Assignment 5 - The Bootstrap

Conceptual

1 - Explain how k-fold cross-validation is implemented.

--- Your answer here ---

2 - What are the advantages and disadvantages of k-fold crossvalidation relative to:

i. The validation set approach?

--- Your answer here ---

ii. LOOCV?

--- Your answer here ---

Practical

Overview of the steps

1. Loading the data and getting an overview of the data
2. Estimating the standard error of parameters of a Linear Regression Model
3. Estimating the standard error of parameters of a Quadratic Regression Model

Steps in detail

Loading the data and getting an overview of the data

Load the data file `Auto.rda` or `Auto.csv`.

In [27]:
```
load(file = "../ISLR/data/Auto.rda")
```

Display the number of predictors (including the response `mpg`) and their names:

In [28]:
```
dim(Auto)[2]
names(Auto)
```

9

'mpg' 'cylinders' 'displacement' 'horsepower' 'weight' 'acceleration' 'year' 'origin' 'name'

Print a statistic summary of the predictors and the response `medv` :

In [29]:
```
summary(Auto)
```

```
      mpg          cylinders      displacement     horsepower    
weight
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Mi
n.   :1613
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st
Qu.:2225
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Med
ian :2804
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mea
n   :2978
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd
Qu.:3615
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Ma
x.   :5140

  acceleration        year           origin                      nam
e
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :
5
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :
5
 Median :15.50   Median :76.00   Median :1.000   toyota corolla    :
5
 Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :
4
 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :
4
 Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:
4
                                                 (Other)           :
365
```

Display the number of data points:

In [30]:
```
dim(Auto)[1]
```

392

Display the data in a table (subset of rows is sufficient):

In [31]:
```
Auto
```

A data.frame: 392 × 9

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	na
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<f
1	18	8	307	130	3504	12.0	70	1	chevr chev mal
2	15	8	350	165	3693	11.5	70	1	bu skylark :
3	18	8	318	150	3436	11.0	70	1	plymo sate
4	16	8	304	150	3433	12.0	70	1	amc re
5	17	8	302	140	3449	10.5	70	1	ford tor
6	15	8	429	198	4341	10.0	70	1	ford gala !
7	14	8	454	220	4354	9.0	70	1	chevr imp
8	14	8	440	215	4312	8.5	70	1	plymo fur
9	14	8	455	225	4425	10.0	70	1	pont cata
10	15	8	390	190	3850	8.5	70	1	a ambassa
11	15	8	383	170	3563	10.0	70	1	do challen
12	14	8	340	160	3609	8.0	70	1	plymo 'cuda :
13	15	8	400	150	3761	9.5	70	1	chevr monte c
14	14	8	455	225	3086	10.0	70	1	buick est wagon (
15	24	4	113	95	2372	15.0	70	3	toy cor ma
16	22	6	198	95	2833	15.5	70	1	plymo dus
17	18	6	199	97	2774	15.5	70	1	amc hor
18	21	6	200	85	2587	16.0	70	1	f mave
19	27	4	97	88	2130	14.5	70	3	dat pl
20	26	4	97	46	1835	20.5	70	2	volkswa 1131 del se
21	25	4	110	87	2672	17.5	70	2	peug !
22	24	4	107	90	2430	14.5	70	2	audi 10
23	25	4	104	95	2375	17.5	70	2	saab !

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	na
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<f
24	26	4	121	113	2234	12.5	70	2	bmw 2(
25	21	6	199	90	2648	15.0	70	1	amc grer
26	10	8	360	215	4615	14.0	70	1	ford f2
27	10	8	307	200	4376	15.0	70	1	chevy (
28	11	8	318	210	4382	13.5	70	1	dodge d2
29	9	8	304	193	4732	18.5	70	1	hi 12(
30	27	4	97	88	2130	14.5	71	3	dats pl5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
368	28	4	112	88	2605	19.6	82	1	chevro cava
369	27	4	112	88	2640	18.6	82	1	chevro cava wag
370	34	4	112	88	2395	18.0	82	1	chevro cavalier d
371	31	4	112	85	2575	16.2	82	1	pont j2000 hatchba
372	29	4	135	84	2525	16.0	82	1	dodge ar
373	27	4	151	90	2735	18.0	82	1	pont phoe
374	24	4	140	92	2865	16.4	82	1	f fairm fut
375	36	4	105	74	1980	15.3	82	2	volkswag rabl
376	37	4	91	68	2025	18.2	82	3	mazda custo
377	31	4	91	68	1970	17.6	82	3	mazda cust
378	38	4	105	63	2125	14.7	82	1	plymo hori: mi
379	36	4	98	70	2125	17.3	82	1	merc lyı
380	36	4	120	88	2160	14.5	82	3	niss stanza
381	36	4	107	75	2205	14.5	82	3	hor acc
382	34	4	108	70	2245	16.9	82	3	toy cor
383	38	4	91	67	1965	15.0	82	3	honda c
384	32	4	91	67	1965	15.7	82	3	honda c (au

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	na
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<f
385	38	4	91	67	1995	16.2	82	3	datsun :
386	25	6	181	110	2945	16.4	82	1	bu cent limi
387	38	6	262	85	3015	17.0	82	1	oldsmol cutl ci (die
388	26	4	156	92	2585	14.5	82	1	chry leba medall
389	22	6	232	112	2835	14.7	82	1	f granac
390	32	4	144	96	2665	13.9	82	3	toy celica
391	36	4	135	84	2370	13.0	82	1	doc charger
392	27	4	151	90	2950	17.3	82	1	chevr cam
393	27	4	140	86	2790	15.6	82	1	f mustan
394	44	4	97	52	2130	24.6	82	2	vw picl
395	32	4	135	84	2295	11.6	82	1	doc rampa
396	28	4	120	79	2625	18.6	82	1	ford ran
397	31	4	119	82	2720	19.4	82	1	chevy s

Compute the pairwise correlation of the predictors in the data set.

In R , we need to download and install a library first.

In [32]:
```
install.packages("corrplot")
source("http://www.sthda.com/upload/rquery_cormat.r")
```

```
The downloaded binary packages are in
	/var/folders/ct/4pcck8t94sdfc73rhymq4t140000gp/T//RtmpNHDrd
X/downloaded_packages
```

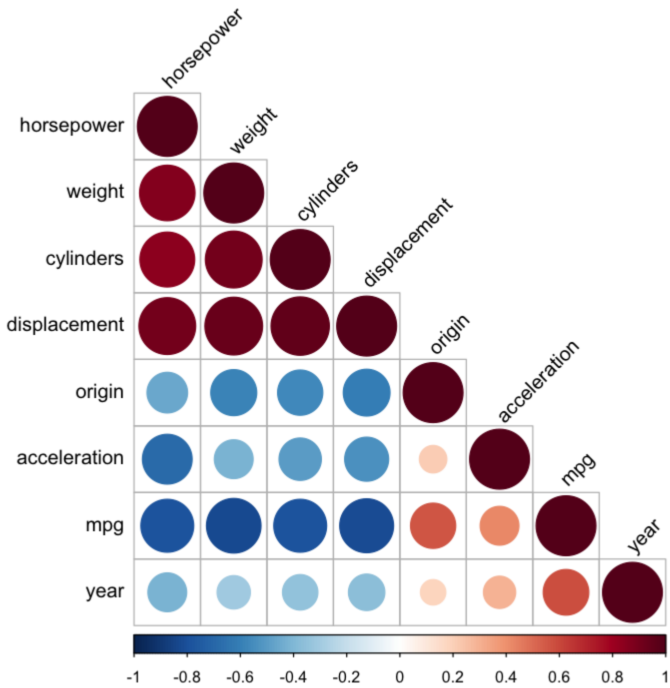
In [33]:
```
rquery.cormat(Auto[,-9])
```

```
$r
             horsepower weight cylinders displacement origin acceler
ation  mpg
horsepower            1
weight             0.86      1
cylinders          0.84    0.9         1
displacement        0.9   0.93      0.95            1
origin            -0.46  -0.59     -0.57        -0.61      1
acceleration      -0.69  -0.42      -0.5        -0.54   0.21
1
mpg               -0.78  -0.83     -0.78        -0.81   0.57
0.42    1
year              -0.42  -0.31     -0.35        -0.37   0.18
0.29 0.58
             year
horsepower
weight
cylinders
displacement
origin
acceleration
mpg
year            1

$p
             horsepower   weight cylinders displacement  origin acce
leration
horsepower            0
weight         1.4e-118        0
cylinders      4.6e-107 9.3e-141         0
displacement   1.5e-140 3.5e-175  1.3e-200            0
origin          1.9e-21  2.3e-37   5.3e-35      4.5e-42       0
acceleration    1.6e-56  6.6e-18     1e-26      1.5e-31 2.2e-05
0
mpg               7e-81   6e-102   1.3e-80      1.7e-90 1.8e-34
1.8e-18
year            7.2e-18    4e-10   1.9e-12      3.7e-14   3e-04
4.7e-09
                 mpg year
horsepower
weight
cylinders
displacement
origin
acceleration
mpg                0
year         1.1e-36    0

$sym
             horsepower weight cylinders displacement origin acceler
ation mpg
horsepower   1
weight       +          1
cylinders    +          +      1
displacement +          *      *         1
origin       .          .      .         ,            1
acceleration ,          .      .         .                   1
mpg          ,          +      ,         +            .      .
1
year         .          .      .         .
.
```

```
                 year
horsepower
weight
cylinders
displacement
origin
acceleration
mpg
year         1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```



Estimating the Accuracy of a Linear Regression Model

We first create a simple function, `boot.fn()` , which takes in the `Auto` data set as well as a set of indices for the observations, and returns the intercept and slope estimates for the linear regression model. We then apply this function to the full set of 392 observations in order to compute the estimates of β_0 and β_1 on the entire data set using the usual linear regression coefficient estimate formulas.

In [34]:
```
boot.fn=function(data,index) return(coef(lm(mpg~horsepower,data=da
boot.fn(Auto ,1:392)
```

(Intercept)	39.9358610211705
horsepower	-0.157844733353654

The `boot.fn()` function can also be used in order to create bootstrap estimates for the intercept and slope terms by randomly sampling from among the observations with replacement. Here two examples where the `sample()` function creates different training data sets based on the original `Auto` data.

In [35]:
```
set.seed(1)
boot.fn(Auto,sample(392,392,replace=T))
```

(Intercept)	40.3404516830189
horsepower	-0.163486837689938

In [36]:
```
boot.fn(Auto,sample(392,392,replace=T))
```

(Intercept)	40.1186906449022
horsepower	-0.157706320543503

Next, we use the `boot()` function to compute the standard errors of $1,000$ bootstrap estimates for the intercept and slope terms.

In [37]:
```
library(boot)
boot(Auto,boot.fn ,1000)
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Auto, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
      original        bias    std. error
t1* 39.9358610  0.0544513229 0.841289790
t2* -0.1578447 -0.0006170901 0.007343073
```

This indicates that the bootstrap estimate for $SE(\hat{\beta}_0) = 0.84$, and that the bootstrap estimate for $SE(\hat{\beta}_1) = 0.0073$.

Statistic formulas can be used to compute the standard errors for the regression coefficients in a linear model. In `R` these can be obtained using the `summary()` function on the results of the fitted logistic regression model.

In [38]:
```
summary(lm(mpg~horsepower ,data=Auto))$coef
```

A matrix: 2 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	39.9358610	0.717498656	55.65984	1.220362e-187
horsepower	-0.1578447	0.006445501	-24.48914	7.031989e-81

This indicates that the standard error for $SE(\beta_0) = 0.72$, and that the bootstrap estimate for $SE(\beta_1) = 0.0064$.

Interprete the results!

Estimating the Accuracy of a Quadratic Regression Model

Below the bootstrap standard error estimates and the standard linear regression estimates that result from fitting the quadratic model to the data. Since this model provides a good fit to the data, there is now a better correspondence between the bootstrap estimates of $SE(\hat{\beta}_0)$, $SE(\hat{\beta}_1)$, and $SE(\hat{\beta}_2)$.

In [23]:
```
boot.fn=function(data,index) coefficients(lm(mpg~horsepower+I(hors
set.seed(1)
boot(Auto, boot.fn, 1000)
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Auto, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
        original        bias     std. error
t1* 56.900099702  3.511640e-02 2.0300222526
t2* -0.466189630 -7.080834e-04 0.0324241984
t3*  0.001230536  2.840324e-06 0.0001172164
```

In [24]:
```
summary(lm(mpg~horsepower+I(horsepower^2),data=Auto))$coef
```

A matrix: 3 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	56.900099702	1.8004268063	31.60367	1.740911e-109
horsepower	-0.466189630	0.0311246171	-14.97816	2.289429e-40
I(horsepower^2)	0.001230536	0.0001220759	10.08009	2.196340e-21

Compare again differences in the standard errors between the bootstrap estimates and the statistic estimates of $SE(\beta_0)$, $SE(\beta_1)$, and $SE(\beta_2)$.

Summarize and then interpret the results!