R 고급

회귀분석

임요한

서울대학교

Aug, 2018

Credit 자료

```
library(datasets)
library(MASS)
library(ISLR)
head(Credit)
```

```
##   ID  Income Limit Rating Cards Age Education Gender Student Marr
## 1  1  14.891  3606    283     2  34        11   Male      No
## 2  2 106.025  6645    483     3  82        15 Female     Yes
## 3  3 104.593  7075    514     4  71        11   Male      No
## 4  4 148.924  9504    681     3  36        11 Female      No
## 5  5  55.882  4897    357     2  68        16   Male      No
## 6  6  80.180  8047    569     4  77        10   Male      No
##   Ethnicity Balance
## 1 Caucasian     333
## 2     Asian     903
## 3     Asian     580
## 4     Asian     964
## 5 Caucasian     331
## 6 Caucasian    1151
```

단순회귀, 중회귀계수의 의미

```
attach(Credit)
y=Balance
inc=Income
g=Gender
lm(y~inc)$coefficients
```

```
## (Intercept)         inc
##  246.514751    6.048363
```

```
beta1=sqrt(var(y)/var(inc))*cor(y,inc)
beta1
```

```
## [1] 6.048363
```

```r
lm(y~inc+g)$coefficients
```

```
## (Intercept)         inc     gFemale
##  233.766327    6.052069   24.310839
```

```r
residx=lm(inc~g)$resid
residy=lm(y~g)$resid
beta.inc=sqrt(var(residy)/var(residx))*cor(residy,residx)
beta.inc
```

```
## [1] 6.052069
```

변수선택

```
credit.fit <-lm(Balance~.,data=Credit)
```

```
summary(credit.fit)
```

```
##
## Call:
## lm(formula = Balance ~ ., data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -166.48  -77.62  -14.37   56.21  316.52
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -487.07424   36.73407 -13.259  < 2e-16 ***
## ID                   0.04105    0.04343   0.945   0.3452
## Income              -7.80740    0.23431 -33.321  < 2e-16 ***
## Limit                0.19052    0.03279   5.811  1.3e-08 ***
## Rating               1.14249    0.49100   2.327   0.0205 *
## Cards               17.83639    4.34324   4.107  4.9e-05 ***
## Age                 -0.62955    0.29449  -2.138   0.0332 *
## Education           -1.09831    1.59817  -0.687   0.4924
## GenderFemale        -9.54615    9.98431  -0.956   0.3396
```

stepwise selection by AIC

```
aic.credit <- stepAIC(credit.fit, direction="both")
```

```
## Start:  AIC=3687.3
## Balance ~ ID + Income + Limit + Rating + Cards + Age + Education
##     Gender + Student + Married + Ethnicity
##
##              Df Sum of Sq      RSS    AIC
## - Ethnicity   2     13972  3791981 3684.8
## - Education   1      4611  3782619 3685.8
## - Married     1      7002  3785011 3686.0
## - ID          1      8721  3786730 3686.2
## - Gender      1      8924  3786933 3686.2
## <none>                     3778009 3687.3
## - Age         1     44612  3822621 3690.0
## - Rating      1     52855  3830864 3690.9
## - Cards       1    164641  3942650 3702.4
## - Limit       1    329664  4107672 3718.8
## - Student     1   6334026 10112035 4079.1
```

stepwise 최종모형

$$Balance\ Income + Limit + Rating + Cards + Age + Student$$

```
credit.step <-lm(Balance~Income+Limit+Rating+Cards+Age+Student,data=Credit)
```

```r
summary(credit.step)
```

```
##
## Call:
## lm(formula = Balance ~ Income + Limit + Rating + Cards + Age +
##     Student, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -170.00  -77.85  -11.84   56.87  313.52
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -493.73419   24.82476 -19.889  < 2e-16 ***
## Income        -7.79508    0.23342 -33.395  < 2e-16 ***
## Limit          0.19369    0.03238   5.981 4.98e-09 ***
## Rating         1.09119    0.48480   2.251   0.0250 *
## Cards         18.21190    4.31865   4.217 3.08e-05 ***
## Age           -0.62406    0.29182  -2.139   0.0331 *
## StudentYes   425.60994   16.50956  25.780  < 2e-16 ***
## ---
```

All subset search

leaps(x=, y=, wt=rep(1, NROW(x)), int=TRUE, method=c("Cp", "adjr2", "r2"), nbest=10, names=NULL, df=NROW(x), strictly.compatible=TRUE)

```r
#install.packages("leaps")
library(leaps)
all.sub<-regsubsets(Balance~.,data=Credit,nbest=2)
head(all.sub)
```

```
## $np
## [1] 13
##
## $nrbar
## [1] 78
##
## $d
##  [1] 4.000000e+02 9.493750e+01 9.998446e+01 4.947910e+05 7.499396
##  [6] 9.978718e+01 3.519613e+06 4.939730e+01 3.547434e+01 1.129859
## [11] 3.845081e+03 9.083460e+06 5.176162e+06
##
## $rbar
##  [1]  6.125000e-01  4.975000e-01  4.521889e+01  2.957500e+00  5.1
##  [6]  3.549400e+02  2.550000e-01  1.000000e-01  5.566750e+01  1.3
## [11]  4.735600e+03  2.005000e+02  1.171824e-02  2.575991e+00 -2.7
## [16]  1.277156e-02  1.165714e+01  7.926267e-02 -4.739961e-02 -2.5
## [21]  3.133641e-01  1.474233e+02  5.324556e+00 -1.415789e+00 -1.5
```

회귀분석 1을 위한 보충 R 코드와 결과

광고자료

```
adv = read.csv("Advertising.csv", header=T, sep=",")
adv = adv[,-1]
names(adv) = tolower(names(adv))
head(adv)
```

```
##      tv radio newspaper sales
## 1 230.1  37.8      69.2  22.1
## 2  44.5  39.3      45.1  10.4
## 3  17.2  45.9      69.3   9.3
## 4 151.5  41.3      58.5  18.5
## 5 180.8  10.8      58.4  12.9
## 6   8.7  48.9      75.0   7.2
```

파일 Advertising.csv의 첫번째 컬럼이 관측치의 번호인데, 첫번째 명령은 이 것도 읽어들여서 하나의 변수로 만든다. 두번째 명령은 이 컬럼을 없앤다. 파일에는 변수 이름의 첫번째 문자가 대문자인데 이것을 소문자로 바꾸었다.

모형 1: sales ∼ TV

모형의 적합

```
lm.fit = lm(sales ~ tv, data=adv)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = sales ~ tv, data = adv)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## tv          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

lm.fit으로부터 정보의 추출

```
names(lm.fit)
str(lm.fit)
```

lm.fit안에 어떤 객체들이 포함되어 있는지 보고, 구조를 본다.

```
coef(lm.fit)
```

```
## (Intercept)          tv
##  7.03259355  0.04753664
```

회귀계수 추정량을 리턴한다.

```
confint(lm.fit, level=0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 6.12971927 7.93546783
## tv          0.04223072 0.05284256
```

회귀계수의 신뢰구간을 구해준다.

```
predict(lm.fit, data.frame(tv=c(230.1, 44.5, 17.2)), interval="confidence")
```

```
##        fit       lwr       upr
## 1 17.970775 17.337774 18.603775
## 2  9.147974  8.439101  9.856848
## 3  7.850224  7.024932  8.675515
```

```
predict(lm.fit, data.frame(tv=c(230.1, 44.5, 17.2)), interval="prediction")
```

```
##        fit       lwr      upr
## 1 17.970775 11.513546 24.42800
## 2  9.147974  2.682867 15.61308
## 3  7.850224  1.371318 14.32913
```

```r
predict(lm.fit, data.frame(tv=c(230.1, 44.5, 17.2)), interval="none")
```

```
##          1          2          3
## 17.970775  9.147974  7.850224
```

주어진 설명변수의 값에서 예측값을 구한다. 옵션 interval의 값이 confidence이면 예측치의 평균의 신뢰구간을 prediction이면 예측구간을 계산하고 none이면 구간추정치를 계산하지 않는다. level이라는 옵션이 있는데, 디폴트가 0.95이다. 이는 95% 레블을 의미한다.

잔차분석

```
fitted(lmfit)
```

$\hat{y}$값들을 추출한다.

```
residuals(lmfit)
```

잔차들을 추출한다.

```
rstandard(lm.fit)
```

표준화된 잔차(standardized residual)

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

을 추출한다. 잔차를 표준정규분포와 비슷하도록 척도를 바꾼것이다.

```
rstudent(lm.fit)
```

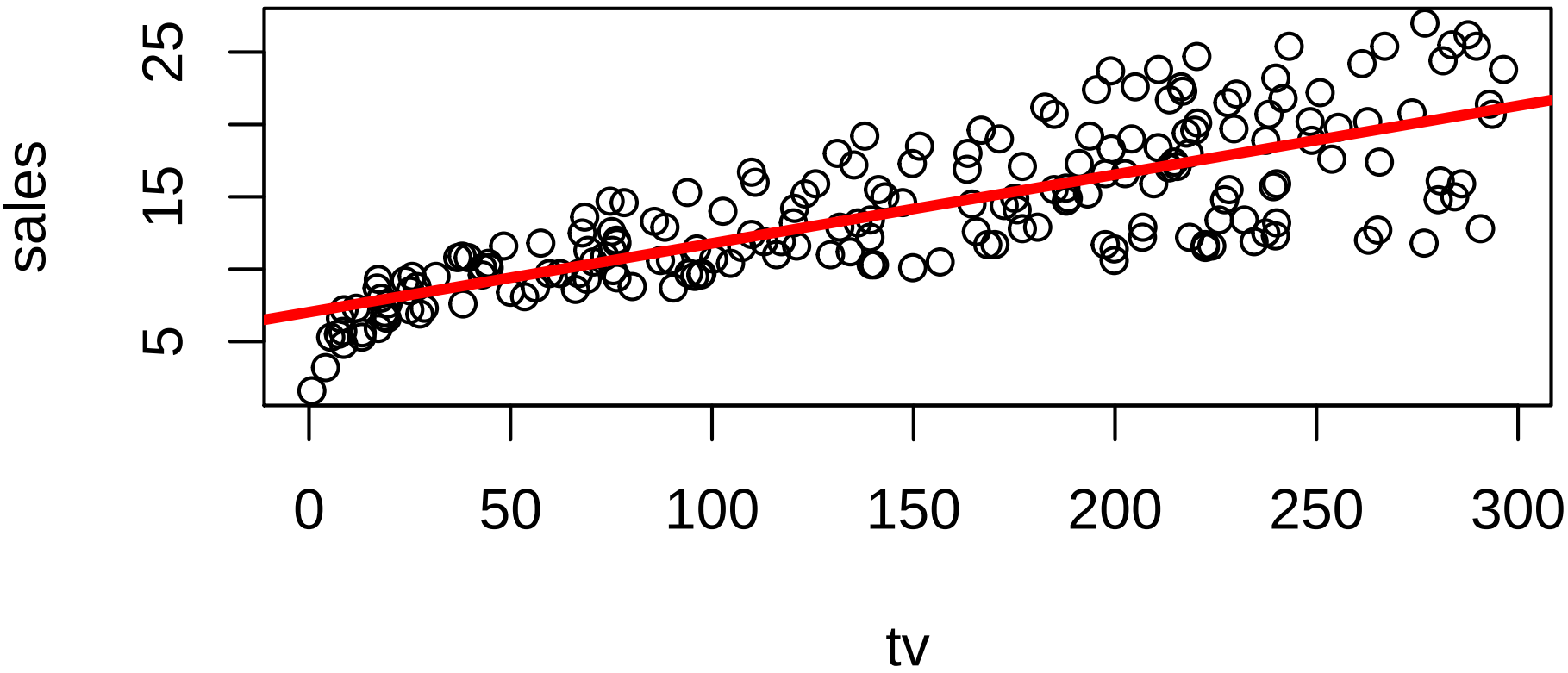
스튜던트화된 잔차(studentized residual)

$$r_i = \frac{e_i}{\hat{\sigma}_{-i}\sqrt{1-h_{ii}}}$$

을 구한다. 위의 식에서 e_i는 잔차, s는 오차항의 표준편차의 추정량, h_{ii}는 지렛데(leverage) 통계량을 말한다. 표준화된 잔차와 개념은 비슷한데, 여기서는 y_i를 빼고 구한 y_i의 잔차를 표준화한 것이다. 즉, 하나빼기(leave-one-out)를 한 잔차의 표준화이다. $\hat{\sigma}_{-i}$는 y_i를 빼고 구한 σ의 추정량이다.

자료와 추정된 직선의 그림

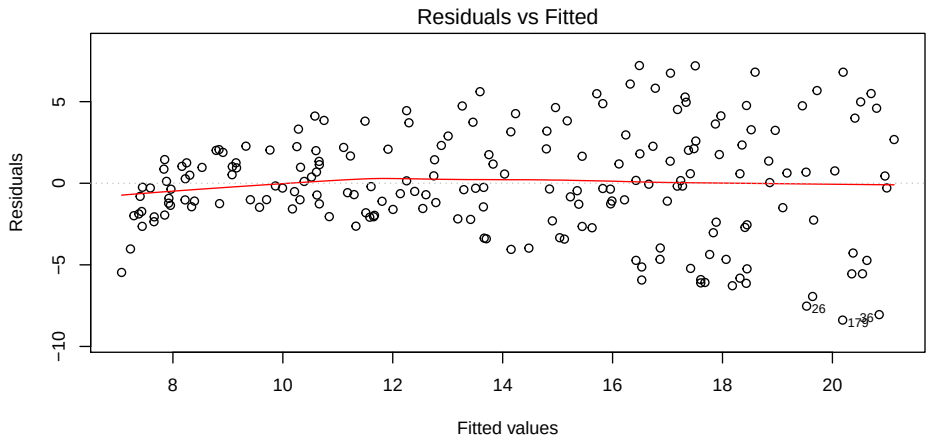
```
attach(adv)
plot(tv, sales)
abline(lm.fit, lwd=3, col="red")
```

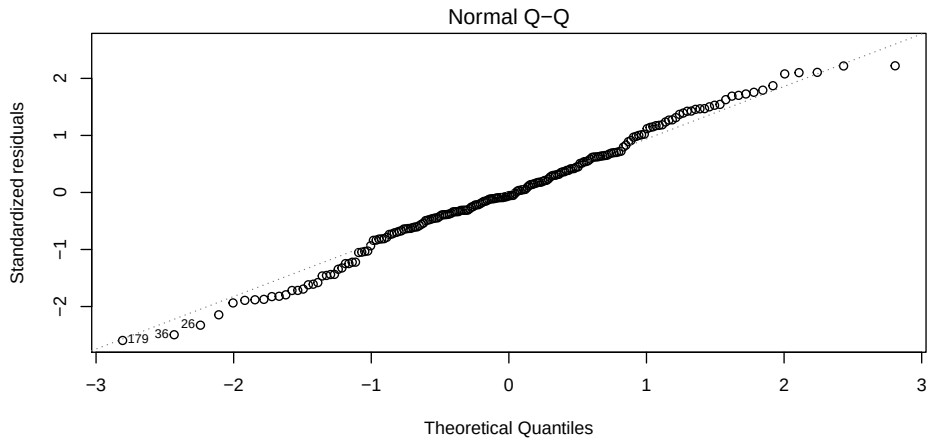


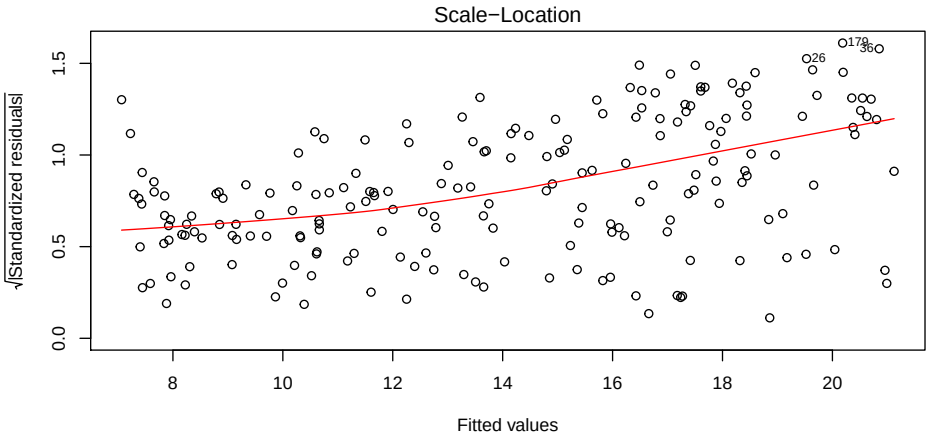
tv와 sales의 산점도를 그리고 그 위에 추정된 회귀직선을 그린다. lwd는 선의 두께를 정하는 옵션으로 디폴트는 1이고 이 값의 의미는 device-specific하다.

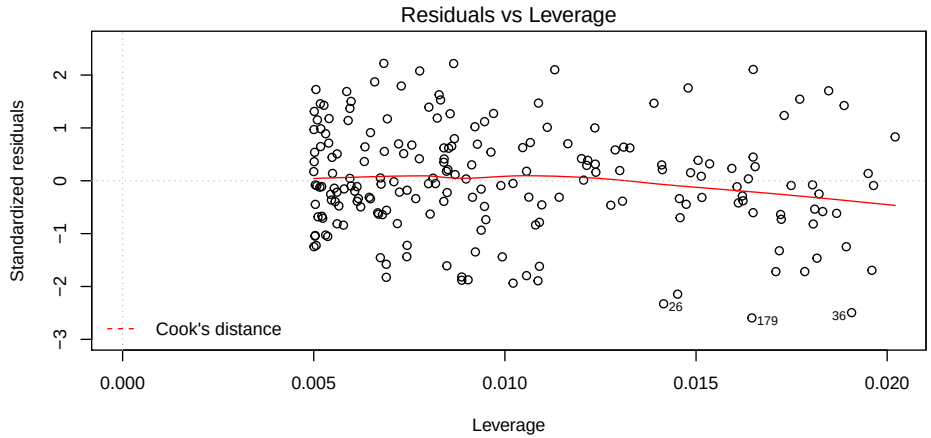
잔차그림: 모형진단을 위한 그림들

```
par(mfrow=c(2,2))
plot(lm.fit)
```










```
par(mfrow=c(1,1))
```

네 개의 그림을 준다. 왼쪽 위부터 시계방향으로 다음과 아래와 같은 그림이다. 실제로

```
par(mfrow=c(2,2))
plot(lm.fit)
par(mfrow=c(1,1))
```

잔차그림은 "잔차" vs "설명변수" 또는 "fitted value"의 산점도를 그린다.

실제로

```
plot(lm.fit, which=1:6)
```

로 쓰면 6개의 그림을 준다. 위의 명령은 디폴트로 4개의 그림을 주는데 6개의 그림 중 1,2,3, 5번이다. 하나의 그림은

```
plot(lm.fit, which=2)
```

와 같은 명령으로 그릴 수 있다.

- Tukey-Anscombe plot. 예측값 vs 잔차. 예측값에 따라 잔차의 분산이 균등한지, 커지는지, 작아지는지를 본다. 균등하지 않다면 등분산가정이 성립하지 않는 것이다. 예측값에 따라서 잔차의 평균을 빨간색으로 그렸는데, 이 것이 거의 0 근처에 있어야 한다. 이 것이 어떤 패턴을 보인다면 모형에 중요한 변수가 빠졌다는 의미이다.
- 표준화된 잔차의 정규분포 QQ 그림. 오차의 정규성가정을 검토하는 그림이다. 기울기가 1인 직선을 따르면 정규가정이 위배되지 않는다는 것이다.
- Scale-location plot. 예측값 vs $\sqrt{|\text{표준화된 잔차}|}$. 등분산을 알아보기 위해 보는 그림이다. 여기에 lowess 그림을 덧그려서 보기도 한다고 한다.

  참고: 왜 제곱근을 취하는지는 잘 모르겠다. 제곱근을 안취하고 |표준화된 잔차|와 그림을 그려도 될 것 같은데 말이다.

$$\sqrt{|\text{표준화된 잔차}|} \approx \sigma + \text{표준정규분포를 따르는 부분}$$

  위와 같은 이유가 있지 않을까 추측을 한다.

모형 2: sales ∼ tv + radio + newspaper

```
lm.fit = lm(sales ~ tv + radio + newspaper, data=adv)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = sales ~ tv + radio + newspaper, data = adv)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## tv           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 ' ' 0.1 ' ' 1
```

상관계수의 계산

```
cor(adv)
```

```
##                    tv      radio  newspaper     sales
## tv         1.00000000 0.05480866 0.05664787 0.7822244
## radio      0.05480866 1.00000000 0.35410375 0.5762226
## newspaper 0.05664787 0.35410375 1.00000000 0.2282990
## sales      0.78222442 0.57622257 0.22829903 1.0000000
```

모형 3: sales ∼ tv + radio + tv×radio (교호작용)

```
lm.fit = lm(sales ~ tv*radio, data=adv)
lm.fit$coefficients
```

```
## (Intercept)          tv       radio    tv:radio
## 6.750220203 0.019101074 0.028860340 0.001086495
```

```
lm.fit = lm(sales ~ tv+radio+tv:radio, data=adv)
lm.fit$coefficients
```

```
## (Intercept)          tv       radio    tv:radio
## 6.750220203 0.019101074 0.028860340 0.001086495
```

위 두 모형식이 동일하다.

자동차 자료

```r
Auto = read.csv("Auto.csv", header=T, sep=",")
#library(ISLR)
Auto$horsepower = as.numeric(Auto$horsepower)
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year
## 1  18         8          307         16   3504         12.0   70
## 2  15         8          350         34   3693         11.5   70
## 3  18         8          318         28   3436         11.0   70
## 4  16         8          304         28   3433         12.0   70
## 5  17         8          302         23   3449         10.5   70
## 6  15         8          429         41   4341         10.0   70
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

알지못하는 이유로 horsepower의 클래스가factor로 입력이 된다. 이를 numeric으로 바꾸었다

모형 1: mpg $\sim$ horsepower + horsepower^2

모형의 적합

```
lm.fit = lm(mpg ~ poly(horsepower, 2, raw=T), data = Auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ poly(horsepower, 2, raw = T), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.056  -5.899  -0.310   4.519  21.196
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|
## (Intercept)                       15.8986199  0.9699768  16.391  < 2e
## poly(horsepower, 2, raw = T)1  0.2229724  0.0494908   4.505 8.75e
## poly(horsepower, 2, raw = T)2 -0.0011007  0.0005109  -2.154   0.0
## ---
```

```r
attach(Auto)
x = poly(horsepower, 2)
t(x) %*% x
```

```
##                 1             2
## 1 1.000000e+00 4.434895e-17
## 2 4.434895e-17 1.000000e+00
```

위에서 결과가 거의 I인 것을 알수 있다.

```r
lm.fit = lm(mpg ~ horsepower + I(horsepower^2), data = Auto)
lm.fit$coefficients
```

```
##      (Intercept)       horsepower I(horsepower^2)
##       15.8986199        0.2229724       -0.0011007
```

I(horsepower^2)는 계산한 식을 새로운 변수로 보라는 뜻이다.

자료 크기순 정렬

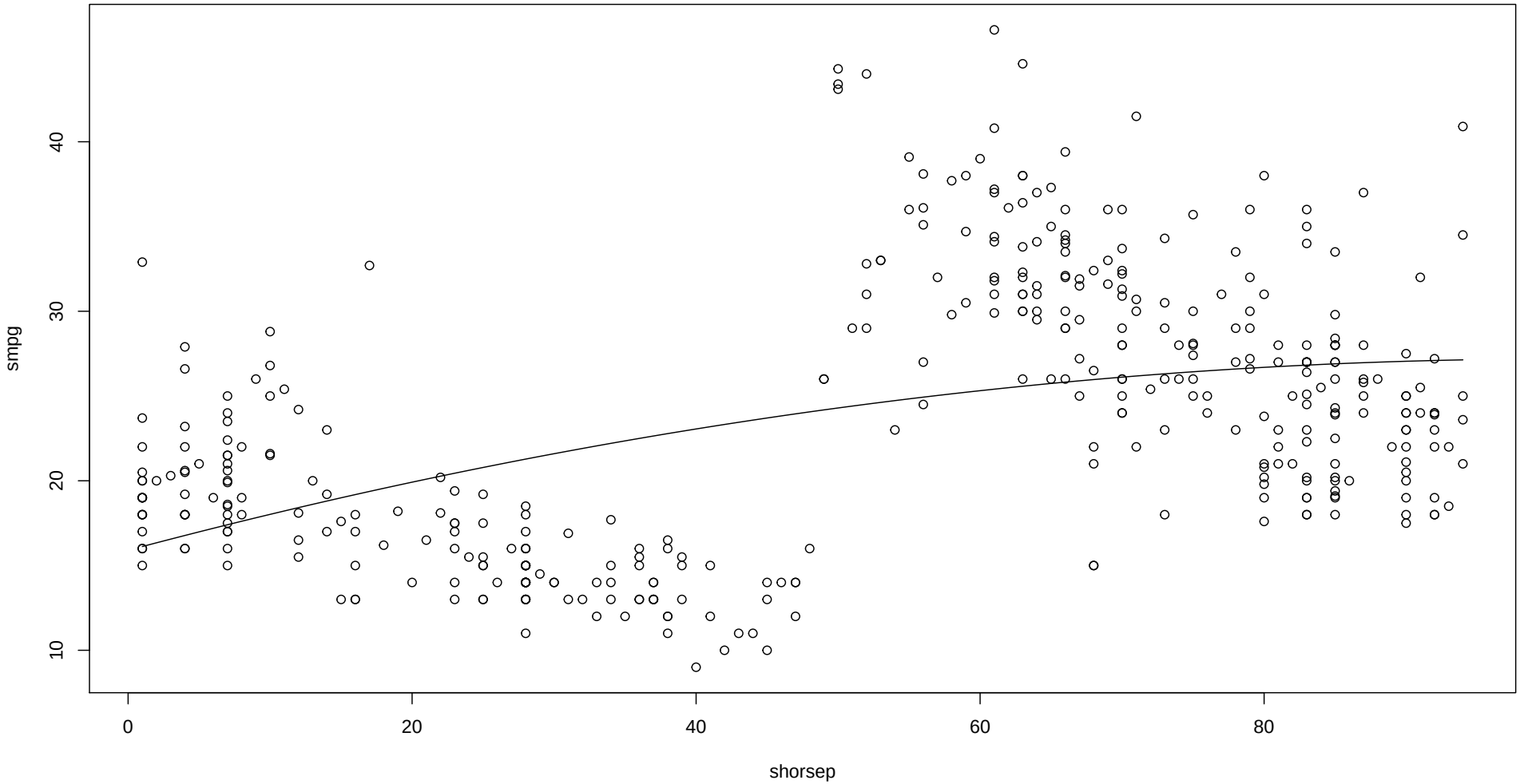
```
r=rank(Auto$horsepower,ties="random")
ind=1:length(Auto$horsepower)
smpg=Auto$mpg
shorsep=Auto$horsepower
smpg[r]=Auto$mpg
shorsep[r]=Auto$horsepower
```

```r
lm.sfit = lm(smpg ~ poly(shorsep, 2))
lm.sfit$coefficients
```

```
##          (Intercept) poly(shorsep, 2)1 poly(shorsep, 2)2
##             23.51587          70.57115         -14.97904
```

그림

```
plot(shorsep, smpg)
lines(shorsep, fitted(lm.sfit))
```



교재 3장에서 사용된 자료, csv 파일, 그리고 ISLR package에서의 자동차 자료가 모두 다르다.