CW1_13128128_N_Katz.rmd

NK

18 November 2018

1. Statistical learning methods

a) inflexible method is better as there are enough data points and not many predictors and so the formula to describe f is simple as there aren't many coafficiants.

b) flexible learning method is better as there are too many predictors to account for

c) flexible learning method is better as it's not a simple relationship i.e. has many coafficiants, many streight lines and so trying to apply a inflexible learning method might fit the traning data well but be not useful for prediction.

d) a felxible method as the high variance in the epsilons tells you that this is probably a complax relations and f is highly non-linar.

2. Descriptive analysis

```
OralExamResults<-c(4, 1, 4, 5, 3, 2, 3, 4, 3, 5, 2, 2, 4, 3, 5, 5, 1, 1, 1, 2
)
WrittenExamResults<-c(2, 3, 1, 4, 2, 5, 3, 1, 2, 1, 2, 2, 1, 1, 2, 3, 1, 2, 3
, 4)
ExamResults<-c(OralExamResults,WrittenExamResults)
mean(OralExamResults)

## [1] 3

median(OralExamResults)

## [1] 3

getMode<-function(x){
  uniqueResults<-unique(x)
#broken up it is
#uniquex - only unique numbers in vector
#match(x, uniquex) - - how many times are each of the unique found in the vec
tor of non-unique
#tabulate(match(x, uniquex)) - place # times corresponding to the values in u
nique, into another vector
#which.max(tabulate(match(x, uniquex))) - comparing unique with tabulate, fin
d the unieqe that corresponds to the most in tabulate and that's the mode fro
m your non-unique vector.
uniqueResults[which.max(tabulate(match(x, uniqueResults)))]
}#taken from here https://stackoverflow.com/questions/2547402/is-there-a-buil
t-in-function-for-finding-the-mode
```

```
summaryExamResults <- c(mean(OralExamResults),median(OralExamResults),getMode
(OralExamResults),var(OralExamResults),sd(OralExamResults),mean(WrittenExamRe
sults),median(WrittenExamResults),getMode(WrittenExamResults),var(WrittenExam
Results),sd(WrittenExamResults), mean(ExamResults),median(ExamResults),getMod
e(ExamResults),var(ExamResults),sd(ExamResults))

  matrixSummaryExamResults<-matrix(seq(summaryExamResults),3,5,byrow=T,
dimnames = list(
c("OralExamResults", "WrittenExamResults", "OralAndWrittenExamResults"),
c("mean", "median", "mode", "variance", "standardDeviation")
))
matrixSummaryExamResults

##                            mean median mode variance standardDeviation
## OralExamResults               1      2    3        4                 5
## WrittenExamResults            6      7    8        9                10
## OralAndWrittenExamResults    11     12   13       14                15

cor(OralExamResults,WrittenExamResults)

## [1] -0.1869531

cov(OralExamResults,WrittenExamResults)

## [1] -0.3157895
```

c)a weak negative correlation between the two as it's quite close to 0. Negative correlation, meaning that getting a low mark at 1 exam would indicate getting a better results at the other. Hence the covariance is negative, showing the above avarage of one variables are related to the lower than avarage of the other and vice versa.

d)This doesn't imply causation, especially when doing well in one should naturally mean they can also do well in the other. There maybe other variables at play, which cause the relation or the relation observed maybe rendom. In order to establish causation, ontrolled two group studies would need to be done.

3 Linear regression a)

```
library(ISLR)
attach(Auto)
lm.fit<-lm(mpg~horsepower, data=Auto)
summary(lm.fit)

##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
```

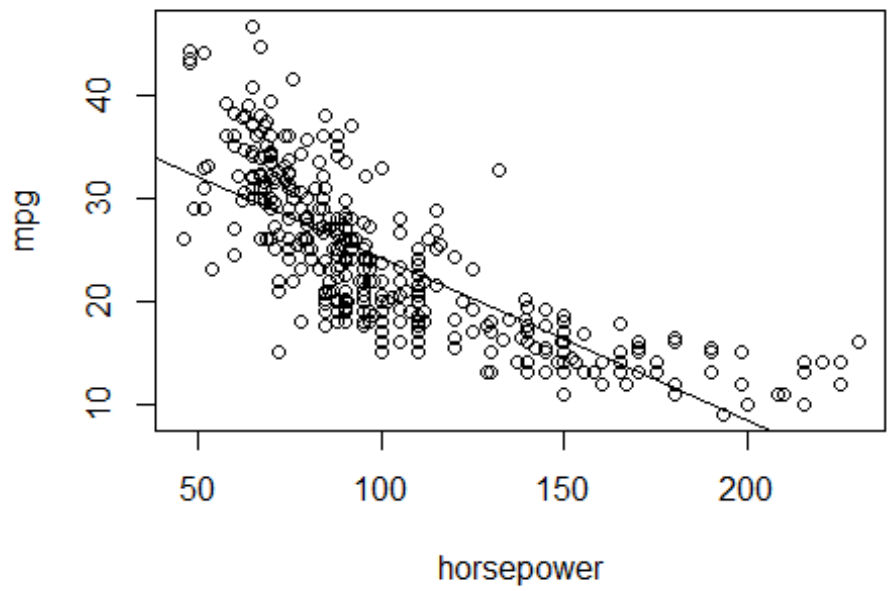
```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

I) t-value is large and p-value is small. We can conclude that B1!=0 and there is a significant a negative relationship between horsepower and miles per gallon; and that for each increase in horespower unit there is a decrease of 0.16 in miles used up per gallon. In this case the intercept is ignored as if the horsepower is 0, there are no mpg being used.

II) R squared after adjusting indicates that 60.5 percent of the variation we see in mpg is explained by the difference between engine horsepower.

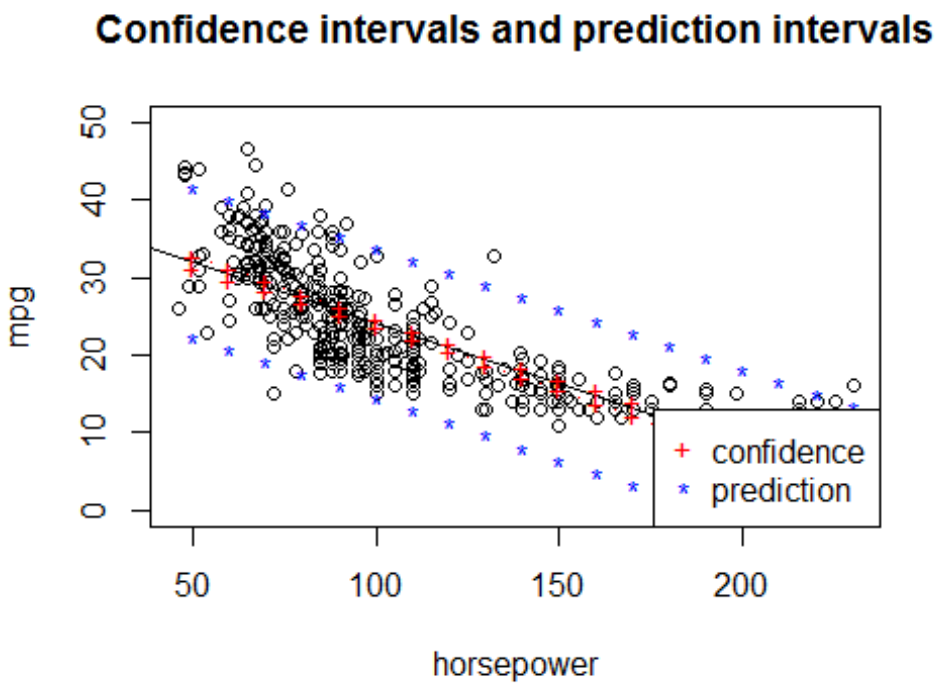
III) The predicted mpg associated with a horsepower of 98 would be 24.5. We estimate (with 95% confidence) that our prediction for 98, would fall not less than 23.97 and not above 24.96 mpg (B1).

```
predict(lm.fit,
data.frame(horsepower=98),
interval="confidence")

##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108

plot(mpg~horsepower, data=Auto)
abline(lm.fit)
```



c)

```
plot(horsepower,mpg,
xlab="horsepower", ylab = "mpg",
main = "Confidence intervals and prediction intervals",
ylim=c(0,50)
)
abline(lm.fit)
newHp <- data.frame(horsepower=seq(50,250,by=10))
p_conf <- predict(lm.fit,newHp,interval="confidence")
p_pred <- predict(lm.fit,newHp,interval="prediction")
lines(newHp$horsepower,p_conf[,"upr"],col="red", type="b",pch="+")
lines(newHp$horsepower,p_conf[,"lwr"],col="red", type="b",pch="+")
lines(newHp$horsepower,p_pred[,"upr"],col="blue", type="b",pch="*")
lines(newHp$horsepower,p_pred[,"lwr"],col="blue",type="b",pch="*")
legend("bottomright",
pch=c("+","*"),
col=c("red","blue"),
legend = c("confidence","prediction"))
```



5)

```
library(MASS)
attach(Boston)
View(Boston)
?Boston

## starting httpd help server ... done

crim.1<-rep(0,506)
y<-which(Boston$crim>median(Boston$crim))
crim.1[y]<-1
a<-glm(crim.1~nox,data=Boston,family="binomial")
summary(a)

##
## Call:
## glm(formula = crim.1 ~ nox, family = "binomial", data = Boston)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.27324  -0.37245  -0.06847   0.39620   2.53124
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -15.818      1.386  -11.41   <2e-16 ***
## nox           29.365      2.599   11.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 320.39  on 504  degrees of freedom
## AIC: 324.39
##
## Number of Fisher Scoring iterations: 6
```

```
a<-glm(crim.1~rad, data=Boston,family="binomial")
summary(a)
```

```
##
## Call:
## glm(formula = crim.1 ~ rad, family = "binomial", data = Boston)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.41951  -0.80288  -0.21392   0.05129   1.94445
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.48670    0.33063  -7.521 5.43e-14 ***
## rad          0.37998    0.06484   5.861 4.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 445.60  on 504  degrees of freedom
## AIC: 449.6
##
## Number of Fisher Scoring iterations: 7
```

```
a<-glm(crim.1~ptratio,data=Boston,family="binomial")
summary(a)
```

```
##
## Call:
## glm(formula = crim.1 ~ ptratio, family = "binomial", data = Boston)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5637  -1.1256   0.1319   1.0030   1.7865
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.60893    0.84072  -5.482 4.20e-08 ***
## ptratio      0.24921    0.04502   5.536 3.09e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 667.94  on 504  degrees of freedom
## AIC: 671.94
##
## Number of Fisher Scoring iterations: 4
```

```
a<-glm(crim.1~medv,data=Boston,family="binomial")
summary(a)
```

```
##
## Call:
## glm(formula = crim.1 ~ medv, family = "binomial", data = Boston)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.61138  -1.15556   0.09793   1.05984   1.96521
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.42773    0.26642   5.359 8.37e-08 ***
## medv        -0.06404    0.01141  -5.612 2.00e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 664.48  on 504  degrees of freedom
## AIC: 668.48
##
## Number of Fisher Scoring iterations: 4
```

```
a<-glm(crim.1~tax,data=Boston,family="binomial")
summary(a)
```

```
##
## Call:
## glm(formula = crim.1 ~ tax, family = "binomial", data = Boston)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.65467  -0.77580  -0.08339   0.30994   2.10786
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.119929   0.375267  -10.98   <2e-16 ***
## tax          0.010708   0.001019   10.51   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 480.54  on 504  degrees of freedom
## AIC: 484.54
##
## Number of Fisher Scoring iterations: 5
```

```
a<-glm(crim.1~zn,data=Boston,family="binomial")
summary(a)
```

```
##
## Call:
## glm(formula = crim.1 ~ zn, family = "binomial", data = Boston)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4319  -1.4319   0.4634   0.9427   1.8532
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.58076    0.10727   5.414 6.17e-08 ***
## zn          -0.09545    0.01349  -7.075 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 559.53  on 504  degrees of freedom
## AIC: 563.53
##
## Number of Fisher Scoring iterations: 6
```

```
a<-glm(crim.1~nox+rad+ptratio+medv, data=Boston,family="binomial")
summary(a)
```

```
##
## Call:
## glm(formula = crim.1 ~ nox + rad + ptratio + medv, family = "binomial",
##     data = Boston)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.07406  -0.30380  -0.01456   0.00718   2.68374
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -26.03154    3.55628  -7.320 2.48e-13 ***
## nox          30.87955    3.57504   8.638  < 2e-16 ***
## rad           0.51220    0.10651   4.809 1.52e-06 ***
## ptratio       0.27281    0.09517   2.867  0.00415 **
## medv          0.07199    0.02606   2.763  0.00573 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 254.20  on 501  degrees of freedom
## AIC: 264.2
##
## Number of Fisher Scoring iterations: 8
```

These variables all seem to suggest a strong relationship with crime individually (nox being the most significant), with medv and zn implying a negative relationship, but combined, it seems that a high nitrogen oxide concentration, low accessibility to highwasy, teacher to pupil ration and median value of owner occupide homes in $1000s, together influence the most, a positive relation with crime, as it appears from the samll p value.

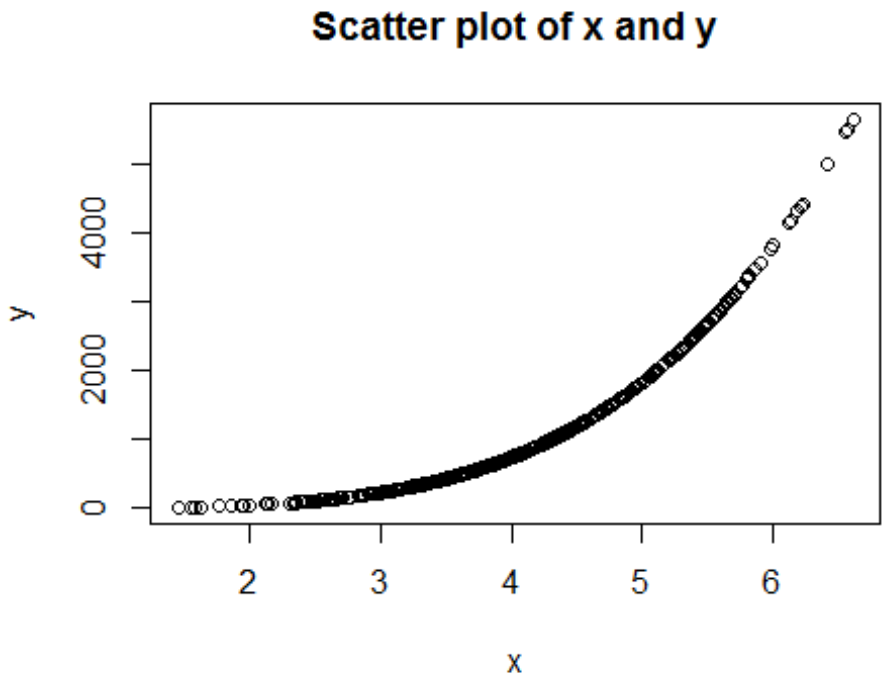
Interestingly, black on its own shows a slight negative relation which really is positive when realising that the the proportions in this data set are the results of the equation 1000(Bk - 0.63)^2 and so rearanging the formula means that where the proportion of black is now the heighest, would after rearnagment come out as 0 in the variable Bk.. However, having said that, when combining black with other significant variables, black does not matter much, suggesting that the reason black shows a relationship on its own is maybe due to that being intself corralated to the other variables that do imply a strong relationship with crime.

7) a), b), c) & d)

```
library(boot)
set.seed(500)
y=rnorm(500)
x=4-rnorm(500)
y=x -(2*(x^2))+(3*(x^4))+rnorm(500)
length(y)
```

```
## [1] 500
```

```
plot(y~x, main = "Scatter plot of x and y", xlab = "x", ylab= "y")
```




```
set.seed(23)
x_2<-x^2
x_3<-x^3
x_4<-x^4

A<-data.frame(y,x,x_2,x_3,x_4)
i<-glm(y~x)
A<-data.frame(y,x)
cv.erri <- cv.glm(A,i, K=10)
cv.erri$delta

## [1] 155466.2 155338.7

ii<-glm(y~x+x_2)
B<-data.frame(y,x,x_2)
cv.errii <- cv.glm(B,ii, K=10)
cv.errii$delta

## [1] 7451.005 7424.222

iii<-glm(y~x+x_2+x_3)
C<-data.frame(y,x,x_2,x_3)
cv.erriii <- cv.glm(C,iii, K=10)
cv.erriii$delta

## [1] 61.00506 60.75021
```

```
iv<-glm(y~x+x_2+x_3+x_4)
D<-data.frame(y,x,x_2,x_3,x_4)
cv.erriv <- cv.glm(D,iv, K=10)
cv.erriv$delta

## [1] 0.9218342 0.9207294

set.seed(46)
x_2<-x^2
x_3<-x^3
x_4<-x^4

A<-data.frame(y,x,x_2,x_3,x_4)
i<-glm(y~x)
A<-data.frame(y,x)
cv.erri <- cv.glm(A,i, K=10)
cv.erri$delta

## [1] 155791.8 155647.0

ii<-glm(y~x+x_2)
B<-data.frame(y,x,x_2)
cv.errii <- cv.glm(B,ii, K=10)
cv.errii$delta

## [1] 7348.591 7327.531

iii<-glm(y~x+x_2+x_3)
C<-data.frame(y,x,x_2,x_3)
cv.erriii <- cv.glm(C,iii, K=10)
cv.erriii$delta

## [1] 62.17963 61.85851

iv<-glm(y~x+x_2+x_3+x_4)
D<-data.frame(y,x,x_2,x_3,x_4)
cv.erriv <- cv.glm(D,iv, K=10)
cv.erriv$delta

## [1] 0.9305837 0.9289982
```

Comment to a): n is 500 and x is p.

Comment to b): The scatter plot demonstartes an exponential relationship between x and y.

Comment to d) The cv isn't the same because seed was changed.

Comment to e) Ex 5 is the smallest; yes it was expected becasue the random data was moduled based on 4th power component and so incorporating x^4 provides a much better fit for this data.

```
summary(i)
```

```
##
## Call:
## glm(formula = y ~ x)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -306.3  -277.2  -156.1   143.6  2253.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2629.02      74.50  -35.29   <2e-16 ***
## x             913.03      17.87   51.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 153659.7)
##
##     Null deviance: 477702712  on 499  degrees of freedom
## Residual deviance:  76522553  on 498  degrees of freedom
## AIC: 7394.2
##
## Number of Fisher Scoring iterations: 2

cv.erri$delta

## [1] 155791.8 155647.0

summary(ii)

##
## Call:
## glm(formula = y ~ x + x_2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -475.10   -52.53     2.45    51.63   547.66
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2122.620     49.131   43.20   <2e-16 ***
## x           -1573.227     24.622  -63.89   <2e-16 ***
## x_2           306.231      2.996  102.21   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6992.535)
##
##     Null deviance: 477702712  on 499  degrees of freedom
## Residual deviance:   3475290  on 497  degrees of freedom
## AIC: 5850.2
```

```
##
## Number of Fisher Scoring iterations: 2
```

```
cv.errii$delta
```

```
## [1] 7348.591 7327.531
```

```
summary(iii)
```

```
##
## Call:
## glm(formula = y ~ x + x_2 + x_3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -11.839   -5.639    1.009    4.245   52.241
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -588.4433    11.8523  -49.65   <2e-16 ***
## x            686.3238     9.4287   72.79   <2e-16 ***
## x_2         -283.0150     2.4050 -117.68   <2e-16 ***
## x_3           48.5304     0.1968  246.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 56.7041)
##
##     Null deviance: 477702712  on 499  degrees of freedom
## Residual deviance:     28125  on 496  degrees of freedom
## AIC: 3443.8
##
## Number of Fisher Scoring iterations: 2
```

```
cv.erriii$delta
```

```
## [1] 62.17963 61.85851
```

```
summary(iv)
```

```
##
## Call:
## glm(formula = y ~ x + x_2 + x_3 + x_4)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.88395  -0.61614   0.02642   0.64776   2.70284
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.47931    3.70732   0.669    0.504
## x           -1.80641    4.12392  -0.438    0.662
```

```
## x_2         -0.81958    1.64709  -0.498    0.619
## x_3         -0.21421    0.28070  -0.763    0.446
## x_4          3.01385    0.01729 174.340   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9105089)
##
##     Null deviance: 4.777e+08  on 499  degrees of freedom
## Residual deviance: 4.507e+02  on 495  degrees of freedom
## AIC: 1379
##
## Number of Fisher Scoring iterations: 2

cv.erriv$delta

## [1] 0.9305837 0.9289982
```