JW Ch 5 Lab

David Brady

2/7/2020

Instructions and time-saving hints. To begin this lab, download the archive and unpack it. Inside the folder, you will find a Rmarkdown file.

The purpose of this lab is to implement linear regression in R. You will reproduce the commands blocks in JW Section 5.3 in this Rmarkdown file. **Please do not retype all the R commands as shown in Section 5.3.** Instead, go to link to text for Lab JW Ch 5, cut and paste the text into your Rmarkdown file. Then break up the R commands into the R chunks shown in the text. (R chunks are braced by triple backticks and a leading {r}, as before.) Do not put all commands into the same block! The idea is to imitate the code chunks in Section 5.3 with the additional plots and output omitted from the text. Below, I have given you the first two blocks for reference.

After creating a new chunk, you may knit the document. Read Section 5.3 as you go. You may comment out the `fix()` commands if you do not wish to close editor windows during knitting. You may also wish to comment out the commands which deliberately produce an error during the exercise. Since R chunks are independent pieces of code, you will produce some errors when imitating the format in the book. For example, the `abline()` commands must be grouped in the same chunk as the initial `plot()` command.

Congratulations! When the document is successfully knitted, you should reread Section 5.3 along with the Rmarkdown output. You should see the commands along with the output plots. Submit your Rmd file along with an unzipped PDF of the result before the deadline.

Chaper 5 Lab: Cross-Validation and the Bootstrap

The Validation Set Approach

```
#chooseCRANmirror()
install.packages("ISLR", repos = "https://cloud.r-project.org/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rx/01w0w5ds7t36p51ggtr6fdr80000gn/T//RtmpzQLFdI/downloaded_packages
```

```
library(ISLR)
set.seed(1)
```

```
train=sample(392,196)
lm.fit=lm(mpg~horsepower,data=Auto,subset=train)
attach(Auto)
mean((mpg-predict(lm.fit,Auto))[-train]^2)
```

```
## [1] 23.26601
```

```r
lm.fit2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)
mean((mpg-predict(lm.fit2,Auto))[-train]^2)
```

```
## [1] 18.71646
```

```r
lm.fit3=lm(mpg~poly(horsepower,3),data=Auto,subset=train)
mean((mpg-predict(lm.fit3,Auto))[-train]^2)
```

```
## [1] 18.79401
```

```r
set.seed(2)
train=sample(392,196)
```

```r
lm.fit=lm(mpg~horsepower,subset=train)
mean((mpg-predict(lm.fit,Auto))[-train]^2)
```

```
## [1] 25.72651
```

```r
lm.fit2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)
mean((mpg-predict(lm.fit2,Auto))[-train]^2)
```

```
## [1] 20.43036
```

```r
lm.fit3=lm(mpg~poly(horsepower,3),data=Auto,subset=train)
mean((mpg-predict(lm.fit3,Auto))[-train]^2)
```

```
## [1] 20.38533
```

Leave-One-Out Cross-Validation

```r
glm.fit=glm(mpg~horsepower,data=Auto)
coef(glm.fit)
```

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

```r
lm.fit=lm(mpg~horsepower,data=Auto)
coef(lm.fit)
```

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

```r
library(boot)
```

```r
glm.fit=glm(mpg~horsepower,data=Auto)
cv.err=cv.glm(Auto,glm.fit)
cv.err$delta
```

```
## [1] 24.23151 24.23114
```

```r
cv.error=rep(0,5)
```

```r
for (i in 1:5){
glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
cv.error[i]=cv.glm(Auto,glm.fit)$delta[1]
}
cv.error
```

```
## [1] 24.23151 19.24821 19.33498 19.42443 19.03321
```

k-Fold Cross-Validation

```r
set.seed(17)
cv.error.10=rep(0,10)
for (i in 1:10){
glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
cv.error.10[i]=cv.glm(Auto,glm.fit,K=10)$delta[1]
}
cv.error.10
```

```
##  [1] 24.27207 19.26909 19.34805 19.29496 19.03198 18.89781 19.12061 19.14666
##  [9] 18.87013 20.95520
```

The Bootstrap

```r
alpha.fn=function(data,index){
X=data$X[index]
Y=data$Y[index]
return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))
}
```

```r
alpha.fn(Portfolio,1:100)
```

```
## [1] 0.5758321
```

```r
set.seed(1)
```

```r
alpha.fn(Portfolio,sample(100,100,replace=T))
```

```
## [1] 0.7368375
```

```r
boot(Portfolio,alpha.fn,R=1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Portfolio, statistic = alpha.fn, R = 1000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.5758321 -0.001695873  0.09366347
```

Estimating the Accuracy of a Linear Regression Model

```r
boot.fn=function(data,index)
return(coef(lm(mpg~horsepower,data=data,subset=index)))
```

```r
boot.fn(Auto,1:392)
```

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

```r
set.seed(1)
```

```r
boot.fn(Auto,sample(392,392,replace=T))
```

```
## (Intercept)  horsepower
##  40.3404517  -0.1634868
```

```r
boot.fn(Auto,sample(392,392,replace=T))
```

```
## (Intercept)  horsepower
##  40.1186906  -0.1577063
```

```r
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
```

```r
summary(lm(mpg~horsepower,data=Auto))$coef
```

```
##               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
```

```r
boot.fn=function(data,index)
coefficients(lm(mpg~horsepower+I(horsepower^2),data=data,subset=index))
set.seed(1)
```

```r
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
```

```r
summary(lm(mpg~horsepower+I(horsepower^2),data=Auto))$coef
```

```
##                     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
```