week5 exercies

Zaiwei

Aug,25,2019

Question 1: Cross-validation

a) Upload the Auto dataset and explore it.

```
Auto <- read.csv('Auto.csv')
attach(Auto)
dim(Auto)
## [1] 397   9
names(Auto)
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "year"         "origin"
## [9] "name"
sapply(Auto,class)
##          mpg    cylinders displacement   horsepower       weight
##    "numeric"    "integer"    "numeric"     "factor"    "integer"
## acceleration         year       origin         name
##    "numeric"    "integer"    "integer"     "factor"
```

b) Fit a polynomial regression model for mpg and horsepower. (Recap of Week 3)

```
horsepower.numc <- as.numeric(Auto$horsepower)
ml.poly <- lm(mpg~poly(horsepower.numc,2), data=Auto)
summary(ml.poly)
##
## Call:
## lm(formula = mpg ~ poly(horsepower.numc, 2), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.9907  -6.0269  -0.2335   4.7160  23.8816
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                23.5159     0.3559  66.067   <2e-16 ***
## poly(horsepower.numc, 2)1  65.8468     7.0920   9.285   <2e-16 ***
## poly(horsepower.numc, 2)2  -9.9834     7.0920  -1.408     0.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.092 on 394 degrees of freedom
## Multiple R-squared:  0.1829, Adjusted R-squared:  0.1787
## F-statistic: 44.09 on 2 and 394 DF,  p-value: < 2.2e-16
ml.I  <- lm(mpg~horsepower.numc+I(horsepower.numc^2),data=Auto)
summary(ml.I)
##
## Call:
## lm(formula = mpg ~ horsepower.numc + I(horsepower.numc^2), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.9907  -6.0269  -0.2335   4.7160  23.8816
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          16.8389456  0.9890343  17.026  < 2e-16 ***
## horsepower.numc       0.1801992  0.0507205   3.553 0.000427 ***
## I(horsepower.numc^2) -0.0007355  0.0005225  -1.408 0.160009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.092 on 394 degrees of freedom
## Multiple R-squared:  0.1829, Adjusted R-squared:  0.1787
```

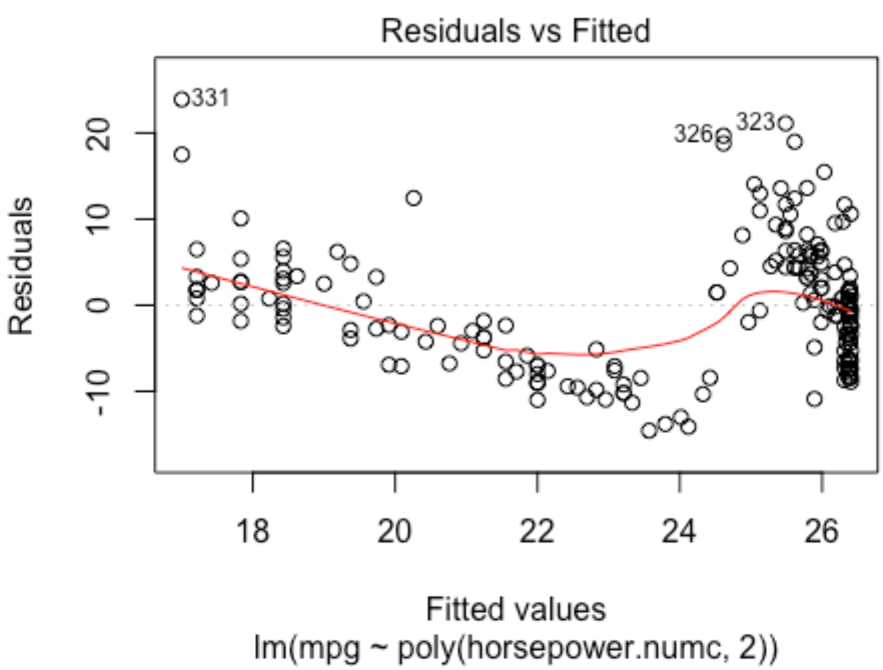
```
## F-statistic: 44.09 on 2 and 394 DF,  p-value: < 2.2e-16
```

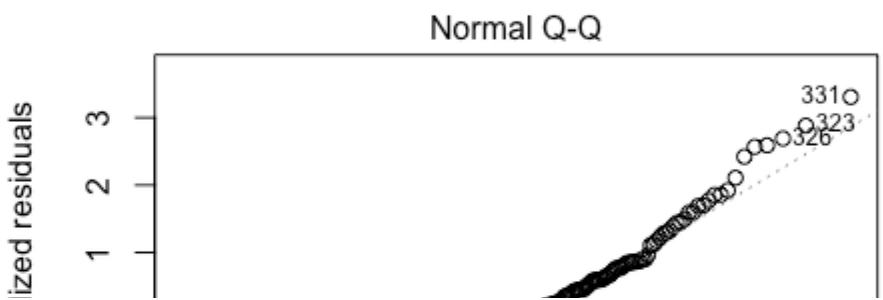
c) Use the validation set approach to select the best model.

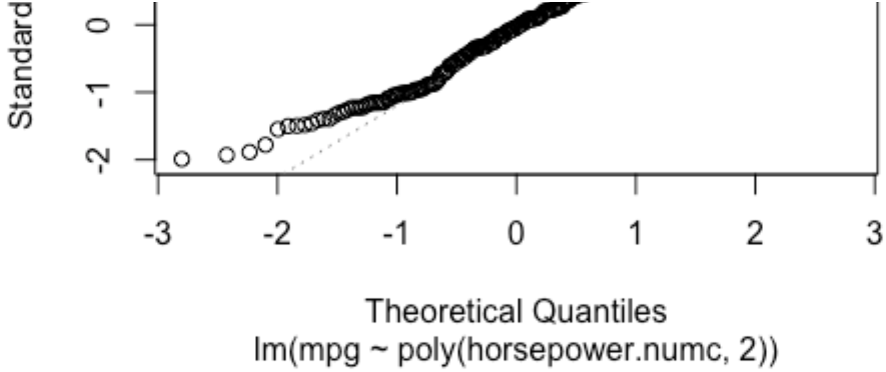
```
set.seed(1)
tr = sample(1:392,196)
train=Auto[tr , ]
dim(train)
## [1] 196   9
model.tr = lm(mpg~poly(horsepower.numc, 2), data=Auto, subset = tr)
coef(model.tr)
##              (Intercept) poly(horsepower.numc, 2)1
##                 23.96553                  58.84296
## poly(horsepower.numc, 2)2
##                -10.29023
mean((mpg -predict (model.tr,Auto))[-tr ]^2)
## [1] 46.07677
```

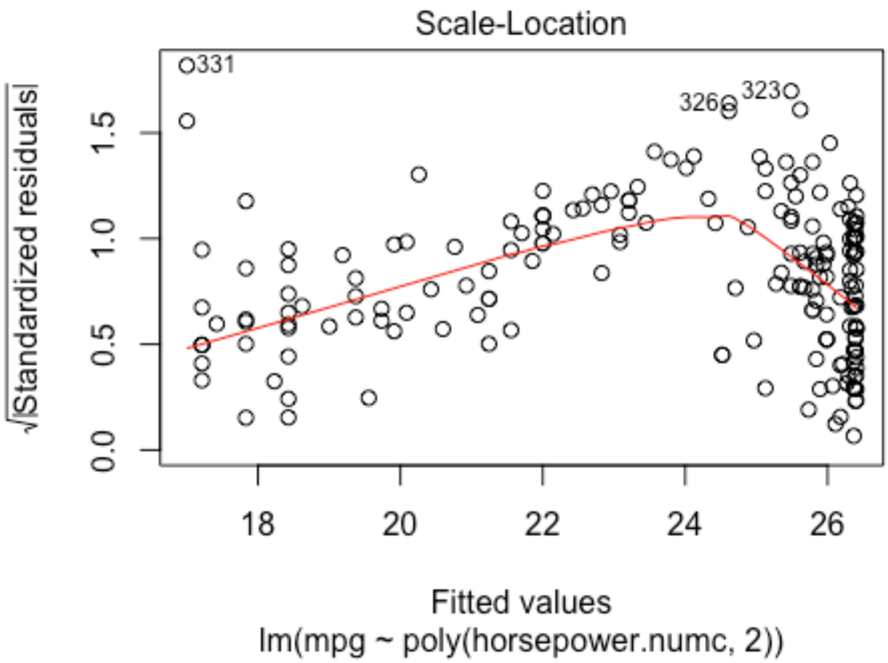
d) Repeat part (c) with set.seed values as 5 and 8. Compare your results.

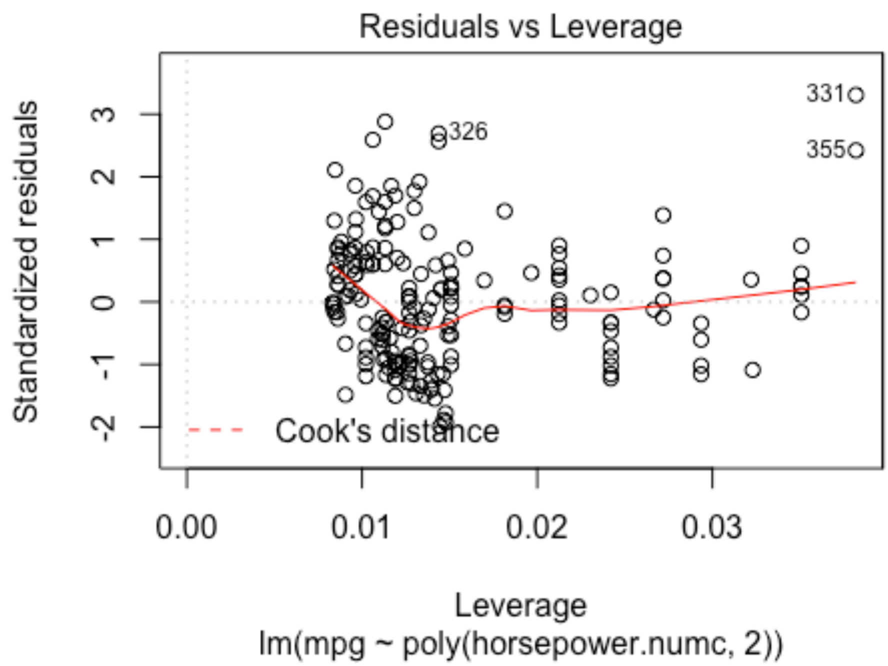
```
set.seed(5)
tr = sample(5:392,196)
train=Auto[tr , ]
dim(train)
## [1] 196   9
model.tr = lm(mpg~poly(horsepower.numc, 2), data=Auto, subset = tr)
set.seed(8)
tr = sample(8:392,196)
train=Auto[tr , ]
dim(train)
## [1] 196   9
model.tr = lm(mpg~poly(horsepower.numc, 2), data=Auto, subset = tr)
plot(model.tr)
```











#

e) What is the drawback in using validation set approach to select the best model?

No randomness of using some observations for training vs. validation set like in validation-set method as each observation is considered for both training and validation

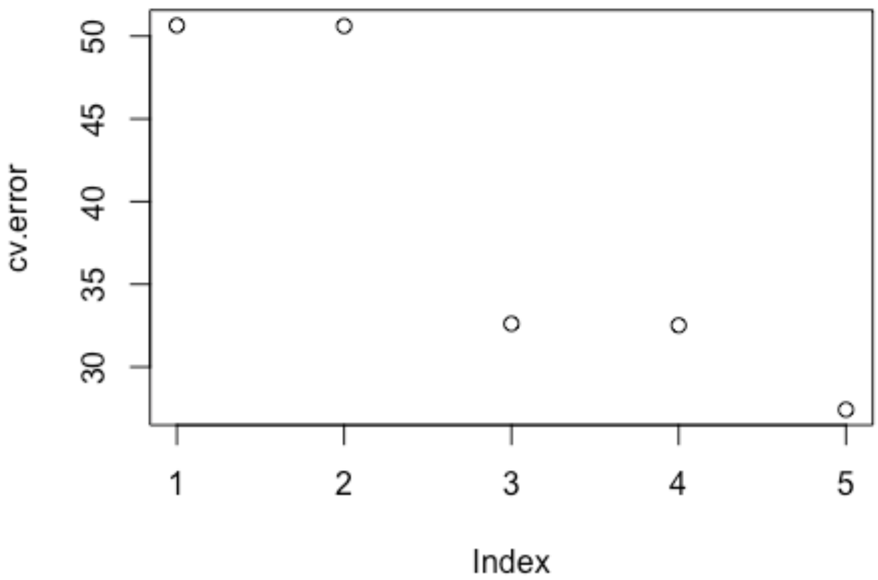
f) Use the LOOCV method for the above model.

```
model3 = glm (mpg~poly(horsepower.numc,2),data=Auto)
coef(model3)
##          (Intercept) poly(horsepower.numc, 2)1
##             23.51587                 65.84683
## poly(horsepower.numc, 2)2
```

```
## poly(horsepower.numc, 2)2
##               -9.98339
```

g) Use k-fold cross validation method for the same model.

```
library(boot)
set.seed(10)
cv.error=rep(0,5)
for(i in 1:5){
 glm.fit = glm(mpg~poly(as.numeric(horsepower),i),data=Auto)
 cv.error[i]=cv.glm(Auto,glm.fit,K=10)$delta[1]
}
cv.error
## [1] 50.64546 50.61590 32.61239 32.51488 27.41722
plot(cv.error)
```



#

h) Compare your results and interpret your findings

```
summary(model3)
##
## Call:
## glm(formula = mpg ~ poly(horsepower.numc, 2), data = Auto)
##
## Deviance Residuals:
##      Min        1Q    Median       3Q       Max
## -13.9907   -6.0269   -0.2335    4.7160   23.8816
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 23.5159     0.3559  66.067   <2e-16 ***
## poly(horsepower.numc, 2)1  65.8468     7.0920   9.285   <2e-16 ***
## poly(horsepower.numc, 2)2  -9.9834     7.0920  -1.408     0.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 50.29654)
##
##     Null deviance: 24252  on 396  degrees of freedom
## Residual deviance: 19817  on 394  degrees of freedom
## AIC: 2687
##
## Number of Fisher Scoring iterations: 2
summary(model.tr)
##
## Call:
## lm(formula = mpg ~ poly(horsepower.numc, 2), data = Auto, subset = tr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5689  -6.2176  -0.2857   4.3398  23.8978
```

```
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             23.5993     0.5266  44.811  < 2e-16 ***
## poly(horsepower.numc, 2)1  60.4225    10.4805   5.765 3.19e-08 ***
## poly(horsepower.numc, 2)2 -16.2034    10.6500  -1.521     0.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.369 on 193 degrees of freedom
## Multiple R-squared:  0.1532, Adjusted R-squared:  0.1444
## F-statistic: 17.45 on 2 and 193 DF,  p-value: 1.079e-07
```

Question 2: Bootstrapping

a) Generate 20 random numbers using the following R code:

```
x <- 10*rexp(20)
```

b) Calculate the mean of x

```
mean(x)
## [1] 8.147539
```

c) Generate 1000 bootstrap samples using the above dataset.

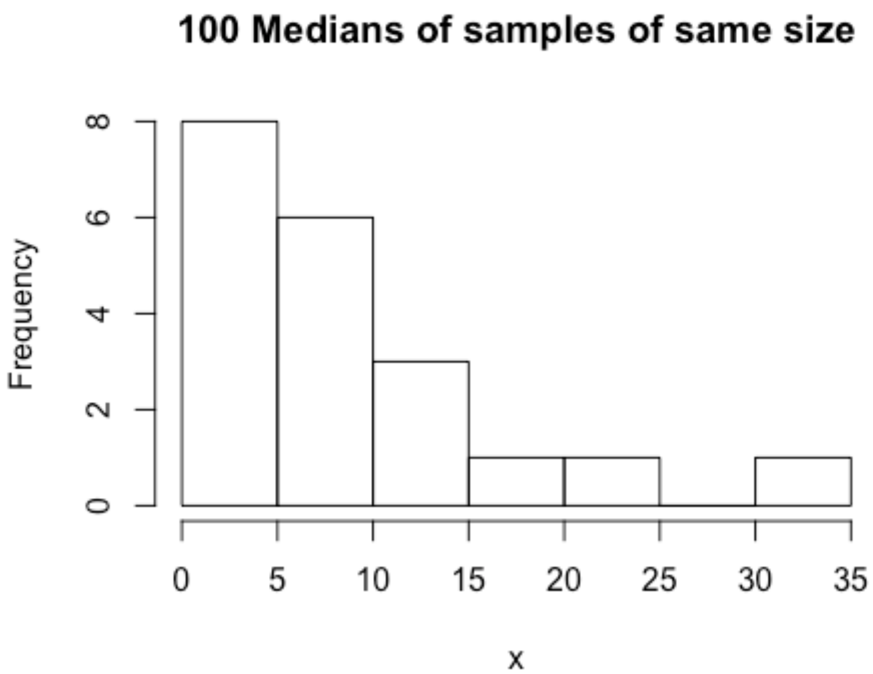
```
x.1000 <- 10*rexp(1000)
```

d) Repeat part (b) for all the 1000 samples.

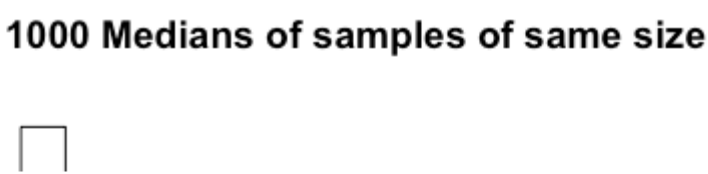
```
mean(x.1000)
## [1] 9.961941
```

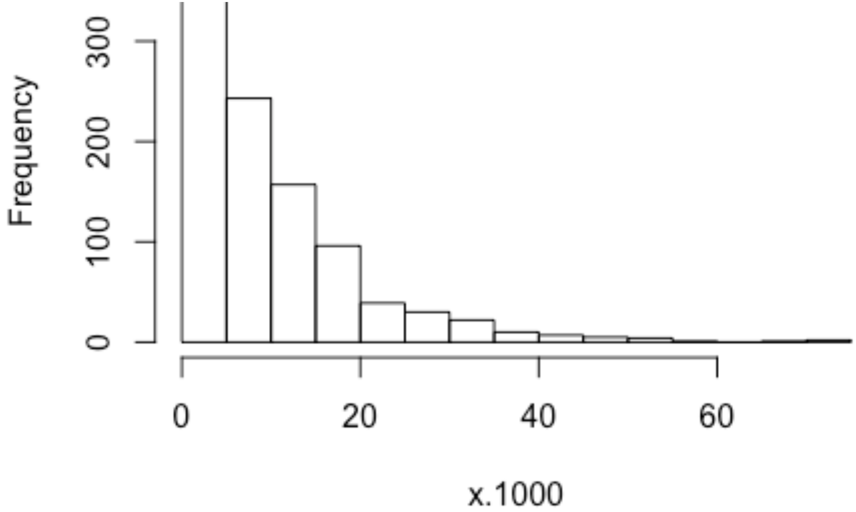
e) Hence, describe the distribution of the mean of the given dataset

```
hist(x, main ="100 Medians of samples of same size")
```




```
hist(x.1000, main ="1000 Medians of samples of same size")
```




Frequency
0
100
200
300
0
20
40
60
x.1000