ISYE 6501 Week 3 Homework

2024-06-02

Load data and libraries

```
temps <- read.table("/Users/avery/Desktop/OMSA/ISYE 6501 Analytics Modeling/week_3_Homework-summer/week
crime <- read.table("/Users/avery/Desktop/OMSA/ISYE 6501 Analytics Modeling/week_3_Homework-summer/week

# libraries
library(tidyverse)
```

Question 7.1

I work at an education technology company, and I think that exponential smoothing could be a useful tool for tracking student performance over time. A student's performance (as measured by tests, quizzes, homeworks, etc.) should be highly correlated with their past performance, as students who perform better on previous assessments and homeworks should have a better understanding of the material, enabling them to perform well on future assignments. However, there is also a lot of room for random effects, from a bad day resulting in a slightly low test score to forgetting about a homework assignment and getting a 0. This type of model would require a students' grade data to be collected for a specific class or subject, and it would be important for all the grades to be standardized on the same scale to ensure a grade of 3 on a quiz worth 4 points won't pull down the average. I believe the value of the smoothing parameter should be closer to 1, as a student's previous performance should have the biggest impact on their future performance, so a value between 0.7 and 0.9 should allow the model to accurately account for any random effects. Additionally, this type of model would be very useful to see if specific interventions, such as tutoring, are helping students. We could look at the trends in the data to see students whose grades are decreasing and might be in need of support, and then after the intervention is given, we could examine whether or not the trend has changed to increasing.

Question 7.2

Before I can build the exponential smoothing model, I need to transform the temperature data so all the temperature values are in a vector and converted to a ts object.

```
tempTransformed <- temps %>% pivot_longer(cols = X1996:X2015) %>%
  mutate(date = as.Date(paste(DAY, str_remove(name, "X"), sep = "-"), "%d-%b-%Y")) %>%
  select(date, temp = value) %>% arrange(date)

temp_ts <- ts(tempTransformed$temp, start = 1996, frequency = 123)
```

Now that my data is formatted properly, I can use the Holt-Winters function to model the data. Since I am looking to see if summers are ending later, which would imply that the temperature cycle is changing across the years, I will use multiplicative seasonality to account for the potential increase in variation.

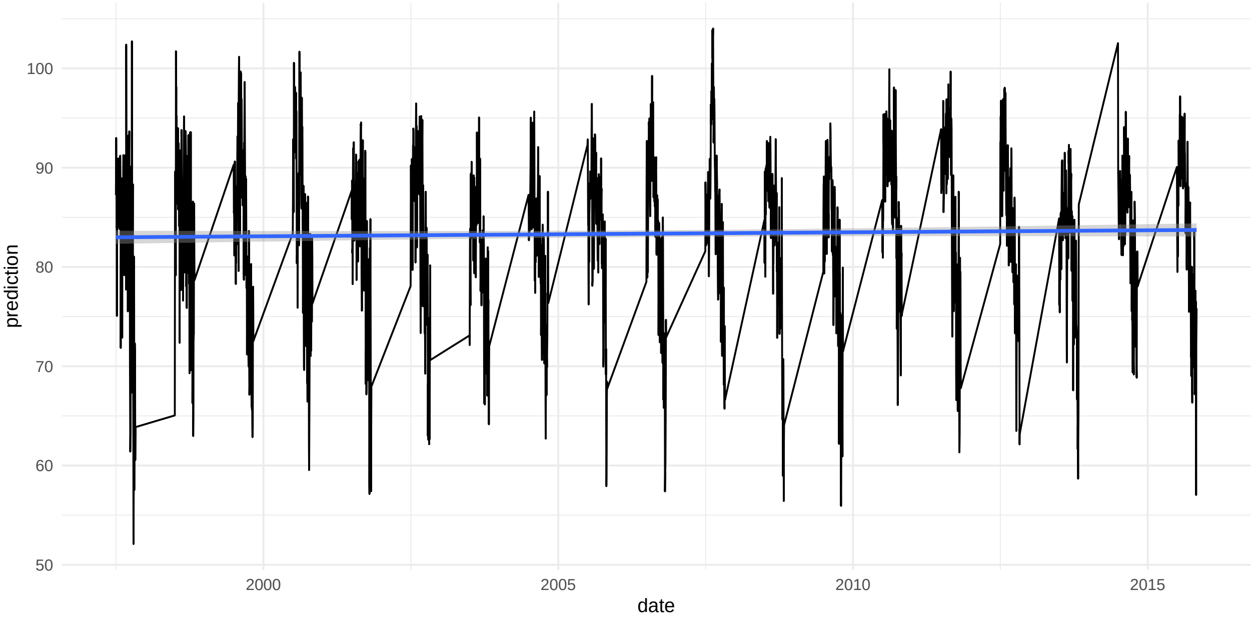
```
esModel <- HoltWinters(temp_ts, seasonal = "multiplicative")

print(esModel[3:5])
```

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

```
##
## $beta
## beta
##    0
##
## $gamma
##     gamma
## 0.5495256
```

The alpha value close to 1 indicates that there was not a great deal of variability due to randomness in this data set, and the beta value of 0 indicates that there were not significant trends within the data set. This beta value suggests there has not been an overall increase in summer temperatures, but I will do a more detailed analysis into the length of summer next. The gamma value reflects the weighting of the seasonality factor, so a gamma value greater than 0 confirms the notion that summer temperature patterns are fairly cyclical.

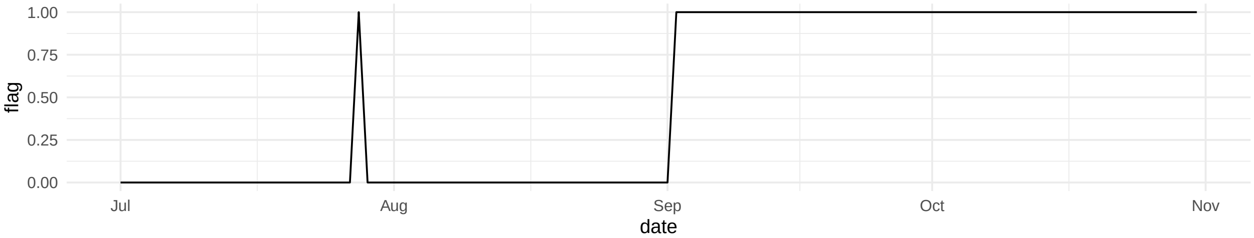
```
tempTransformed <- tempTransformed %>%
  mutate(prediction = append(rep(NA, 123), esModel$fitted[,1]))

ggplot(tempTransformed %>% filter(!is.na(prediction)), aes(date, prediction)) +
  geom_line() + theme_minimal() + geom_smooth(method = "lm")
```



The above graph shows the smoothed data points for each summer after 1996 plotted in a line graph, with a simple linear regression model overlaid. While the regression model does appear to have a slight positive slope, the data does not appear to be changing very significantly over time, so I will need a different approach to determine whether or not summers have gotten longer. For this analysis, I will define a longer summer as a summer where there has not been a substantial decrease in temperature by the expected date. In order to identify if summers are getting longer, I will use the CUSUM approach to find out when summer temperatures start decreasing substantially. I will build the model based on the predicted values from the Holt-Winters model, since these values are adjusted to account for some of the random variation that occurs, meaning the change detection model will be less likely to detect an errant decrease.

First, I will find the expected date of temperature decrease by looking at the 1996 data. Because temperatures are fairly consistent during July and then start to drop off in August and September, I will usage the first 31 days (the month of July) to calculate the mean against which following data points are compared. My critical value will be equal to the standard deviation of the data set to help account for some of the random variation, and the threshold will be 3 times the standard deviation to ensure the model only picks up significant changes

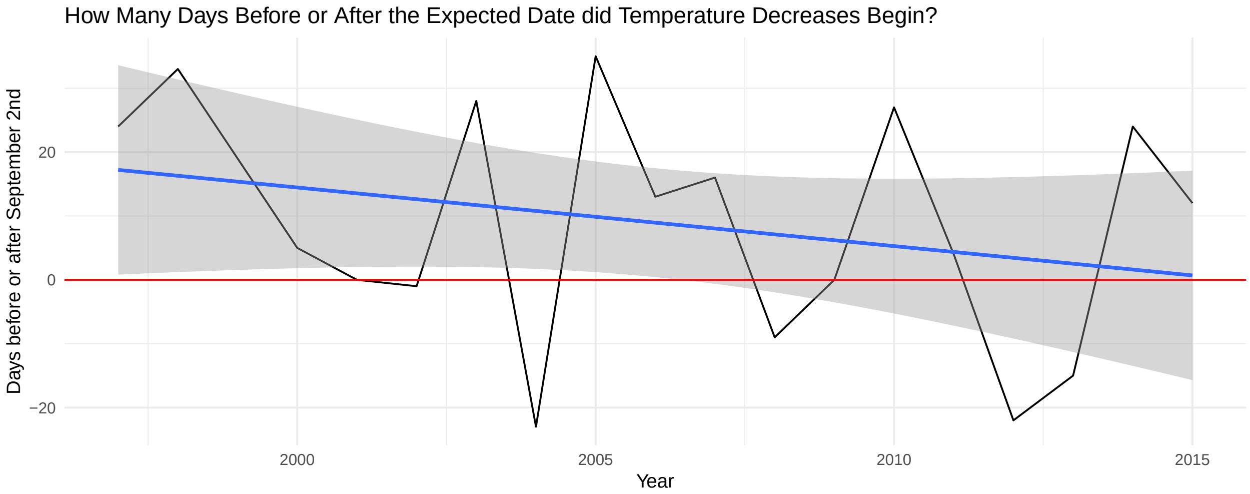
```r
data_1996 <- tempTransformed %>%
  filter(between(date, as.Date("1996-1-1"), as.Date("1996-12-31"))) %>%
  mutate(flag = NA)

mean = mean(data_1996$temp[1:31])
sd = sd(data_1996$temp[1:31])
threshold = sd*3

for(row in 1:nrow(data_1996)){
  if(row == 1){
    data_1996$st[row] = 0
  } else {
    data_1996$st[row] = max(0, data_1996$st[row - 1] + (mean - data_1996$temp[row] - sd))

    if(data_1996$st[row] >= threshold){
      data_1996$flag[row] <- "Yes"
    }
  }
}
```



September 2nd is the first date at which a decrease is flagged and the decrease continues, so I will use September 2nd as the expected decrease date for the other years.

Now that I know on which date I expected to see significant decreases, I can apply the same change detection model to the other years and then calculate how many days before or after September 2nd the significant temperature decrease began. As with the 1996 data, I will ignore points where only a single day or a few days were flagged as a decrease, and use the first point at which 5 or more days in a row are flagged as the official date when the temperature started to drop.

```r
cusumData <- data.frame(years = 1997:2015, dateDecrease = NA, difference = NA)

for(i in 1:19){

  data <- data.frame(date = tempTransformed$date[(i * 123 + 1):((i + 1) * 123)],
                     temp = tempTransformed$prediction[(i * 123 + 1):((i + 1) * 123)],
                     st = NA, flag = NA)

  mean = mean(data$temp[1:31])
  sd = sd(data$temp[1:31])
  threshold = sd*3

  for(row in 1:nrow(data)){
    if(row == 1){
      data$st[row] = 0
      data$flag[row] <- "No"
    } else {
      data$st[row] = max(0, data$st[row - 1] + (mean - data$temp[row] - sd))

      if(data$st[row] >= threshold){
        data$flag[row] <- "Yes"
      } else {
        data$flag[row] <- "No"
      }
    }
  }

  for(row in 1:nrow(data)){
    if(data$flag[row] == "Yes" & data$flag[row + 1] == "Yes" &
       data$flag[row + 2] == "Yes" & data$flag[row + 3] == "Yes" &
       data$flag[row + 4] == "Yes"){
      cusumData$dateDecrease[i] <- as.Date(data$date[row])
      cusumData$difference[i] <- as.numeric(difftime(data$date[row],
                                                     data$date[64], unit = "days"))
      break
    }
  }
}
```



Data points above the red line indicate summers where significant decreases in temperature did not start until after September 2nd, while data points below the line indicate years where the temperature drops occurred before September 2nd. While there is a lot of variability in the graph, it does not appear that there has been an increased delay in temperatures dropping (i.e., a longer summer) in recent years. In fact, a linear model (shown in blue) indicates that, if anything, summers have been getting shorter in recent years (at least compared to the previous years in this data set). Therefore, I cannot conclude that the end of summer has gotten later over the past 20 years.

Question 8.1

Linear models are frequently used to describe and predict student performance, and I think this approach would be well-suited for predicting student performance on college entrance exams. Additionally, once the model was generated, teachers could see which factors were the most important to increasing exam scores and focus on helping students in those areas. The following predictors could be used:

1. GPA - a student's overall academic performance is likely highly correlated with how well they perform on any assessment
2. Average score on in-class exams - while exam scores affect GPA, GPA is also influenced by homework, projects, etc.. A student who does very well on other assignments but has a great deal of test anxiety might not perform well on an entrance exam due to anxiety, so separating out this predictor could help to better account for actual performance in the real testing situation.
3. Performance on similar standardized exams - students often take preparatory entrance exams or multiple college entrance exams, so these scores should be also highly correlated with how students do on other entrance exams
4. Household income - students' socioeconomic statuses are correlated with test scores due to factors like access to outside resources and availability of parents to help students with homework, so incorporating household income in the model could help ensure that students with less access to resources at home receive extra support in school

Question 8.2

I will start by building a linear model that predicts crime based on all the other variables in the data set

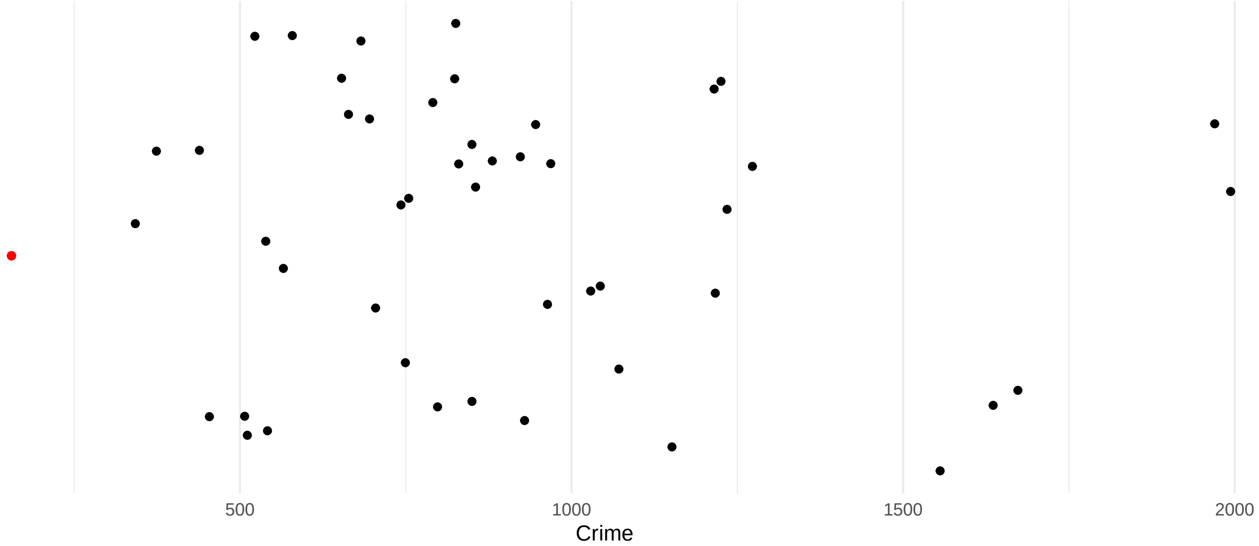
```r
linearModel <- lm(Crime ~ ., crime)

summary(linearModel)
```

```
##
## 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
```

Since there are so many predictor variables in this model, I will measure model fit using the adjusted R-squared value. The adjusted R-squared is 0.7078, which means that about 71% of the variability in crime is accounted for by the predictors included in the model, so this model appears to pretty crime very well. However, only 4 of the predictor variables are significant, so this model may be including unnecessary information.

Now I will use the predict function to create a prediction for the example observation

```
observation <- data.frame(M = 14, 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)

prediction <- predict(linearModel, observation)
```



According to this model, the crime rate in the example city would be 155, which is lower than any other observation in the data set.

I will now build a linear model that only includes the significant covariates in order to see if this model produces a different response.

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

summary(simplifiedLinearModel)
```

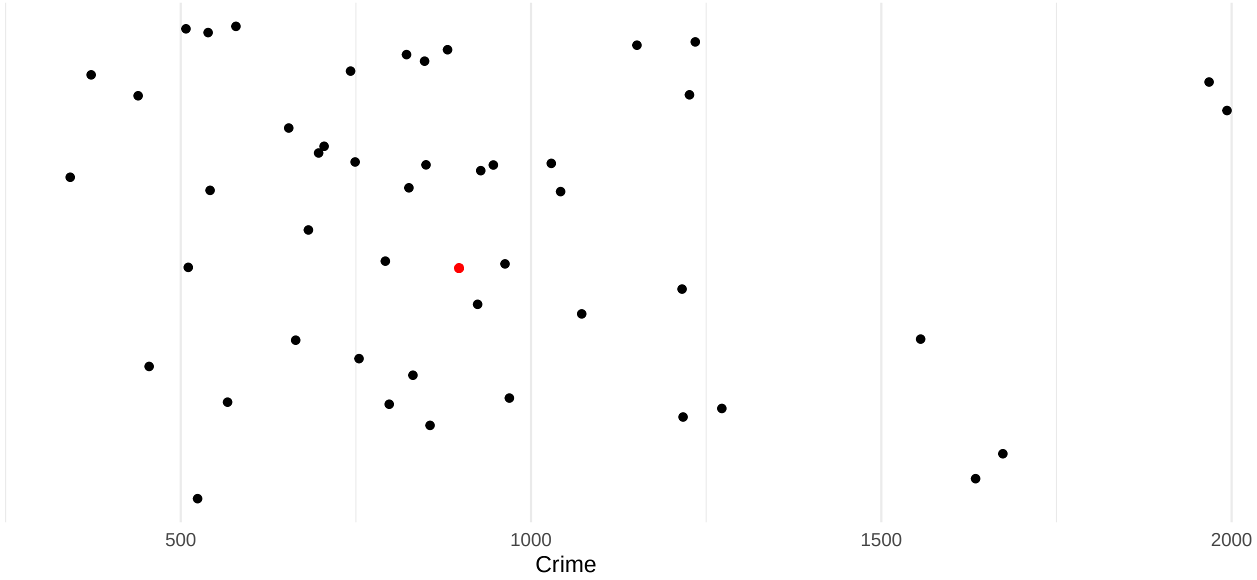
```
##
## Call:
## lm(formula = Crime ~ M + Ed + Ineq + Prob, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -532.97 -254.03  -55.72  137.80  960.21
##
## Coefficients:
```

```
##              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
```

The adjusted R-squared here is 0.1927, which is a significant decrease from the full model. Despite most of the predictors in the full model not being significant, it appears that all of them are needed to create a model that explains most of the variability in the response variable. A good next step would be exploring the interactions between some of the predictors in case those interactions are driving the adjusted R-squared in the full model.

Now I will use the predict function to create a prediction for the example observation

```
prediction <- predict(simplifiedLinearModel, observation)
```



This more simplified model predicted a crime rate of 897, which is more aligned with the other data points in the Crime data set. However, since the simplified model was a poorer fit according to adjusted R-squared, more exploration would need to be done in order to determine which prediction is more likely to be accurate.