R Exercise 4

Greg Ceccarelli

August 12, 2015

Part 1: OLS using a recursive function (optional)

Did not complete

Part 2: OLS using numerical optimization

In the case below, I've generated data that is heteroskedastic. When I compare the confidence intervals between regression estimates regular vs. bootstrapped, it would actually appear (per the output below) that they are very close. Additionally, the robust - corrected - standard errors are very similar although slightly higher.

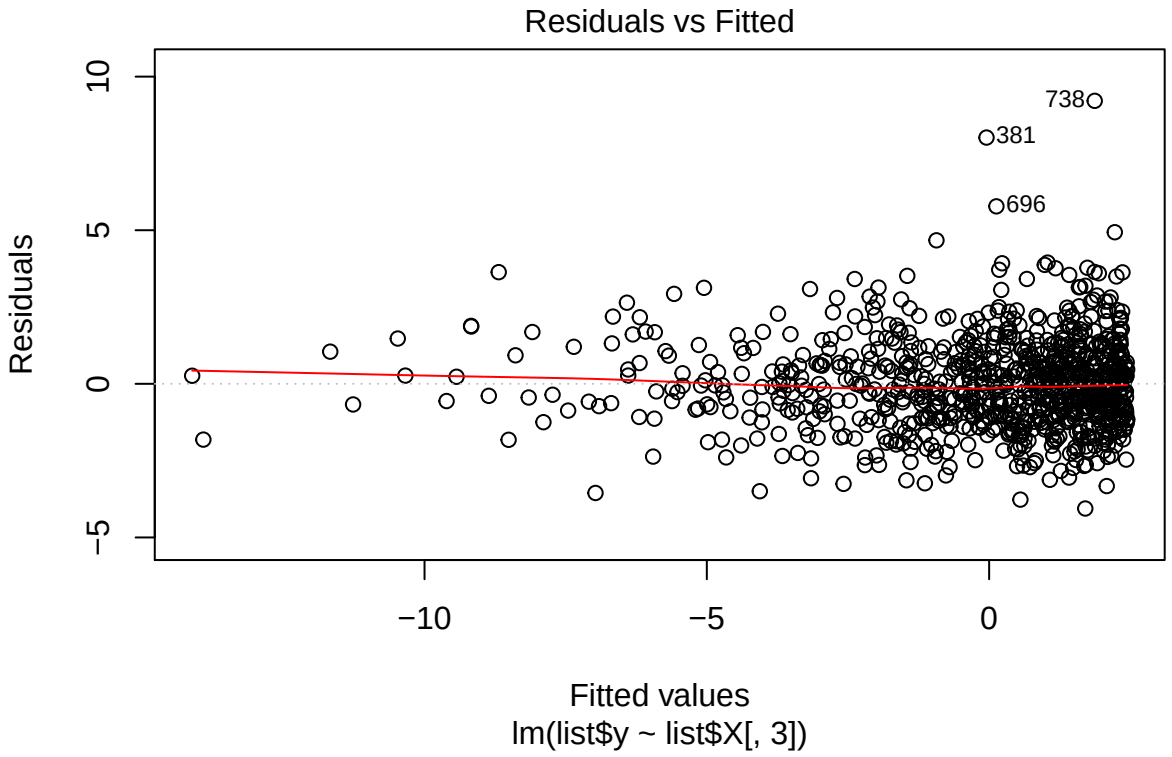
Thus this may suggest that we can relax the heteroskedasticity concern.

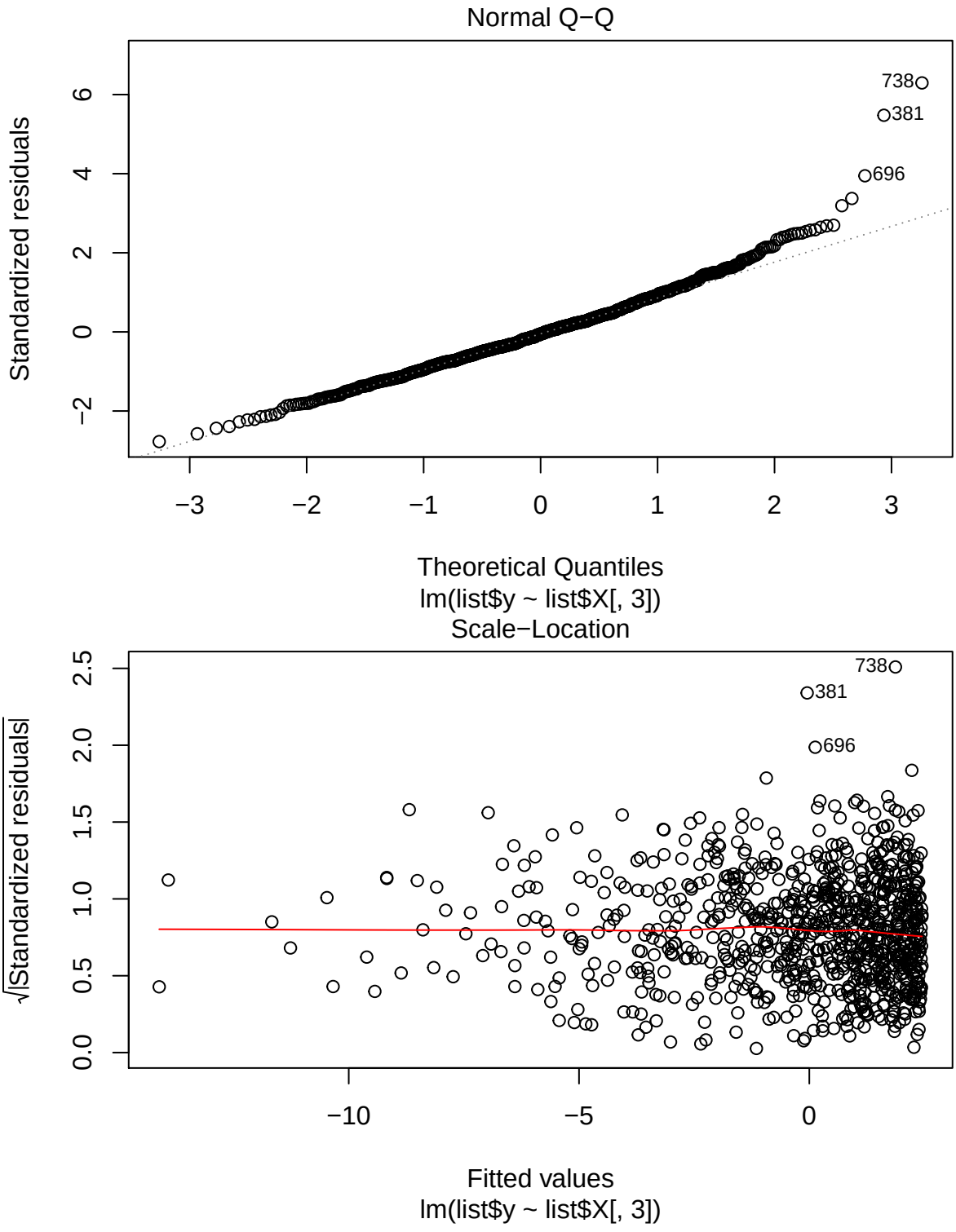
```
## Wrap Ali Gen data commands into a function to create
## data that violates homoskedasticity

gen_data <- function(N,ncol=3) {
   #sig <- matrix(mat, nrow=3)
   #M = mvrnorm(n = N, mu = rep(1,3), Sigma = sig, )

   ##########
   #modify function to generate heteroskedastic data
   M <- matrix(rgamma(N, shape=1, scale=1/2),ncol = ncol)
   y.cont = 1 + 2* M[,1] - 5 * M[,2] + M[,3] + rnorm(N/ncol) ##modified
   ############

   y.bin = as.numeric ( y.cont > 0 )
   X = cbind (1, M )
   y = y.cont
   #df <- data.frame(X,y)
   xy <- list("X"=X,"y"=y)
   return (xy)
}

##Run function and store results in list
list <- gen_data(N= 2700)

##regress y vs 2nd variable
reg <- lm(list$y ~ list$X[,3])

##regress y vs 1st variable
reg1 <- lm(list$y ~ list$X[,2])

#validate that homoskedasticity violated
summary(reg)
```

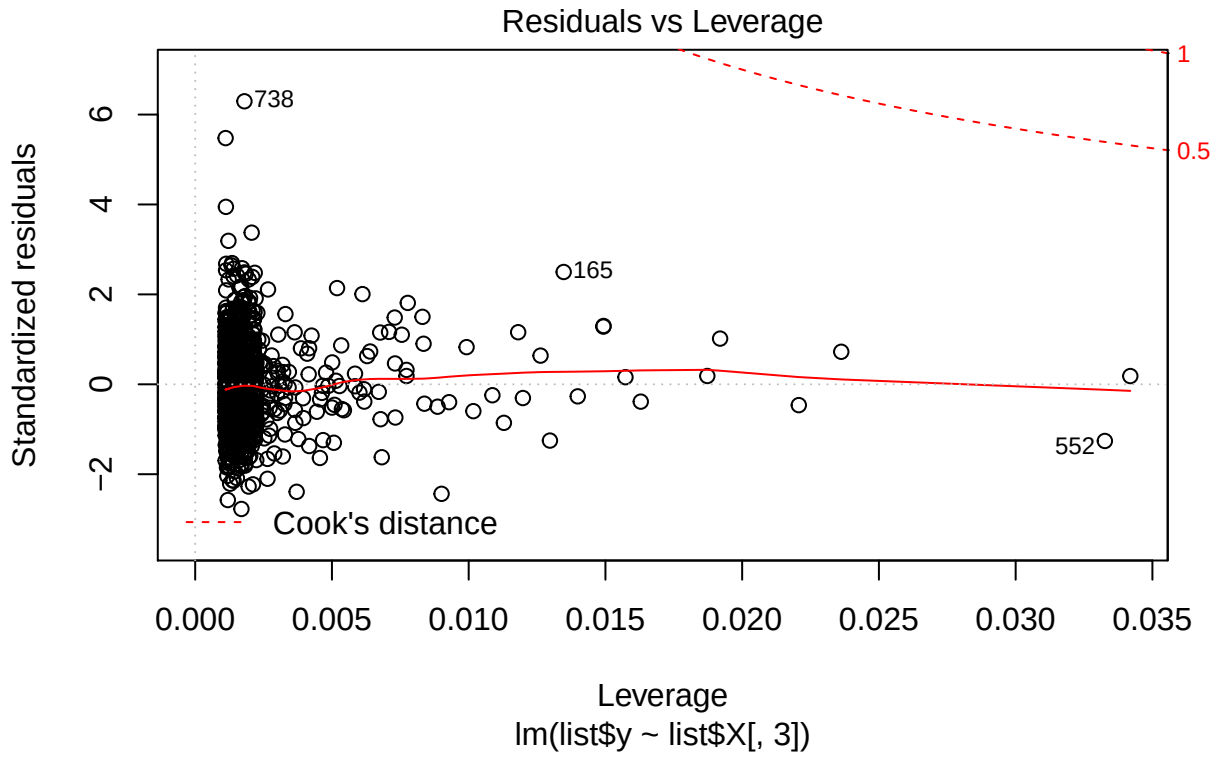
```
##
## Call:
## lm(formula = list$y ~ list$X[, 3])
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0582 -0.9650 -0.0803  0.8234  9.2113
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.45105    0.06955   35.24   <2e-16 ***
## list$X[, 3] -5.09838    0.09722  -52.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.465 on 898 degrees of freedom
## Multiple R-squared:  0.7538, Adjusted R-squared:  0.7536
## F-statistic:  2750 on 1 and 898 DF,  p-value: < 2.2e-16
```

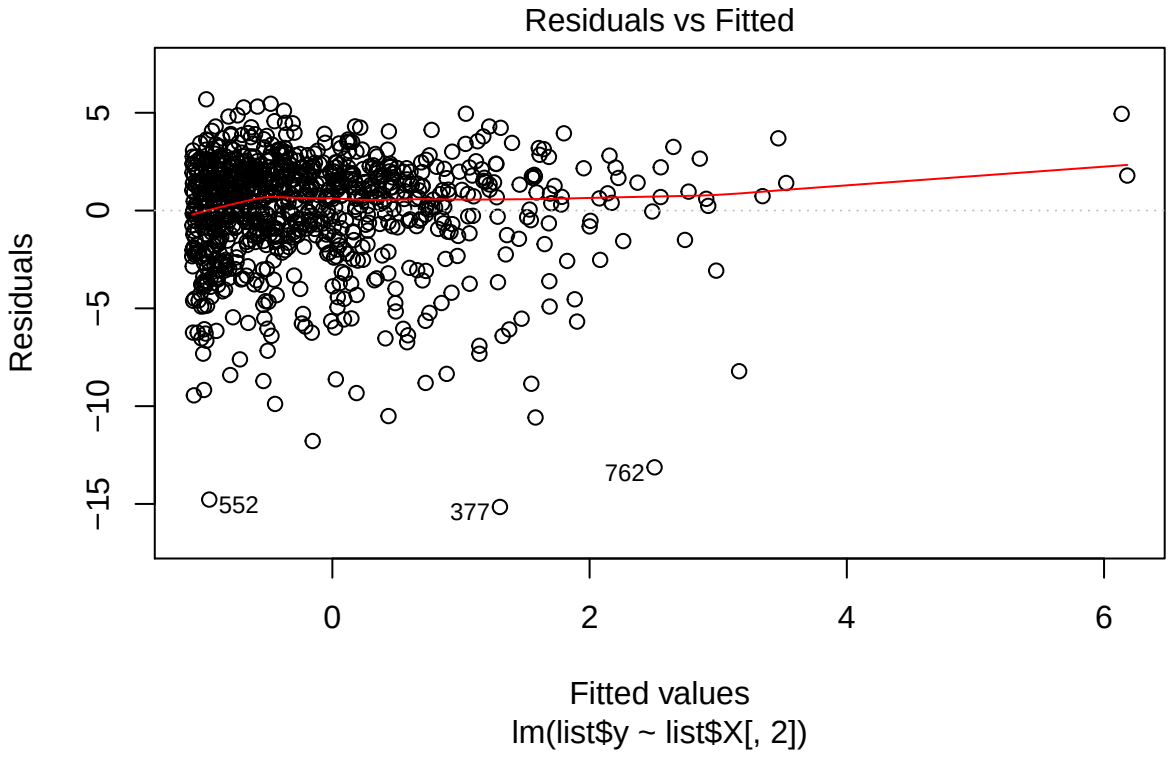
```
plot(reg)
```

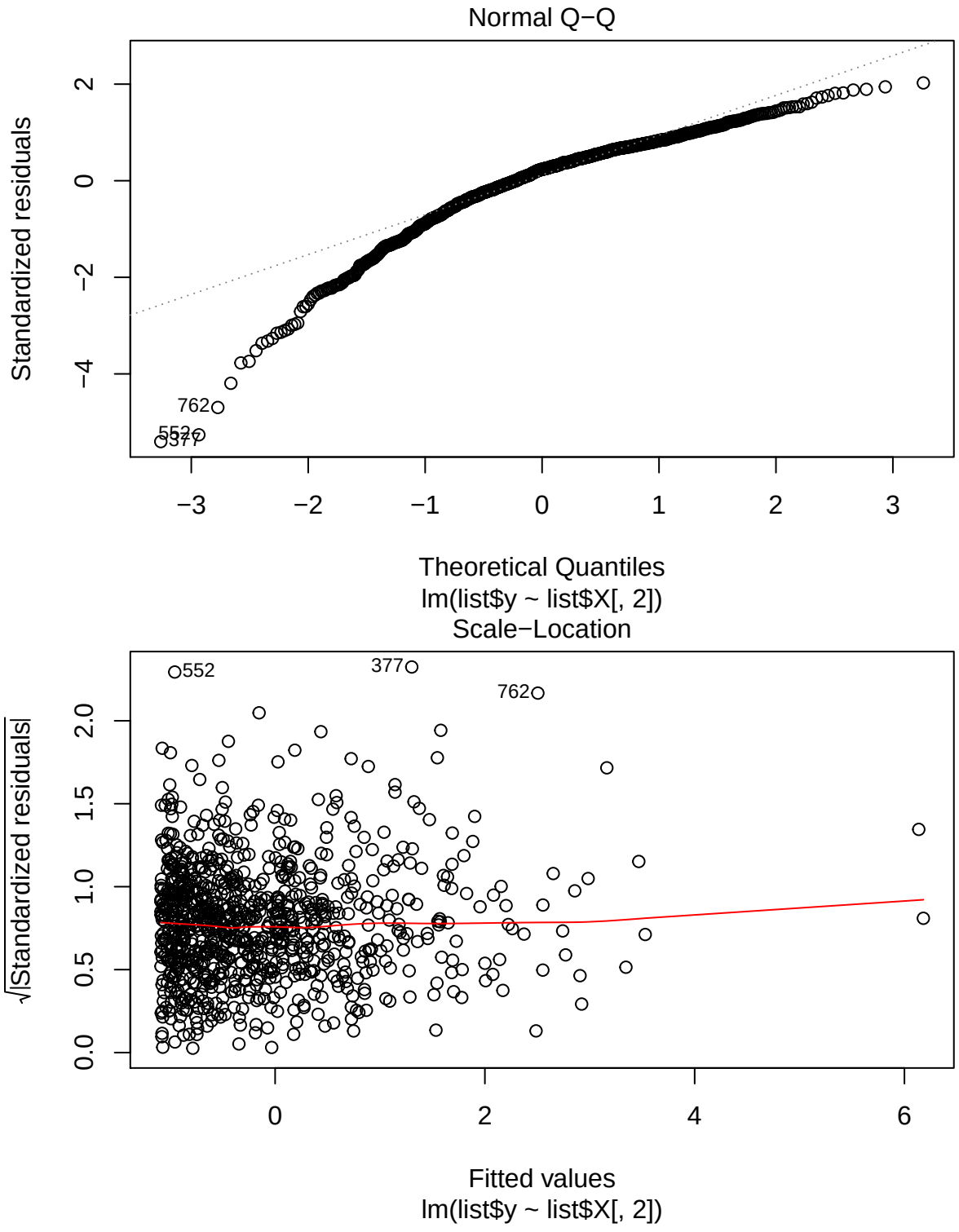



Normal Q−Q
Standardized residuals
738
381
696
−3
−2
−1
0
1
2
3
6
4
2
0
−2
Theoretical Quantiles
lm(list$y ~ list$X[, 3])
Scale−Location
√|Standardized residuals|
738
381
696
2.5
2.0
1.5
1.0
0.5
0.0
−10
−5
0
Fitted values
lm(list$y ~ list$X[, 3])


Residuals vs Leverage
Standardized residuals
Leverage
lm(list$y ~ list$X[, 3])
Cook's distance
1
0.5
738
165
552

```
plot(reg1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(list$y ~ list$X[, 2])
552
377
762




```
##plot(list$y, list$X[,3])
##plot(list$y, list$X[,2])


## Obtain initial multivariate regression
reg.initial <- lm(list$y ~ list$X[,3]+list$X[,2])
```

```r
##Obtain Confidence intervals for comparison
confint(reg.initial)
```

```
##                 2.5 %    97.5 %
## (Intercept)  1.341027  1.595594
## list$X[, 3] -5.285303 -4.996512
## list$X[, 2]  1.831647  2.131757
```

```r
#store as a dataframe
df <- data.frame(y = list$y, x = list$X[,3], x1 = list$X[,2])

library(boot)
#function to run bootstrapped
bootReg <- function (formula, data, i)
{
d <- data [i,]
fit <- lm(formula, data = d)
return(coef(fit))
}

bootResults<-boot(statistic = bootReg, formula = y ~ x+x1, data = df, R = 2000)

##compare bootstrap confidence intervals
boot.ci(bootResults, type = "bca", index = 2)
```

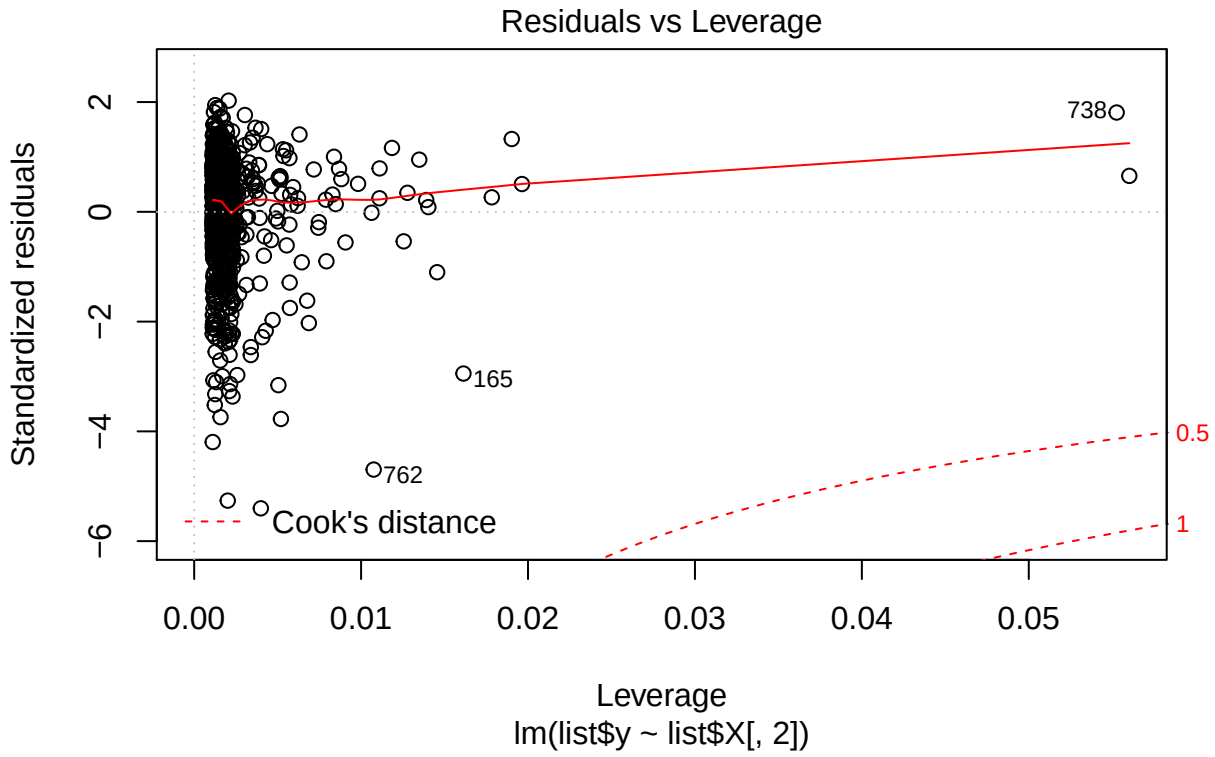
```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootResults, type = "bca", index = 2)
##
## Intervals :
## Level       BCa
## 95%   (-5.280, -4.989 )
## Calculations and Intervals on Original Scale
```

```r
boot.ci(bootResults, type = "bca", index = 3)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootResults, type = "bca", index = 3)
##
## Intervals :
## Level       BCa
## 95%   ( 1.831,  2.154 )
## Calculations and Intervals on Original Scale
```

```
#getting robust standard errors
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```




```
library(sandwich)

summary(reg.initial)
```

```
##
## Call:
## lm(formula = list$y ~ list$X[, 3] + list$X[, 2])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3617 -0.8070 -0.0750  0.7469  4.2908
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.46831    0.06485   22.64   <2e-16 ***
## list$X[, 3] -5.14091    0.07357  -69.88   <2e-16 ***
## list$X[, 2]  1.98170    0.07646   25.92   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.108 on 897 degrees of freedom
## Multiple R-squared:  0.8593, Adjusted R-squared:  0.8589
## F-statistic:  2738 on 2 and 897 DF,  p-value: < 2.2e-16
```

```
coeftest(reg.initial,  vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  1.468311   0.066826  21.972 < 2.2e-16 ***
## list$X[, 3] -5.140908   0.074715 -68.807 < 2.2e-16 ***
## list$X[, 2]  1.981702   0.086754  22.843 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#use waldtest to get F stat
waldtest(reg.initial, vcov = vcovHC)
```

```
## Wald test
##
## Model 1: list$y ~ list$X[, 3] + list$X[, 2]
## Model 2: list$y ~ 1
##   Res.Df Df      F    Pr(>F)
## 1    897
## 2    899 -2 2502.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Part 3: Logit

Did not complete