General Linear Test with R

When gasoline is pumped into the tank of a car, vapors are vented into the atmosphere. An experiment was conducted to determine whether y, the amount of vapor, can be predicted using the following four variables based on initial conditions of the tank and the dispensed gasoline:

```
x1 = tank temperature (Degrees F)
x2 = gasoline temperature (Degrees F)
x3 = vapor pressure in tank (psi)
x4 = vapor pressure of gasoline (psi)
```

```
> gas = 
read.table("http://www.utstat.toronto.edu/~brunner/302f14/code_n_data/lecture/vapou
r.data", header=T)
> head(gas)
   y x1 x2   x3   x4
1 29 33 53 3.32 3.42
2 24 31 36 3.10 3.26
3 26 33 51 3.18 3.18
4 22 37 51 3.39 3.08
5 27 36 54 3.20 3.41
6 21 35 35 3.03 3.03
> cor(gas)
           y        x1        x2        x3        x4
y  1.0000000 0.8260665 0.9093507 0.8698845 0.9213333
x1 0.8260665 1.0000000 0.7742909 0.9554116 0.9337690
x2 0.9093507 0.7742909 1.0000000 0.7815286 0.8374639
x3 0.8698845 0.9554116 0.7815286 1.0000000 0.9850748
x4 0.9213333 0.9337690 0.8374639 0.9850748 1.0000000
> fullmodel = lm(y ~ x1+x2+x3+x4, data=gas)
> summary(fullmodel)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4, data = gas)

Residuals:
   Min     1Q Median     3Q    Max 
-5.586 -1.221 -0.118  1.320  5.106 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  1.01502    1.86131   0.545  0.59001   
x1          -0.02861    0.09060  -0.316  0.75461   
x2           0.21582    0.06772   3.187  0.00362 **
x3          -4.32005    2.85097  -1.515  0.14132   
x4           8.97489    2.77263   3.237  0.00319 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.73 on 27 degrees of freedom
Multiple R-squared: 0.9261, Adjusted R-squared: 0.9151 
F-statistic: 84.54 on 4 and 27 DF,  p-value: 7.249e-15
```

Tank temperature = x_1 and Vapour pressure in tank = x_3 are highly correlated, r = 0.96. They could be washing each other out. Test them simultaneously.

H_0: $\beta_1 = \beta_3 = 0$ H_0: $\mathbf{C\beta} = \mathbf{t}$ $$F = \frac{(\mathbf{C}\widehat{\boldsymbol{\beta}}-\mathbf{t})'(\mathbf{C}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C}')^{-1}(\mathbf{C}\widehat{\boldsymbol{\beta}}-\mathbf{t})}{q\,s^2}$$

```
> # Call it LL instead of C, because R uses C for contrasts.
> LL = rbind(c(0,1,0,0,0),
+            c(0,0,0,1,0))
> V = vcov(fullmodel) # Don't need MSE because it's already in V
> q=dim(LL)[1]
> betahat = fullmodel$coefficients
> Cbeta = LL %*% betahat; center = solve(LL %*% V %*% t(LL))
> Fstat = as.numeric( t(Cbeta) %*% center %*% Cbeta ) / q ; Fstat
[1] 2.49292
> dfe = fullmodel$df.residual; dfe
[1] 27
> pval = 1-pf(Fstat,q,dfe); pval
[1] 0.1015035
```

Conclusion: Controlling for gasoline temperature and vapour pressure of gasoline, there is no evidence that tank temperature or vapour pressure in tank are related to amount of vapour released from the gas tank.

Get the same test with full versus reduced model.

```
> redmodel = lm(y ~ x2 + x4, data=gas)
> anova(redmodel,fullmodel)
Analysis of Variance Table

Model 1: y ~ x2 + x4
Model 2: y ~ x1 + x2 + x3 + x4
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     29 238.39
2     27 201.23  2    37.159 2.4929 0.1015
```

```
RSS = SSE
```

```
> # A handy function for the general linear test: Use it freely
> source("http://www.utstat.toronto.edu/~brunner/Rfunctions/ftest.txt")
> ftest # To see function definition
function(model,L,h=0)
# General linear test of H0: L beta = h
    {
    BetaHat = model$coefficients
    dimL = dim(L)
    if(length(BetaHat) != dimL[2]) stop("Sizes of L and Beta are incompatible")
    r = dimL[1]
    if(qr(L)$rank != r) stop("Rows of L must be linearly independent.")
    out = numeric(4)
    names(out) = c("F","df1","df2","p-value")
    dfe = df.residual(model)
    diff = L%*%BetaHat-h
    fstat = t(diff) %*% solve(L%*%vcov(model)%*%t(L)) %*% diff / r
    # Note vcov = MSE * XtXinv
    fstat = as.numeric(fstat)
    out[1] = fstat; out[2]=r; out[3]=dfe
    out[4] = 1-pf(fstat,r,dfe)
    return(out)
    }
> # Reproduce the test of H0: beta1=beta3=0
> # Recall F=2.49292, df = (2,27), p = 0.1015035

> ftest(fullmodel,LL)
         F        df1        df2    p-value
 2.4929204  2.0000000 27.0000000  0.1015035
```