STAT 2131
Orly Olbum

HW 8

1. (Leave one out)

$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times p}$ full rank design matrix

$H = X(X^TX)^{-1}X^T$, h_{ii} are i^{th} diagonal element of H.

$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$ is response.

For $i = 1, \ldots, n$, $X_{(-i)} \in \mathbb{R}^{(n-1)\times p}$ be design matrix with i^{th} row removed $\rightarrow$ $Y_{(-i)} \in \mathbb{R}^{n-1}$

$\hat{\beta}_{(-i)} = (X_{(-i)}^T X_{(-i)})^{-1} X_{(-i)}^T$ is OLS w/o i^{th} sample

@ Show that $(X_{(-i)}^T X_{(-i)})^{-1} = (X^TX)^{-1} + (1-h_{ii})^{-1}(X^TX)^{-1}x_i x_i^T (X^TX)^{-1}$

If $X_{(-i)}$ is the leave-one-out design matrix,

x_i is the i^{th} row of X and

$\hat{\beta}_{(i)} = (X_{(-i)}^T X_{(-i)})^{-1} X_{(-i)}^T Y_{(i)}$

The LOO residual is $\epsilon_{(-i)} = y_i - x_i^T \hat{\beta}_{(i)}$

And we can rewrite $X_{(i)}^T X_{(i)} = (X^TX - x_i x_i^T)$

Since $OVC = \sum_{i=1}^n \left(\frac{\hat{\epsilon}_i}{1-h_{ii}}\right)^2$, and $h_{ii} = x_i^T (X^TX)^{-1} x_i$.

$$(X_{(-i)}^T X_{(-i)})^{-1} = (X^TX)^{-1} + \frac{(X^TX)^{-1}x_i x_i^T (X^TX)^{-1}}{1-h_{ii}}$$

$$= (X^TX)^{-1} + (1-h_{ii})^{-1}(X^TX)^{-1} x_i x_i^T (X^TX)^{-1}$$

⑥ Show that LOOCV error

$PRESS = \sum_{i=1}^n (y_i - x_i^T \hat{\beta}_{(-i)})^2$ can be written $= \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{1-h_{ii}}\right)^2$

where $\hat{y}_i$ is i^{th} element of $HY \in \mathbb{R}^n$

If @ holds, we can rewrite $\hat{\beta}_{(-i)}$:

$$\hat{\beta}_{(-i)} = \left[(X^TX)^{-1} + (1-h_{ii})^{-1}(X^TX)^{-1}x_i x_i^T (X^TX)^{-1}\right](X^TY - x_i y_i)$$

↳ where $X_{(-i)}^T Y_{(-i)} = (X^TY - x_i y_i)$

$$= (X^TX)^{-1}XY - (1-h_{ii})^{-1}(X^TX)^{-1}x_i x_i^T (X^TX)^{-1}X^TY$$

$$- (X^TX)^{-1}x_i y_i - (1-h_{ii})^{-1}(X^TX)^{-1}x_i x_i^T (X^TX)^{-1}x_i y_i$$

$$= \hat{\beta} - \left[(1-h_{ii})^{-1}(X^TX)^{-1}x_i\right]\left[y_i(1-h_{ii}) - x_i^T\hat{\beta} + h_{ii}y_i\right]$$

$$= \hat{\beta} - \frac{(X^TX)^{-1}x_i \epsilon_i}{1-h_{ii}}$$

So, $\epsilon_{(i)} = y_i - x_i^T \hat{\beta}_{(i)}$

$$= y_i - x_i^T\left[\hat{\beta} - \frac{(X^TX)^{-1}x_i\epsilon_i}{1-h_{ii}}\right]$$

$$= \epsilon_i + \frac{h_{ii}\epsilon_i}{1-h_{ii}} = \frac{(1-h_{ii})\epsilon_i + h_{ii}\epsilon_i}{1-h_{ii}} = \boxed{\frac{\epsilon_i}{1-h_{ii}}} \rightarrow$$

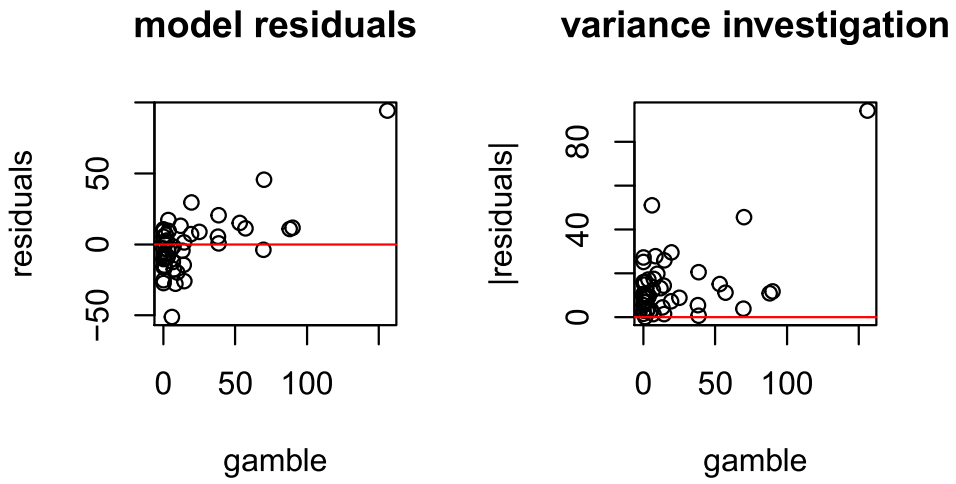
Question 2

(a) Regress gamble onto the other four predictors. Do you see any evidence that the mean model or constant variance assumption is violated? Are the errors normally distributed?

```
model = lm(gamble ~ sex + status + income + verbal, data = gamble)
summary(model)
```

```
##
## Call:
## lm(formula = gamble ~ sex + status + income + verbal, data = gamble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.55565   17.19680   1.312   0.1968
## sex         -22.11833    8.21111  -2.694   0.0101 *
## status        0.05223    0.28111   0.186   0.8535
## income        4.96198    1.02539   4.839 1.79e-05 ***
## verbal       -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

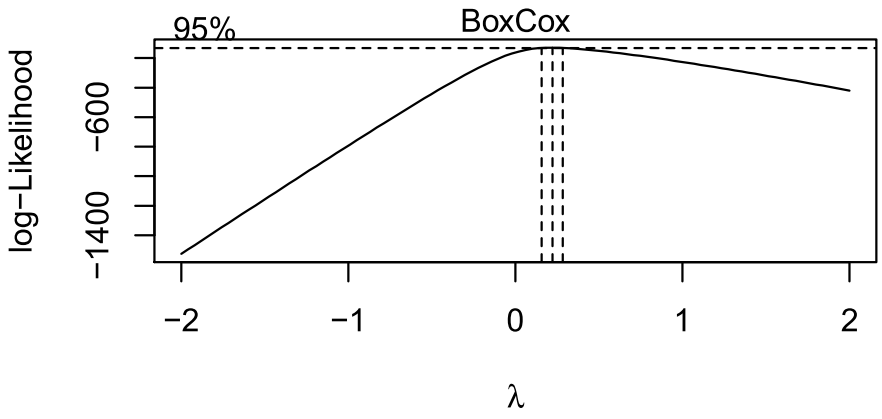
```
par(mfrow = c(1, 2))
plot(gamble$gamble, model$residuals, main = "model residuals", xlab = "gamble",
     ylab = "residuals")
abline(h = 0,col = "red")
plot(gamble$gamble, abs(model$residuals), main = "variance investigation", xlab = "gamble",
     ylab = "|residuals|")
abline(h = 0, col = "red")
```



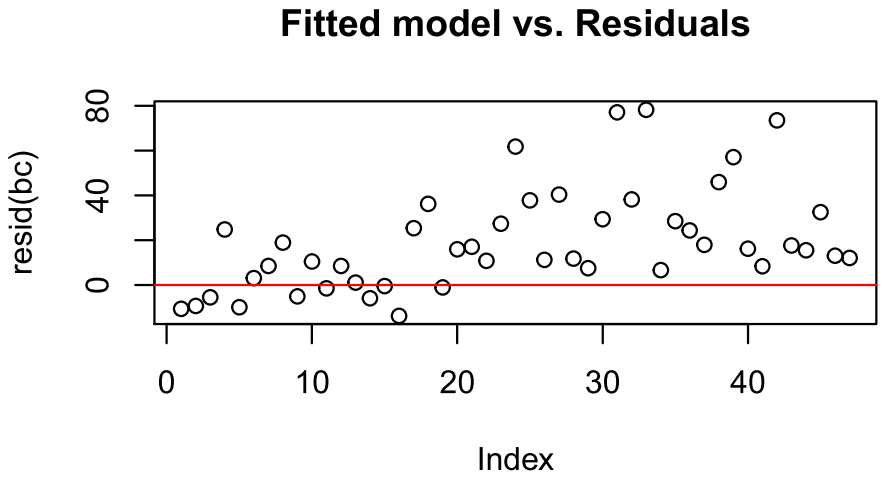
From the residual plot we can see a potential outlier and a potential for the data not to fit a linear model very well. A plot of the absolute value of the residuals shows non-constant variance as well, and we will have to do something about that. The summary tells us similar things - the R^2 is not ideal, and only 2 of the predictors are significant at an alpha of 0.05.

(b) Use Box Cox with gamma > 0 to suggest a transformation of the response, ~Y , so that ~Y satisfies the usual mean and variance assumptions. Plot ^~Y vs. the estimated residuals. Does your new model appear to satisfy the constant variance assumption? Does ~Y appear to be normally distributed? (Hint: the Box Cox example code can be found in Transformations.Rmd on blackboard. You may need to add a small delta > 0 to gamble to get the function to work, since the function requires Y > 0. delta can be chosen to be arbitrarily small, like 10^-8).

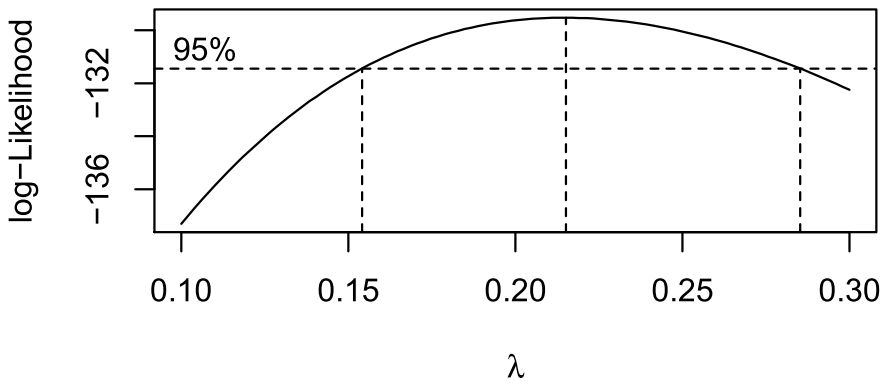
```
gamble$gamble2 = gamble$gamble + 0.0000001 # correct Y to satisfy boxcox requirements
model2 = lm(gamble2 ~ sex + status + income + verbal, data = gamble)

par(mfrow = c(1, 1))
bc = boxcox(model2)
mtext("BoxCox", 3)
```



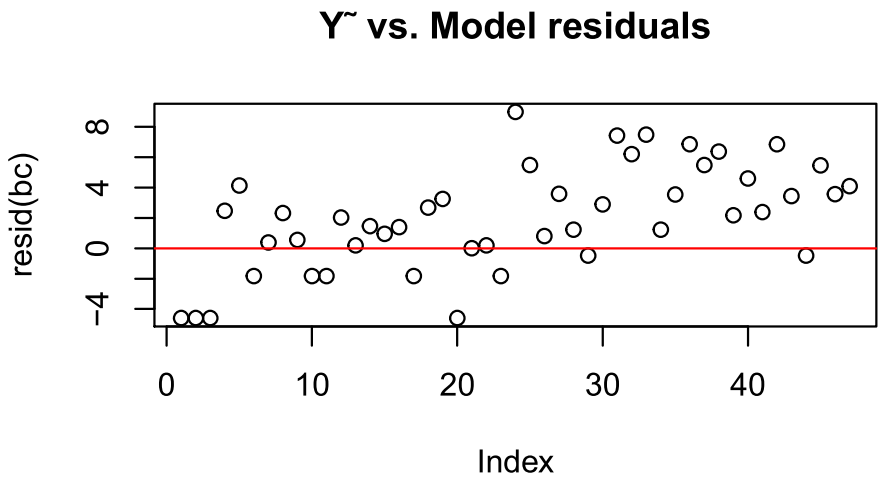

```
plot(fitted(model2), resid(bc), main = "Fitted model vs. Residuals")
abline(h = 0,col = "red")
```




```
bc2 = boxcox(model2, lambda = seq(.1, .3, by = .01))
```




```
lambda = .21
Y.tilde = ((gamble$gamble2^lambda) - 1)/lambda
plot(Y.tilde, resid(bc), main = "Y~ vs. Model residuals")
abline(h = 0,col = "red")
```



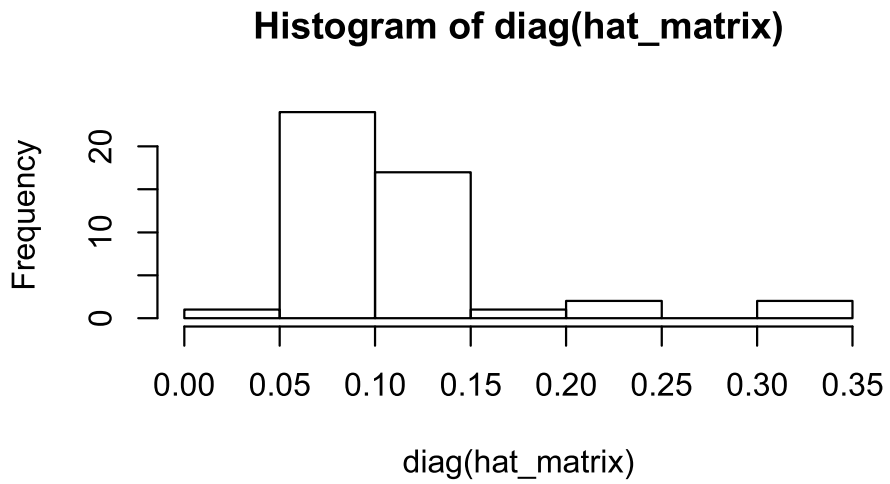
The first graph is the boxcox function onto the new model that incorporates the non-zero Y values (just barely non-zero, to comply with boxcox model requirements). The second plot graphs the residuals against the fitted values, and we can see that they are slightly slanted up but much better than before. Now we can analyze the boxcox interval to determine a lambda appropriate for transforming Y. The next plot shows this interval, and lambda can be narrowed down to about .21. Y~ is calculated using this lambda and the next plot shows the model residuals plotted against the transformed Y, Y~, and we can see that the errors are far less correlated.

(c) Compute the hat matrix and plot a histogram of the leverage scores.

```r
model2 = lm(gamble2 ~ sex + status + income + verbal, data = gamble)
X = model.matrix(model2)
hat_matrix = X%*%(solve(t(X)%*%X)%*%t(X))
diag(hat_matrix)
```

```
##          1          2          3          4          5          6          7
## 0.07988226 0.10851291 0.06347643 0.10273955 0.13866946 0.16378563 0.06893316
##          8          9         10         11         12         13         14
## 0.07110422 0.07051619 0.10871017 0.10096045 0.07004982 0.10529294 0.07363771
##         15         16         17         18         19         20         21
## 0.05826146 0.08372118 0.14000342 0.11994753 0.06905872 0.08776687 0.05010691
##         22         23         24         25         26         27         28
## 0.11496664 0.06693234 0.12380463 0.10273229 0.11229428 0.10583404 0.11010633
##         29         30         31         32         33         34         35
## 0.07729898 0.05145946 0.23950314 0.10677583 0.22134389 0.10707672 0.31180294
##         36         37         38         39         40         41         42
## 0.04659071 0.08212433 0.09761361 0.09155208 0.08739786 0.11648008 0.30160877
##         43         44         45         46         47
## 0.08715765 0.06897428 0.06737462 0.08718333 0.07887418
```

```r
hist(diag(hat_matrix))
```



(i) Why should one be concerned if there are any abnormally large leverage scores? Do you see any evidence of large leverage points in these data?

Abnormally large leverage scores will potentially affect the model fit in a few different ways. If it is an outlier in Y, it may be further away from the desired fit and sway the regression line in one way or another. If it is an outlier in X, it may not effect the model itself, but is still far away from most of the data. If it is an outlier in both X and Y, the regression line will cross it and it still sways the linear regression fit. We can use the leverage scores, or the diagonals of the hat matrix, to determine these outliers. The histogram above shows a potential large leverage score in the .30-.35 box, since it is further from the main group of the data. We can use the leverage score/s of this point/s to diagnose a potential outlier, and determine what if anything should be done about it.

(ii) Re-estimate the model from part (b) after removing the points with leverage scores $> 2p/n$. Do the parameter estimates or standard errors change substantially?

2p/n is equal to twice the average of the leverage scores. Here, that is:

```
2*mean(diag(hat_matrix))
```

```
## [1] 0.212766
```

```
p = 5
n = 47
2*p/n
```

```
## [1] 0.212766
```

Now we can remove the points with leverage scores above this value and re-estimate the model. Observations 31, 33, 35, and 42 have leverage scores above the threshhold.

```
diag(hat_matrix) > 2*p/n
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    14    15    16    17    18    19    20    21    22    23    24    25    26
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    27    28    29    30    31    32    33    34    35    36    37    38    39
## FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
##    40    41    42    43    44    45    46    47
## FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
```

```
which(diag(hat_matrix) > 2*p/n)
```

```
## 31 33 35 42
## 31 33 35 42
```

```
adj.data = gamble[-c(31, 33, 35, 42),]

model3 = lm(gamble ~ sex + status + income + verbal, data = adj.data)
summary(model3)
```

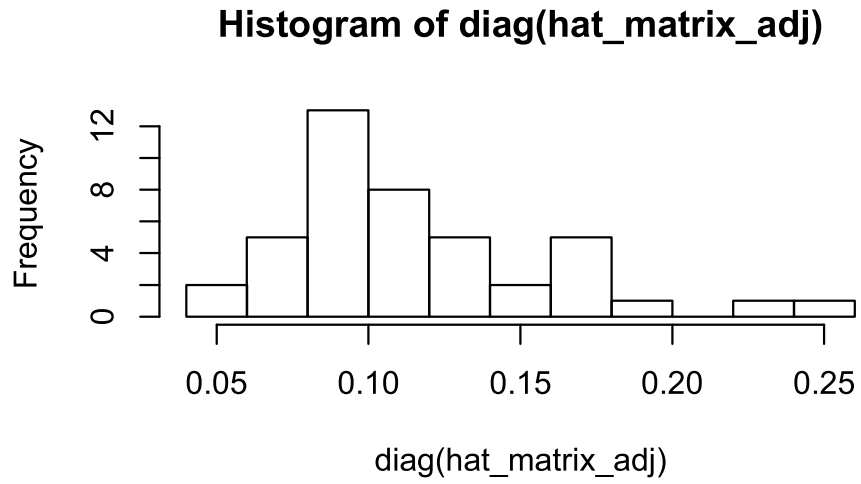
```
##
## Call:
## lm(formula = gamble ~ sex + status + income + verbal, data = adj.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.346 -12.001  -2.798   7.379  97.139
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  32.36969   25.36559   1.276   0.2097
## sex         -22.58401    9.01035  -2.506   0.0166 *
## status        0.04236    0.29890   0.142   0.8881
## income        4.04125    1.57414   2.567   0.0143 *
```

```
## verbal       -3.76615    2.76219  -1.363   0.1808
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.47 on 38 degrees of freedom
## Multiple R-squared:  0.3581, Adjusted R-squared:  0.2905
## F-statistic: 5.299 on 4 and 38 DF,  p-value: 0.001716
```

```
X.adj = model.matrix(model3)
hat_matrix_adj = X.adj%*%(solve(t(X.adj)%*%X.adj)%*%t(X.adj))
diag(hat_matrix_adj)
```

```
##          1          2          3          4          5          6          7
## 0.09405380 0.11614772 0.07813603 0.12660492 0.15986646 0.16586079 0.07519834
##          8          9         10         11         12         13         14
## 0.08203647 0.08246894 0.11615946 0.11544078 0.07115959 0.16595418 0.09353023
##         15         16         17         18         19         20         21
## 0.06325877 0.09710536 0.24314762 0.17409644 0.08412798 0.09543357 0.05179887
##         22         23         24         25         26         27         28
## 0.11739760 0.11325456 0.22509106 0.13790038 0.13696518 0.16231507 0.16782588
##         29         30         32         34         36         37         38
## 0.10429427 0.07527210 0.13394424 0.13283490 0.05739638 0.08526421 0.14248531
##         39         40         41         43         44         45         46
## 0.18408660 0.11090536 0.11900855 0.09299035 0.08274807 0.08139259 0.09459478
##         47
## 0.09044624
```

```
hist(diag(hat_matrix_adj))
```




```
coef(summary(model2))[, 1]; coef(summary(model3))[, 1] # coefficients
```

```
##  (Intercept)          sex       status       income       verbal
##  22.55565073 -22.11833009   0.05223384   4.96197922  -2.95949350
```

```
##  (Intercept)          sex       status       income       verbal
##  32.36968574 -22.58401237   0.04235876   4.04124750  -3.76614850
```

```
coef(summary(model2))[, 2]; coef(summary(model3))[, 2] # standard errors
```

```
## (Intercept)         sex      status      income      verbal
##  17.1968034   8.2111145   0.2811115   1.0253923   2.1721503

## (Intercept)         sex      status      income      verbal
##  25.3655947   9.0103506   0.2989023   1.5741423   2.7621927
```

It does not appear that the coefficients or the standard errors change too much after removing the points that have large leverage scores. The intercept changes the most (magnitude), and verbal changes slightly, but coefficients appear to stay in the same direction and generally of the same magnitude after removing the points.

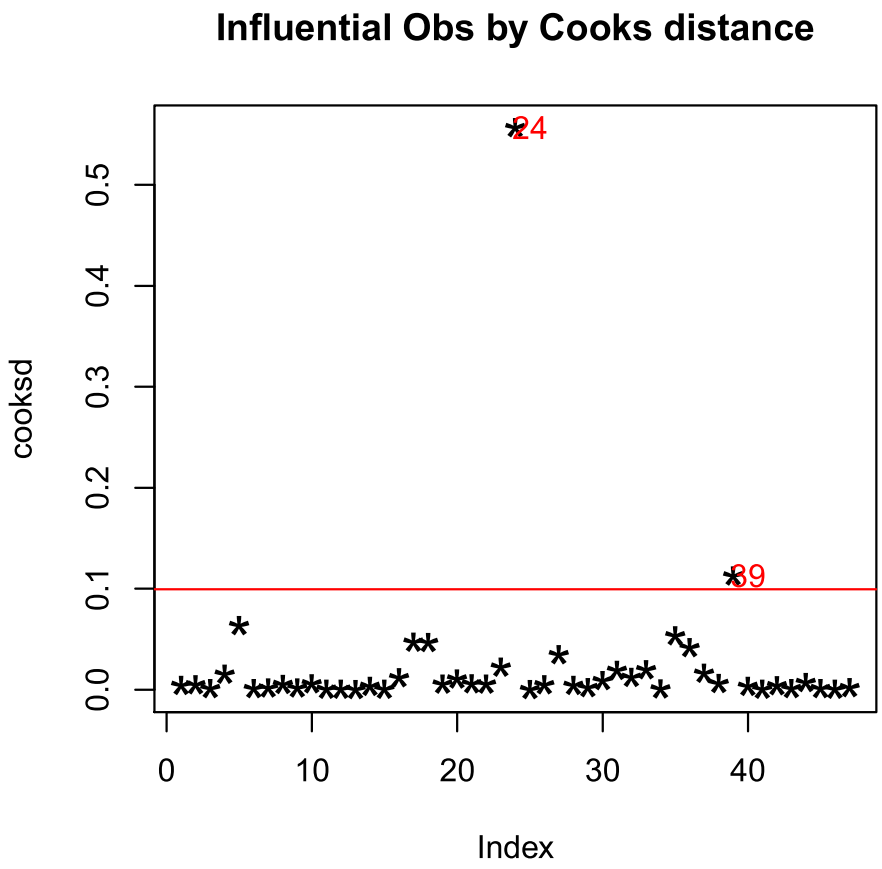
(d) Compute the Cook's distance for each of the n points. Do any of the points appear to be influential points?

For Cook's distance, we might use a cutoff of 4/(n-p), and determine that the points with distances over this value are potentially influential to the model fit. We can first look at a histogram of the distances (similar to the leverage scores - it's always helpful to SEE the data).

```
4/(n-p)
```

```
## [1] 0.0952381
```

```
par(mfrow = c(1, 1))

cooksd = cooks.distance(model)
plot(cooksd, pch = "*", cex = 2, main = "Influential Obs by Cooks distance")
abline(h = 4*mean(cooksd, na.rm = TRUE), col = "red")
text(x = 1:length(cooksd) + 1, y = cooksd, labels = ifelse(cooksd > 4/(n-p), names(cooksd), ""), col =
```



The plot shows that one point is very far away from the rest of the data. The red line shows the cutoff of 4/(n-p), and we can see that observations 24 and 39 are potentially influential based on Cook's distance diagnostics.

3. $Y = (y_1, \dots, y_n)^T \in \mathbb{R}^n$, $x_1, \dots, x_n \in \mathbb{R}$

$y_i = f(x_i) + \epsilon_i$, $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$

(f is unknown but smooth)

$K(x)$ is a kernel that satisfies

- $K(x) \geq 0$
- $\underset{x \in \mathbb{R}}{\operatorname{argmax}} \; K(x) = 0$
- $\int K(x)\,dx = 1$
- for $h > 0$, $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$

(a) Define $\hat{f}^{(h)}$ as a local polynomial smoothing estimator for f, $d > 0$, kernel function K_h

Show that

$$\hat{Y} = \begin{bmatrix} \hat{f}^{(h)}(x_1) \\ \vdots \\ \hat{f}^{(h)}(x_n) \end{bmatrix} = L^{(h)} Y$$

for $L^{(h)} = [L_{ij}^{(h)}] \in \mathbb{R}^{n \times n}$. What are the rows of $L^{(h)}$ in terms of $x_1, \dots, x_n$, h, and K?

Based on the above definitions and assumptions, we also know that for WLS,

$\hat{\beta} = (X^T W X)^{-1} X^T W Y$, where $W = \operatorname{diag}(w_1, \dots, w_n)$

and we can define $\hat{f}^{(h)}(x)$ as:

$$\hat{f}^{(h)}(x) = \frac{\sum_{i=1}^n K_h(x_i - x) y_i}{\sum_{i=1}^n K_h(x_i - x)} = \sum_{i=1}^n w_i y_i$$

where w_i are the weights for smoothing.

If $\hat{Y} = \hat{f}^{(h)}(x) = \begin{bmatrix} \hat{f}^{(h)}(x_1) \\ \vdots \\ \hat{f}^{(h)}(x_n) \end{bmatrix}$, and $\hat{Y} = X\hat{\beta}$

$= X(X^T W X)^{-1} X^T W Y$

Then we can call $L^{(h)} = \begin{bmatrix} \frac{K_h(x_1 - x_1)}{\sum K_h(x_i - x_1)} & \cdots & \frac{K_h(x_n - x_1)}{\sum K_h(x_i - x_1)} \\ \vdots & \ddots & \vdots \\ \frac{K_h(x_1 - x_n)}{\sum K_h(x_i - x_n)} & & \frac{K_h(x_n - x_n)}{\sum K_h(x_i - x_n)} \end{bmatrix}$

the "smoother" of y_i, an $n \times n$ matrix.

The rows of $L^{(h)}$ are local smoothing weights in terms of observations of y, as shown above. ith rows are $K_h(x_i - x_i)$

3, continued

(b) $W_i^{(h)} = \text{diag}\{K_h(x_1 - x_i), \ldots, K_h(x_n - x_i)\}$ and

$$X_i = \begin{bmatrix} 1 & x_1 - x_i & \cdots & (x_1 - x_i)^d \\ \vdots & & & \vdots \\ 1 & x_n - x_i & \cdots & (x_n - x_i)^d \end{bmatrix}$$

$X_i^T W_i^{(h)} X_i$ is invertible for $i = 1, \ldots, n$

(i) Show that $0 < L_{ii}^{(h)} \leq 1$

we know that for $h > 0$, $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$

and $K(x) \geq 0$ (always positive), and max $K(x) = 0$

write $L_{ii}^{(h)} = L_i(x_i) = K(0) \Big/ \sum_j K\left(\frac{\| x_j - x_i \|}{h}\right)$

$K(0)$ is also always > 0, making all of $L_{ii}^{(h)} > 0$ (minimum 0, exclusive)

the denominator is maximized at $K(0)$, where $L_{ii}^{(h)} = \frac{K(0)}{K(0)} = 1$, the max of L_{ii},

so $0 < L_{ii}^{(h)} \leq 1$

(ii) Bandwidth h, $dfh = \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{cov}\{\hat{f}(x_i), y_i\}$. Find dfh, does it have to be an integer?

rewrite: $dfh = \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{cov}\{\hat{f}(x_i), y_i\} = \sum_{i=1}^{n} L_i(x_i)$

And since $0 < L_{ii}^{(h)} \leq 1$, $\sum_{i=1}^{n} L_i(x_i)$ is max at 1 in and min at 0 in making

$0 \leq dfh \leq n$,

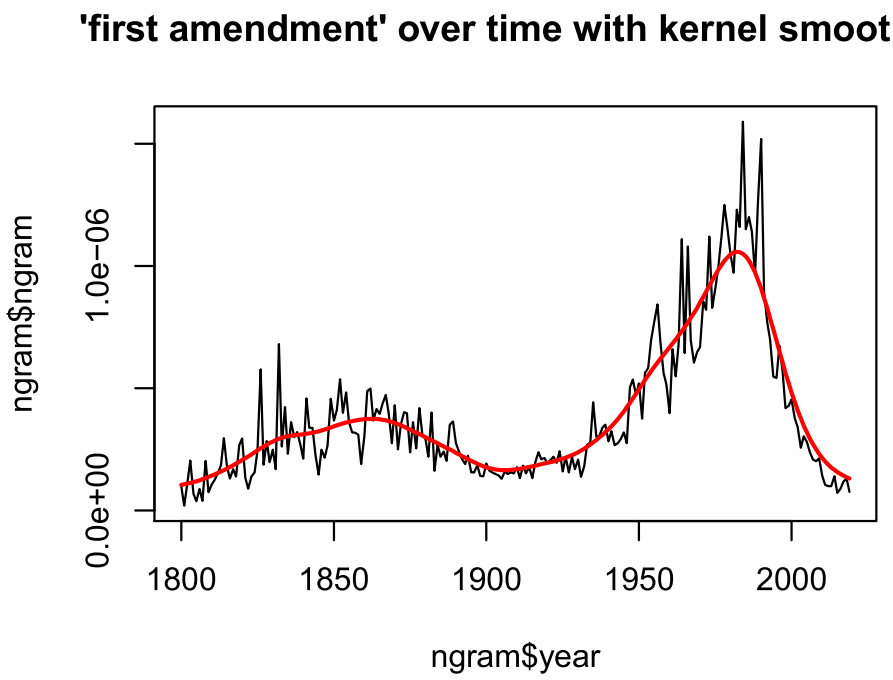
so dfh has to be a positive integer.

Problem 3 Part (c)

For this problem, choose your own word or phrase that you consider interesting, and fit a non parametric local linear regression to the word frequency over time.

Plot your estimated function on top of the raw data and report how you chose the bandwidth h, the effective degrees of freedom of your fitted function and if your fit was dependent on the choice of kernel K.

```
ngram = read.csv("ngram_data.csv")

dens = density(ngram$year, kernel = "gaussian")
bw = dens$bw

plot(ngram$year, ngram$ngram, type = "l",
     main = " 'first amendment' over time with kernel smoother ")
lines(ksmooth(ngram$year, ngram$ngram, "normal", bandwidth = bw), col = "red", lwd = 2)
```



Using the density() function, I found the gaussian kernel and subsequent bandwidth based on the year. The function ksmooth then takes in this bandwidth and graphs a fitted model on top, depending on this gaussian kernel.

The degrees of freedom is the trace of the hat matrix, in this case 1.

```
X = ngram$year
H = X %*% solve(t(X) %*% X) %*% t(X)
sum(diag(H))
```

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