Cars

Trevor Hastie and Robert Tibshirani

Here, I am adapting the lab associated with Chapter 5 of the textbook.

```
library(ISLR2)
library(boot)
```

Estimating the Accuracy of a Linear Regression Model

The bootstrap approach can be used to assess the variability of the coefficient estimates and predictions from a statistical learning method. Here we use the bootstrap approach in order to assess the variability of the estimates for β_0 and β_1, the intercept and slope terms for the *simple* linear regression model that uses `horsepower` to predict `mpg` in the `Auto` data set. We will compare the estimates obtained using the bootstrap to those obtained using the formulas for $\mathrm{SE}(\hat{\beta}_0)$ and $\mathrm{SE}(\hat{\beta}_1)$ described in Section 3.1.2 (*and the slides from class*).

Let's make some plots of the data to begin with.

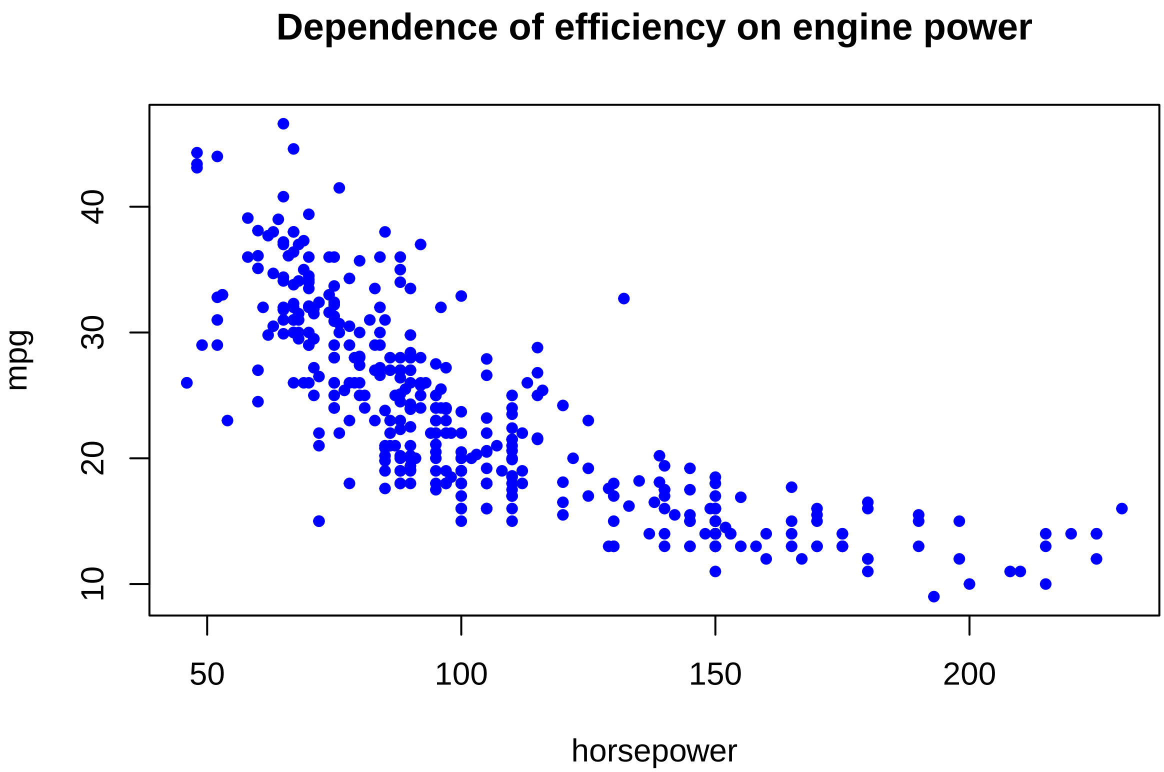
```
Auto
```

```
##    mpg cylinders displacement horsepower weight acceleration year origin
## 1   18         8          307        130   3504         12.0   70      1
## 2   15         8          350        165   3693         11.5   70      1
## 3   18         8          318        150   3436         11.0   70      1
## 4   16         8          304        150   3433         12.0   70      1
## 5   17         8          302        140   3449         10.5   70      1
## 6   15         8          429        198   4341         10.0   70      1
## 7   14         8          454        220   4354          9.0   70      1
## 8   14         8          440        215   4312          8.5   70      1
## 9   14         8          455        225   4425         10.0   70      1
## 10  15         8          390        190   3850          8.5   70      1
## 11  15         8          383        170   3563         10.0   70      1
##                         name
## 1  chevrolet chevelle malibu
## 2          buick skylark 320
## 3         plymouth satellite
## 4              amc rebel sst
## 5                ford torino
## 6           ford galaxie 500
## 7           chevrolet impala
## 8          plymouth fury iii
## 9           pontiac catalina
## 10        amc ambassador dpl
## 11       dodge challenger se
##  [ reached 'max' / getOption("max.print") -- omitted 381 rows ]
```

```
attach(Auto)
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

```r
#start with the scatterplot
plot(horsepower, mpg,
     main="Dependence of efficiency on engine power",
     pch=20, col="blue")
```



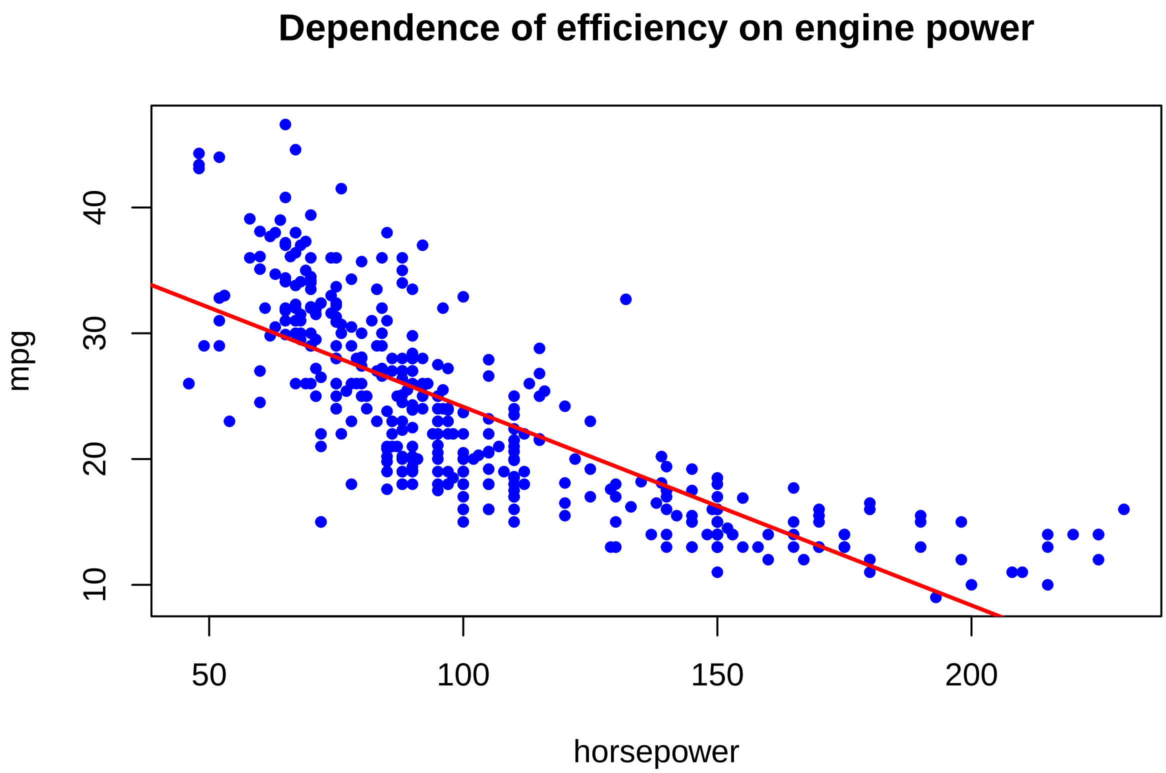
It looks suspiciously non-linear. So, let's add the least-squares line.

```r
plot(horsepower, mpg,
     main="Dependence of efficiency on engine power",
     pch=20, col="blue")
reg=lm(mpg ~ horsepower)
summary(reg)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower)
##
## 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
##
```

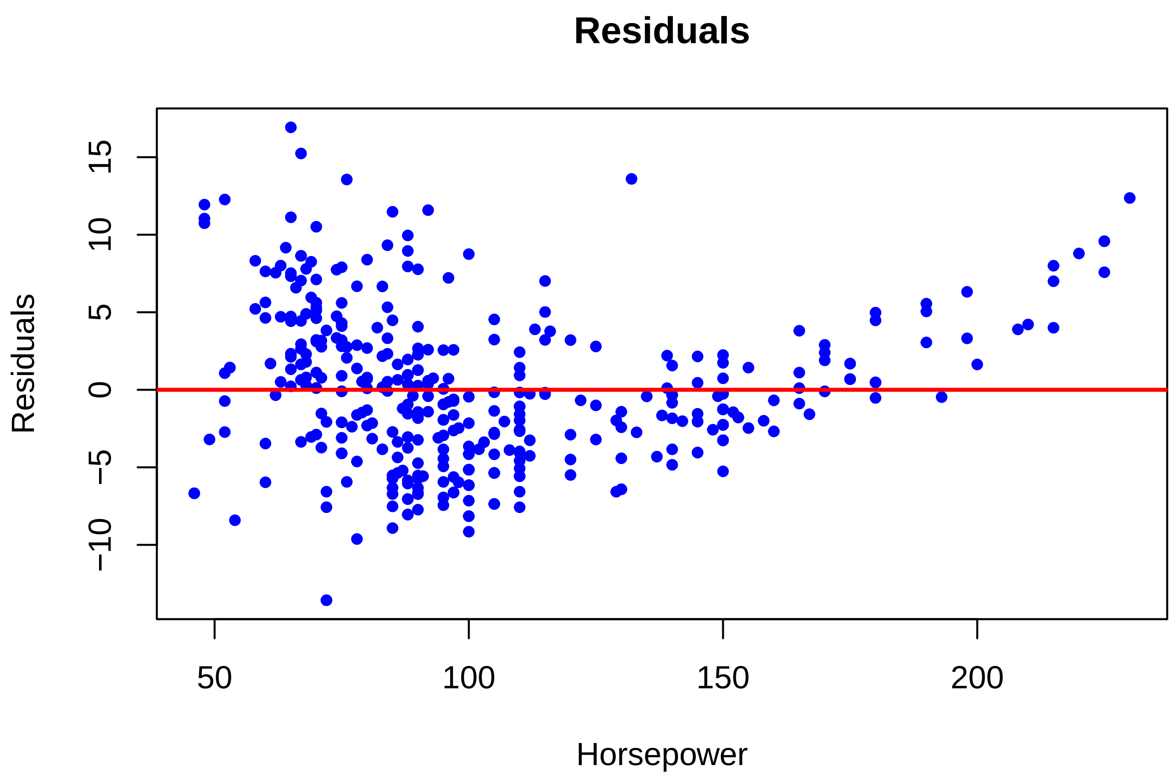
```
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
abline(reg, col="red", lwd=2)
```



Now, what about the residuals? We want to see if the residuals have an association with the explanatory, i.e., engine power.

```r
res=summary(reg)$residuals
plot(horsepower, res,
     main="Residuals",
     xlab="Horsepower", ylab="Residuals",
     pch=20, col="blue")
abline(0,0, col="red", lwd=2)
```



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 $n = 392$ 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 from Chapter 3. Note that we do not need the `{` and `}` at the beginning and end of the function because it is only one line long.

```
boot.fn <- function(data, index)
  coef(lm(mpg ~ horsepower, data = data, subset = index))
boot.fn(Auto, 1:392)
```

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

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 we give two examples.

```
set.seed(1)
boot.fn(Auto, sample(392, 392, replace = T))
```

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

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

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

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

```
boot(Auto, boot.fn, R=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($\hat{\beta}_0$) is 0.84, and that the bootstrap estimate for SE($\hat{\beta}_1$) is 0.0073. As discussed in Section 3.1.2, standard formulas can be used to compute the standard errors for the regression coefficients in a linear model. These can be obtained using the `summary()` function.

```
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
```

The standard error estimates for $\hat{\beta}_0$ and $\hat{\beta}_1$ obtained using the formulas from Section 3.1.2 are 0.717 for the intercept and 0.0064 for the slope. Interestingly, these are somewhat different from the estimates obtained using the bootstrap. Does this indicate a problem with the bootstrap? In fact, it suggests the opposite. Recall that the standard formulas for the standard errors rely on certain assumptions. For example, they depend on the unknown parameter σ^2, the noise variance. We then estimate σ^2 using the RSS. **Now, although the formulas for the standard errors do not rely on the linear model being correct, the estimate for σ^2 does.** We **earlier** that there is a non-linear relationship in the data, and so the residuals from a linear fit will be inflated, and so will $\hat{\sigma}^2$. Secondly, the standard formulas assume (somewhat unrealistically) that the x_i are fixed, and all the variability comes from the variation in the errors ϵ_i. The bootstrap approach does not rely on any of these assumptions, and so it is likely giving a more accurate estimate of the standard errors of $\hat{\beta}_0$ and $\hat{\beta}_1$ than is the `summary()` function.

Below we compute 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 (Figure 3.8), there is now a better correspondence between the bootstrap estimates and the standard estimates of SE($\hat{\beta}_0$), SE($\hat{\beta}_1$) and SE($\hat{\beta}_2$).

```
boot.fn <- function(data, index)
  coef(
      lm(mpg ~ horsepower + I(horsepower^2),
        data = data, subset = index)
    )
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  0.035116401844 2.0300222526
```

```
## t2* -0.466189630 -0.000708083404 0.0324241984
## t3*  0.001230536  0.000002840324 0.0001172164
```

```
q.reg=lm(mpg ~ horsepower + I(horsepower^2), data = Auto)
betas=q.reg$coef
betas
```

```
##     (Intercept)      horsepower I(horsepower^2)
##    56.900099702    -0.466189630     0.001230536
```

How about a picture?

```
plot(horsepower, mpg,
     main="Dependence of efficiency on engine power",
     pch=20, col="blue")
b.0=betas[[1]]
b.1=betas[[2]]
b.2=betas[[3]]
curve(b.0+b.1*x+b.2*x^2, col="red", lwd=2, add=TRUE)
```