Section 4.3 Statistical Inference (UFFI Example, p. 102-111)

Doing multiple linear regression and statistical inference using matrices and vectors in R.

```
> dat=read.table("http://www.stat.ucla.edu/~hqxu/stat100C/data/uffi.txt", h=T)  #  Table 1.2, p.7
> dim(dat)  # 24*3
[1] 24  3
> dat[1,]  # first case
   CH2O Air UFFI
1 31.33   0    0
> y=dat$CH2O;  x1=dat$UFFI; x2=dat$Air;    # 24*1 vectors
> x0=rep(1, nrow(dat))   # x0 is a vector of 24 ones
> X=cbind(x0, x1, x2)  # X is a 24*3 matrix
> XX = t(X) %*% X   # X'X :  t(X) is the transpose of X,  %*% is for matrix multiplication
> XX
    x0 x1  x2
x0  24 12 123
x1  12 12  61
x2 123 61 823
> Xy = t(X) %*% y  # X'y is a 3*1 matrix
> Xy
      [,1]
x0 1215.81
x1  662.35
x2 6776.22
> XX.inv = solve( XX )  # inverse of X'X
> XX.inv  # p. 104
            x0            x1            x2
x0  0.22194577 -0.0855690177 -0.0268282129
x1 -0.08556902  0.1667027261  0.0004327131
x2 -0.02682821  0.0004327131  0.0051925573
>
> beta.hat = XX.inv %*% Xy # LSE of beta =(X'X)^{-1} X'y
> beta.hat   # p.104
        [,1]
x0 31.373371
x1  9.312042
x2  2.854509
>
> mu.hat=X %*% beta.hat    # fitted values
> e=y-mu.hat   # residuals
> SSE=t(e) %*% e   # =sum(e^2)
> s.sq = SSE/(nrow(X)-ncol(X)); s.sq  # df=nrow(X)-ncol(X)=n-p-1=24-3=21
        [,1]
[1,] 27.2807
> s=sqrt(s.sq); s   # s
         [,1]
[1,] 5.223093
> s*sqrt(diag(XX.inv))  # SE of beta.hat, p. 105
[1] 2.4606576 2.1325495 0.3763730
> beta.hat/( s*sqrt(diag(XX.inv)) ) # t values
        [,1]
x0 12.749994
x1  4.366624
x2  7.584256
```

```
>
> a=c(1, 1, 5)  # a case with UFFI=1, Air=5, p. 105
> t(a) %*% beta.hat  # estimate of theta=a'beta
         [,1]
[1,] 54.95796
> t(a) %*% XX.inv %*% a  # a'(X'X)^{-1}a
          [,1]
[1,] 0.0833694
> se = s * sqrt(t(a) %*% XX.inv %*% a)  # SE
> qt(1-0.05/2, 21) # alpha=0.05, df=21
[1] 2.079614
> qt(1-0.05/2, 21)* se # 95% CI error margin, p.106
         [,1]
[1,] 3.136272
> qt(1-0.05/2, 21)* s * sqrt(1 + t(a) %*% XX.inv %*% a) # 95% PI error margin, p. 111
         [,1]
[1,] 11.30573
>
> # an easy way using R
> g = lm(CH2O ~ Air + UFFI, dat)
> summary(g)

Call:
lm(formula = CH2O ~ Air + UFFI, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5460 -3.1312 -0.1389  3.5779  8.3620

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  31.3734     2.4607  12.750 2.36e-11 ***
Air           2.8545     0.3764   7.584 1.92e-07 ***
UFFI          9.3120     2.1325   4.367 0.000270 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 5.223 on 21 degrees of freedom
Multiple R-Squared: 0.7827,Adjusted R-squared: 0.762
F-statistic: 37.82 on 2 and 21 DF,  p-value: 1.095e-07

> # residual plots, p.103
> par(mfrow=c(2,2))  # display 4 graphs in 2 rows and 2 columns on a page
> plot(resid(g), xlab="Observation numbers", ylab="Residuals")  # Fig 4.5(a)
> plot(fitted(g), resid(g),  xlab="Fitted values", ylab="Residuals") # Fig. 4.5(b)
> plot(x2, resid(g),  xlab="Airtightness", ylab="Residuals")  # Fig. 4.5(c)
> plot(x1, resid(g),  xlab="UFFI", ylab="Residuals")  #
```