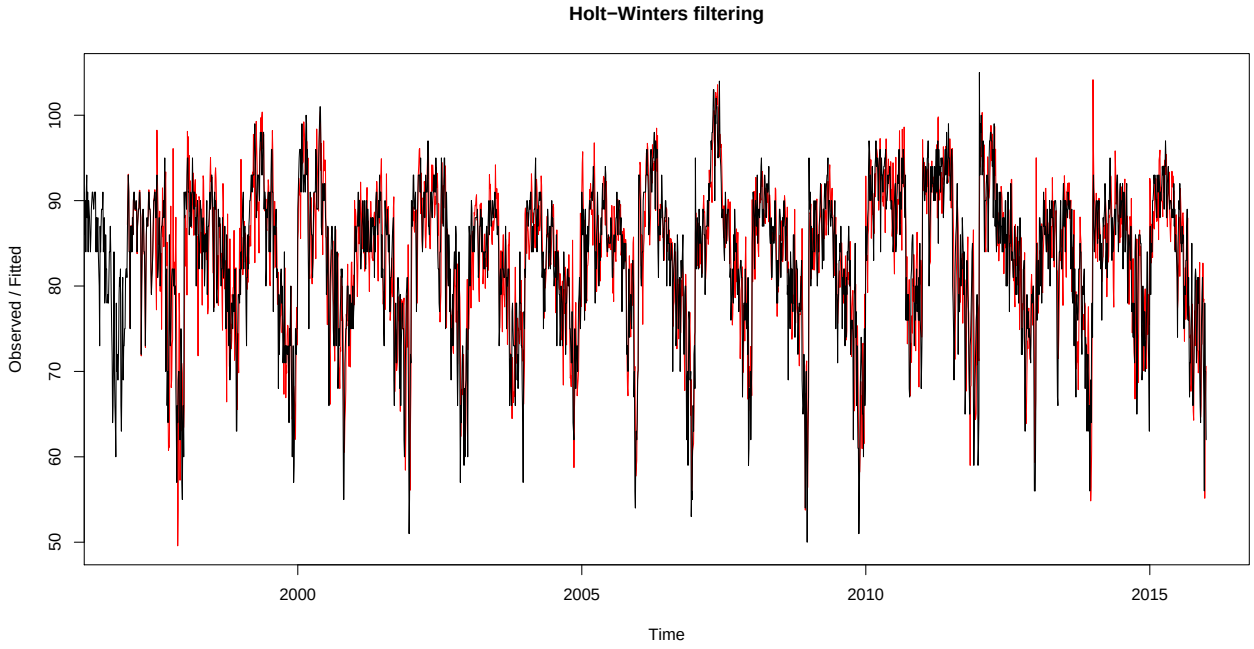
Question 7.2

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).

Procedure:
* Smooth the data by tuning alpha, beta, gamma in HoltWinters function
* Because smoothed data will not contains the first year, I will add first year to the smoothed data
* Using cusum to detect summer ends for each year from 1996-2015
* Determine whether summer end has gotten later. To answer this, I will use linear regression with year as x and summer end as y. Positive slope means the summer end has gotten latter. We set p value of the predictor/slope must be less than 0.05.

```
rm(list=ls())
temps_raw = read.table('temps.txt', header=TRUE)
temp_ts = ts(as.vector(unlist(temps_raw[,-1])),
```

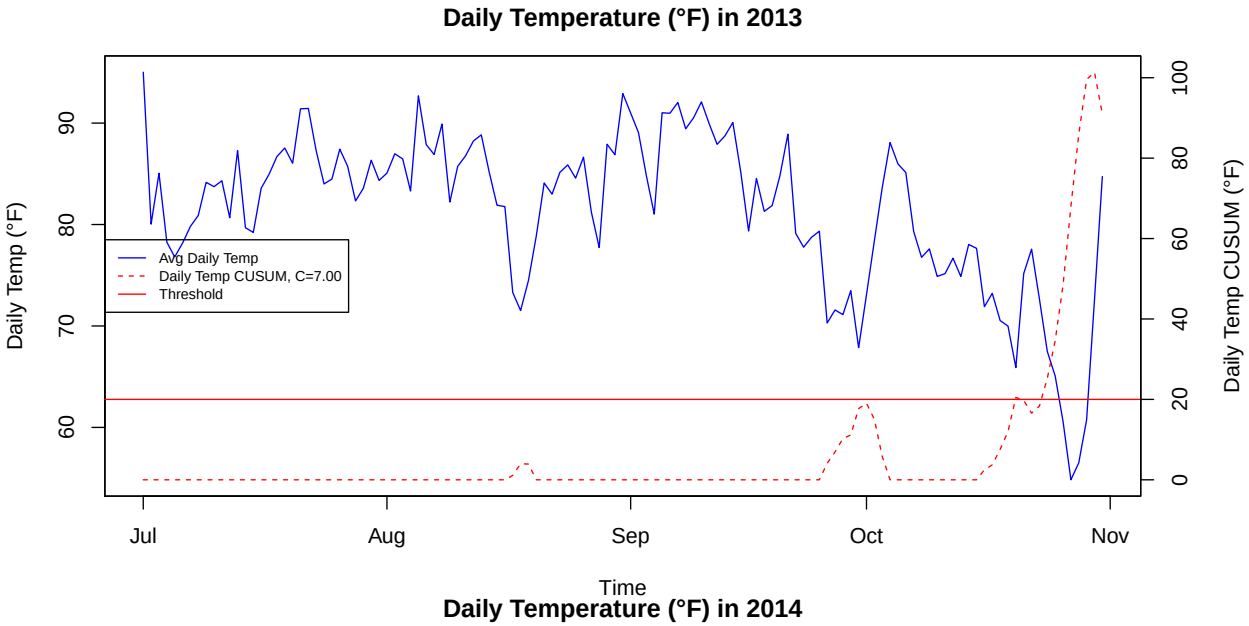
```
            start = 1996, frequency = nrow(temps_raw))
alpha_list = c(0.01, 0.04, 0.08, 0.2, 0.5)
beta_list = c(0.01, 0.1, 0.3, 0.4, 0.6, 0.8)
gamma_list = c(0.01, 0.1, 0.4, 0.5, 0.6, 0.8)
tuning_res_df = NULL
for (alpha in alpha_list){
  for (beta in beta_list){
    for (gamma in gamma_list){
        temp_hw = HoltWinters(temp_ts, alpha=alpha, beta=beta, gamma=gamma,
                              seasonal='additive')
        y_hat = temp_hw$fitted[,1]
        y = tail(temp_ts, length(y_hat))
        error = sqrt(mean(y_hat-y)^2)
        tuning_res_df = rbind(tuning_res_df,
                              data.frame(error, alpha, beta, gamma))
    }
  }

}
lowest_error = min(tuning_res_df[,"error"])
best_row = tuning_res_df[tuning_res_df$error == lowest_error,]
best_alpha = best_row$alpha
best_beta = best_row$beta
best_gamma = best_row$gamma
best_temp_hw = HoltWinters(temp_ts, alpha=best_alpha, beta=best_beta,
                           gamma=best_gamma, seasonal='additive')
plot(best_temp_hw)
```

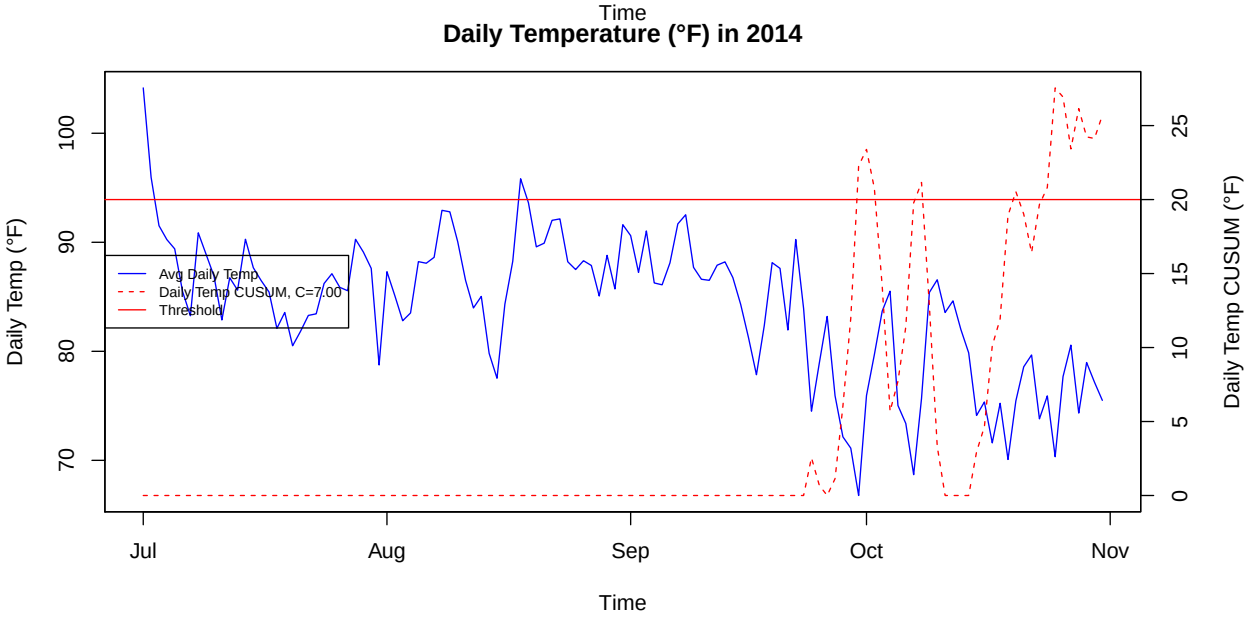


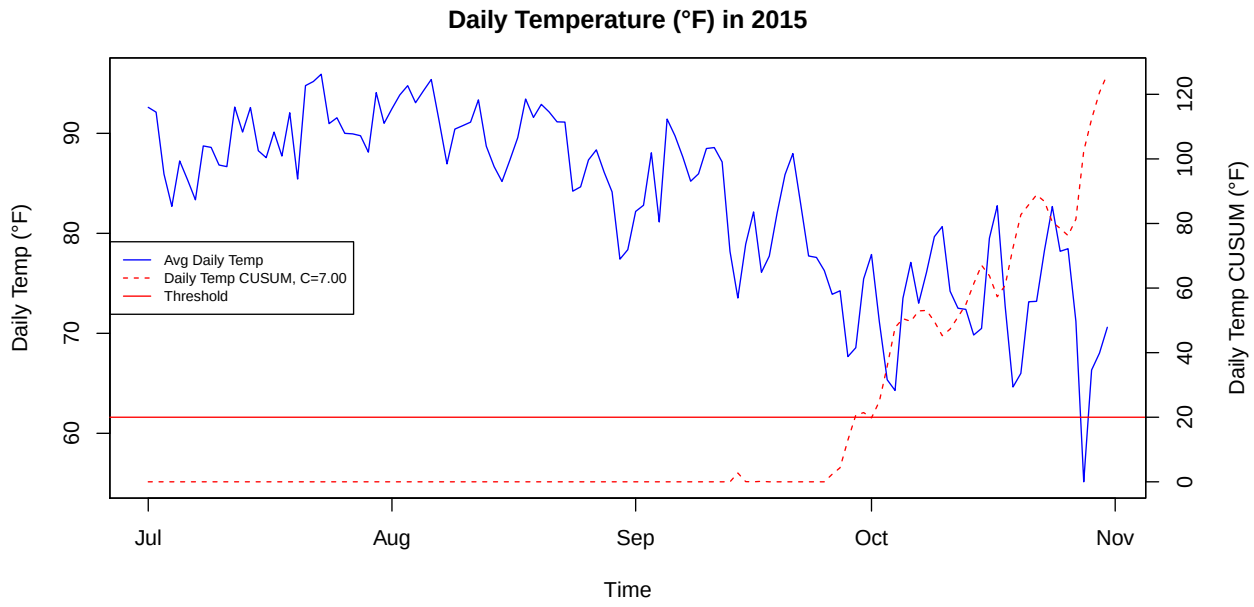

```
temp_hw_sf = matrix(best_temp_hw$fitted[,4], nrow=nrow(temps_raw))
temp_hw_fit = matrix(best_temp_hw$fitted[,1], nrow=nrow(temps_raw))
```

```
temps = read.table('temps.txt', header=TRUE)
temps$DAY = as.Date(temps$DAY, format='%d-%b')
```

```
for (j in 3:ncol(temps)){
  temps[,j] = temp_hw_fit[,j-2]
}
#temps now contains original raw 1996 data and smoothed 1997-2015 data
crit_val = 7
threshold = 20
years_col = names(temps)[c(2:length(names(temps)))]
years_int = rep(NA, length(years_col))
summer_end_idx = rep(NA, length(years_col))
summer_end_date = rep(NA, length(years_col))
summer_end_date_same_year = rep(NA, length(years_col))
temp_summer_end = rep(NA, length(years_col))
class(summer_end_date) = 'Date'
class(summer_end_date_same_year) = 'Date'
for (k in 1:length(years_col)){
  year_temp = as.matrix(temps[years_col[k]])
  avg_year_temp = mean(year_temp)
  S = rep(0, length(year_temp))
  i_summer_end = NULL
  i_flag = FALSE
  for (i in 2:length(S)){
    S[i] = max(0, S[i-1] - year_temp[i] + avg_year_temp - crit_val)
    if (S[i] > threshold & i_flag==FALSE){
      i_summer_end = i
      i_flag = TRUE
    }
  }
  summer_end_idx[k] = i_summer_end
  year_str = substr(years_col[k], 2, nchar(years_col[k]))
  years_int[k] = as.integer(year_str)
  temp_summer_end[k] = year_temp[i_summer_end]
  month_day_str = format(temps$DAY[i_summer_end], '%m-%d')
  summer_end_date[k] = as.Date(sprintf('%s-%s', year_str, month_day_str),
                               format='%Y-%m-%d')
  summer_end_date_same_year[k] = as.Date(sprintf('2019-%s', month_day_str),
                               format='%Y-%m-%d')
  if (any(c('2013', '2014', '2015') == year_str)){
    par(mar = c(5, 5, 3, 5))
    plot(x=temps$DAY, y=year_temp, type ="l", ylab = "Daily Temp (°F)",
         main = sprintf("Daily Temperature (°F) in %s", year_str),
         xlab = "Time", col = "blue")
    par(new = TRUE)
    plot(x=temps$DAY, y=S, type = "l", xaxt = "n", yaxt = "n",
         ylab = "", xlab = "", col = "red", lty = 2)
    abline(h=threshold, col='red')
    axis(side = 4)
    mtext("Daily Temp CUSUM (°F)", side = 4, line = 3)
    legend("left", c("Avg Daily Temp",
                     sprintf("Daily Temp CUSUM, C=%.2f", crit_val),
                    'Threshold'),
           col = c("blue", "red", 'red'), lty = c(1, 2), cex=0.7)
  }
}
```


Daily Temperature (°F) in 2013
Daily Temp (°F)
60
70
80
90
Daily Temp CUSUM (°F)
0
20
40
60
80
100
Avg Daily Temp
Daily Temp CUSUM, C=7.00
Threshold
Jul
Aug
Sep
Oct
Nov
Time


Daily Temperature (°F) in 2014
Daily Temp (°F)
70
80
90
100
Daily Temp CUSUM (°F)
0
5
10
15
20
25
Avg Daily Temp
Daily Temp CUSUM, C=7.00
Threshold
Jul
Aug
Sep
Oct
Nov
Time

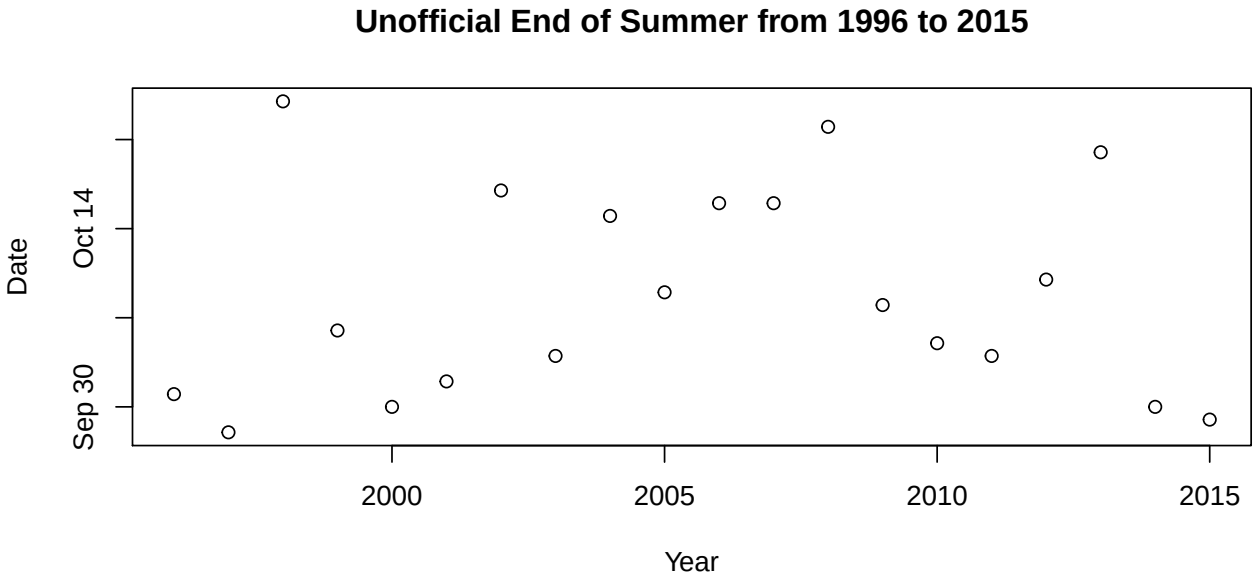


The unofficial summer end dates:

```
summer_end_date
```

```
##  [1] "1996-10-01" "1997-09-28" "1998-10-24" "1999-10-06" "2000-09-30"
##  [6] "2001-10-02" "2002-10-17" "2003-10-04" "2004-10-15" "2005-10-09"
## [11] "2006-10-16" "2007-10-16" "2008-10-22" "2009-10-08" "2010-10-05"
## [16] "2011-10-04" "2012-10-10" "2013-10-20" "2014-09-30" "2015-09-29"
```

```
plot(x=years_int, y=summer_end_date_same_year,
     main='Unofficial End of Summer from 1996 to 2015',
     xlab='Year', ylab='Date')
```

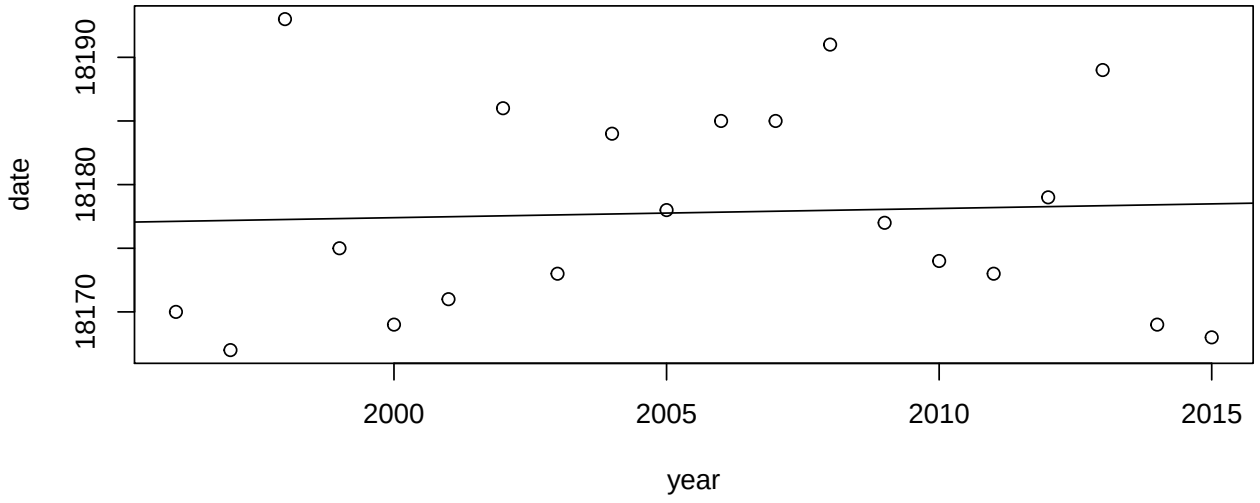



```
summer_end_df = data.frame(year=years_int, date=as.numeric(summer_end_date_same_year))
model = lm(date ~ year, data = summer_end_df)
summary(model)
```

```
##
```

```
## Call:
## lm(formula = date ~ year, data = summer_end_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.486  -6.635  -1.692   7.110  15.741
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.803e+04  6.563e+02  27.478 3.77e-16 ***
## year        7.218e-02  3.272e-01   0.221    0.828
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.438 on 18 degrees of freedom
## Multiple R-squared:  0.002696,   Adjusted R-squared:  -0.05271
## F-statistic: 0.04866 on 1 and 18 DF,  p-value: 0.8279
```

```
{plot(date ~ year, data=summer_end_df)
abline(lm(date ~ year, data = summer_end_df))}
```



Answer to Question 7.2

From the summary of the linear regression model, we understand that the coefficient of year is 0.072 with p value 0.828. The p value is higher than our standard 0.05 so we regard there are no strong evidence for positive trend. This means we do not have evidence that summer end is getting later.

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.

Answer

Analytics can be used to help increase productivity of your banana farm. Using linear regresion model, we can know what type of stimulation we need to give to the plant to make it produce more bananas. Target variable (y): number of bananas produced
Predictor:
1. Temperature
2. Humidity
3. Fertilizer
4. Water
5. Precipitation

Question 8.2

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

```
rm(list=ls())
raw_data = read.table('uscrime.txt', header=TRUE)
test_data = 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)
model = lm(Crime~., data=raw_data)
summary(model)
```

```
##
## 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 model summary above, we can see the significant predictors are (based on p value): Ed, Ineq, M, Prob, U2, Po1. The R-squared is 0.8031 and Adjusted R-squared 0.7078.

Redo modelling with only siginificant predictors, we will call this new model 'significant model' and old model 'old model'.

```
model_significant = lm(Crime ~ Ed + Ineq + M + Prob + U2 + Po1, data=raw_data)
summary(model_significant)
```

```
##
## Call:
## lm(formula = Crime ~ Ed + Ineq + M + Prob + U2 + Po1, data = raw_data)
##
## 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 ***
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## M             105.02      33.30   3.154  0.00305 **
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## U2             89.37      40.91   2.185  0.03483 *
## Po1           115.02      13.75   8.363 2.56e-10 ***
## ---
## 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
```

```
predict(model_significant, test_data)
```

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

The R-squared for this model is is 0.7659 and Adjusted R-squared 0.7307. The significant model has lowest R-squared but higher adjusted R-squared.
Source click here:
The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance.

This means the significant model is better than old model. And we will use it to predict the test data.

Answer to Question 7.2

The predicted crime is:

```
predict(model_significant, test_data)
```

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