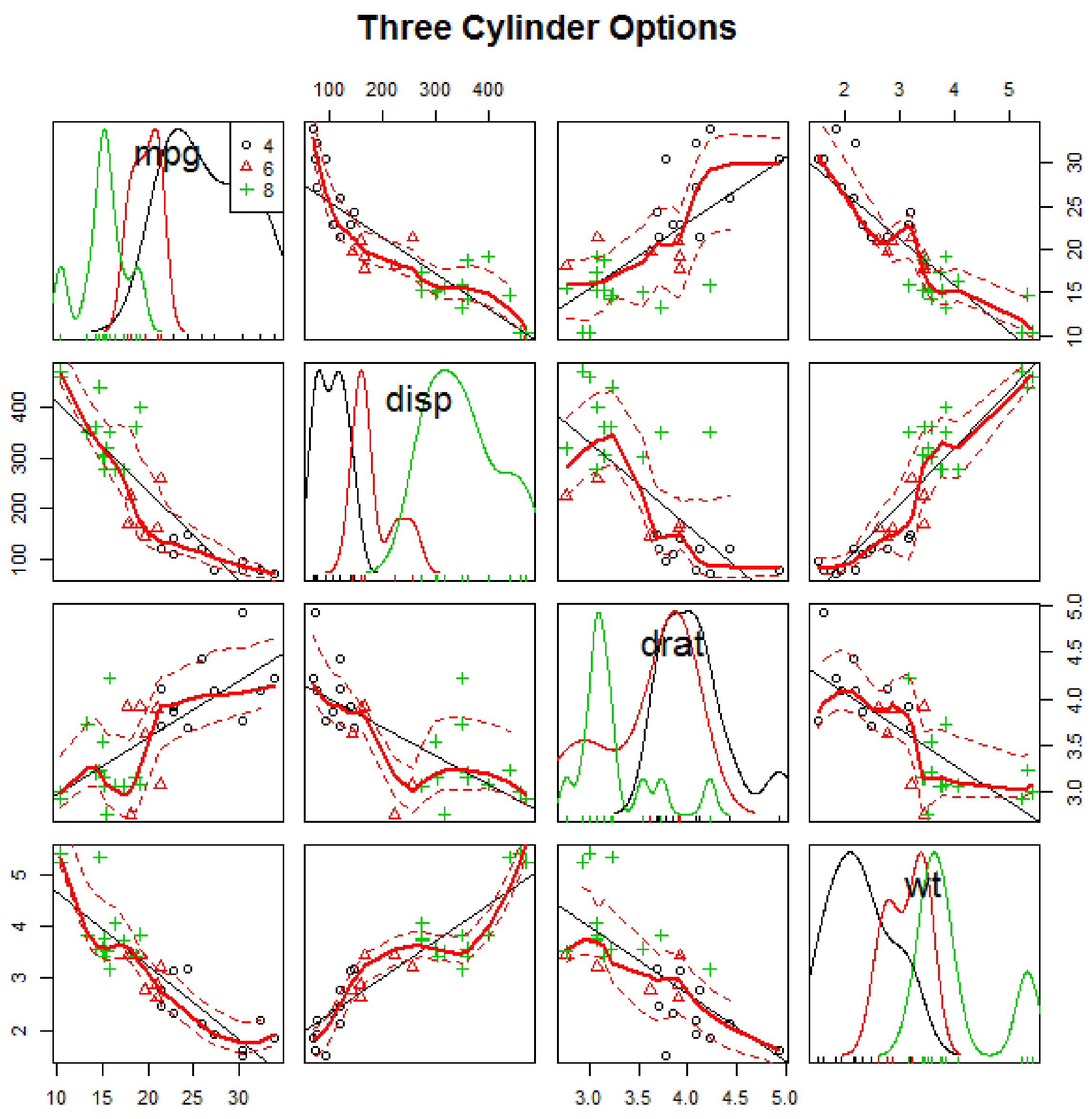
GLM Solutions

5.



It seems that mpg generally decreases with displacement, increases with rear-axle ratio, and decreases with weight. Displacement seems to decrease with rear-axle ratio, and increase with weight. Rear-axle ratio seems to decrease with weight. It seems the lower number of cylinders corresponds to better gas mileage, smaller engine displacement, lower weight, and higher rear-axle ratio. This all seems to make some sense.

6. Here's the sample correlation matrix:

```
           mpg          disp          drat          wt
mpg      1.0000000   -0.8475514    0.6811719    -0.8676594
disp    -0.8475514    1.0000000   -0.7102139     0.8879799
drat     0.6811719   -0.7102139    1.0000000    -0.7124406
wt      -0.8676594    0.8879799   -0.7124406     1.0000000
```

Miles per gallon and weight seem highly negatively correlated, and also miles per gallon and displacement seem highly negatively correlated. There is a high positive correlation between displacement and weight. Actually, all of the correlation values seem somewhat significant in magnitude. The directions of the correlations make sense, too: larger engine displacement tends to correspond with larger automobiles, which corresponds with higher gas mileage.

7.

```
Call:
lm(formula = mpg ~ disp + drat + wt)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2342 -2.3719 -0.3148  1.6315  6.2820

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 31.043257   7.099792   4.372 0.000154 ***
disp        -0.016389   0.009578  -1.711 0.098127 .
drat         0.843965   1.455051   0.580 0.566537
wt          -3.172482   1.217157  -2.606 0.014495 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.951 on 28 degrees of freedom
Multiple R-squared:  0.7835,    Adjusted R-squared:  0.7603
F-statistic: 33.78 on 3 and 28 DF,  p-value: 1.92e-09
```

According to the output here, the intercept term is statistically significant, and also the weight variable and possibly the displacement variable. The p-value on the drat variable is not very low, so we might be able to eliminate that variable as a good predictor.

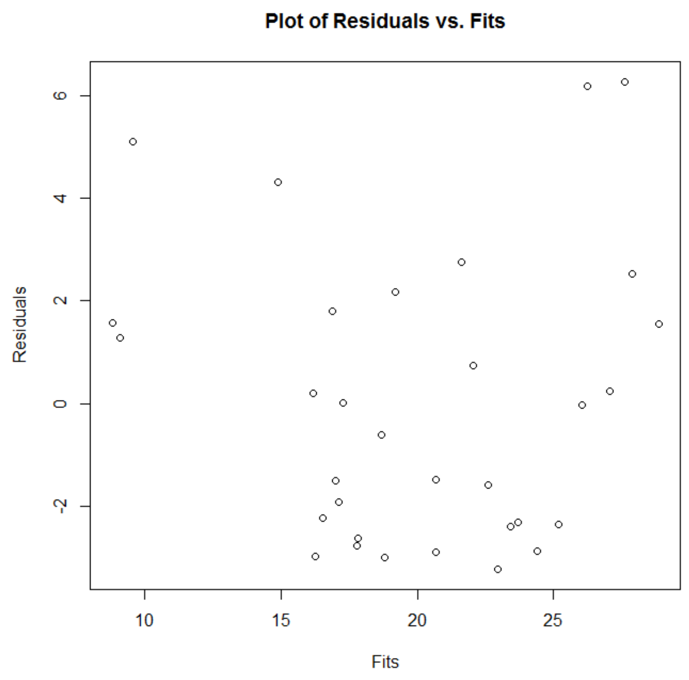
8.

```
> ones <- rep(1:1, nrow(mtcars))
> X <- cbind(ones, disp, drat, wt)
> b <- solve(t(X) %*% X) %*% t(X) %*% mpg
> b
           [,1]
ones 31.04325728
disp -0.01638916
drat  0.84396531
wt   -3.17248250
```

Yep, the model coefficients are the same.

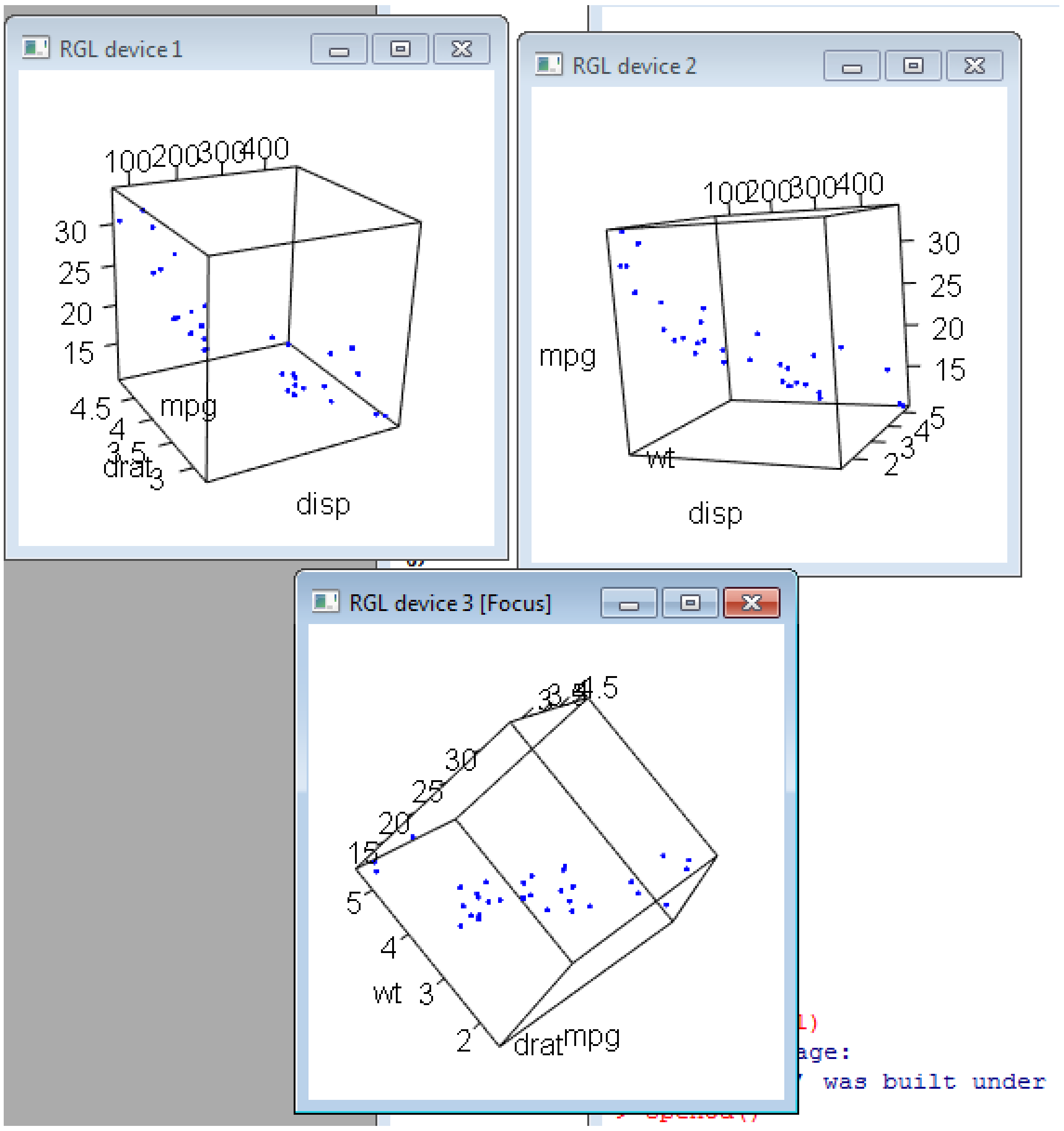
9.

```
> H <- X %*% solve(t(X) %*% X) %*% t(X)
> fits <- H %*% mpg
> res <- mpg – fits
> plot(res ~ fits, main="Plot of Residuals vs. Fits",
+ xlab = "Fits", ylab="Residuals")
```



The residuals don't seem to be centered around 0 here: there is a vague "U" pattern.

10. The mpg vs. displacement and rear-axle ratio plot indicates some curvature, as does the mpg vs. disp and wt plot. The mpg vs. wt and drat looks more linear. You'll really have to look at these 3d plots, and rotate them around some to see this apparent behavior.



11. So because a couple of those plots indicated possible curvature, we should probably think about doing a lack-of-fit test. Here's output:

```
Analysis of Variance Table

Response: mpg
           Df Sum Sq Mean Sq  F value  Pr(>F)
disp        1 808.89  808.89 825.3964 0.02215 *
drat        1  14.26   14.26  14.5537 0.16320
wt          1  59.14   59.14  60.3494 0.08150 .
Residuals    28 243.75    8.71
 Lack of fit 27 242.77    8.99   9.1751 0.25616
 Pure Error   1   0.98    0.98
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The lack-of-fit p-value is not small: it's .25616. According to this test, there's insufficient evidence to conclude a lack of linear fit.

12. Polynomial model:

```
Call:
lm(formula = mpg ~ disp + drat + wt + I(disp^2) + I(drat^2) +
   I(wt^2))

Residuals:
    Min      1Q  Median      3Q     Max
-3.5105 -1.5957 -0.4667  1.5837  4.1922

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.742e+01  1.947e+01   2.949  0.00682 **
disp        -8.749e-02  4.040e-02  -2.166  0.04008 *
drat        -6.684e+00  1.091e+01  -0.612  0.54581
wt          -3.953e+00  5.194e+00  -0.761  0.45372
I(disp^2)    1.322e-04  7.588e-05   1.743  0.09370 .
I(drat^2)    7.534e-01  1.473e+00   0.512  0.61343
I(wt^2)      5.806e-02  7.397e-01   0.078  0.93806
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.477 on 25 degrees of freedom
Multiple R-squared:  0.8638,    Adjusted R-squared:  0.8311
F-statistic: 26.43 on 6 and 25 DF,  p-value: 1.121e-09
```

It seems the intercept term is significant, as well as the displacement and possibly the square of the displacement. The weight variable suddenly no longer seems significant, although it was arguably the most significant variable in the linear model. It could be that allowing so many other variables to enter into the model that weight has just been supplanted by them, even though it is important. Or, since weight is very correlated with engine displacement, it could be that using disp and disp^2 is enough. Let's see what happens to those p-values when we build a couple smaller models:

```
> out5 <- lm(mpg ~ disp + I(disp^2))
> summary(out5)

Call:
lm(formula = mpg ~ disp + I(disp^2))

Residuals:
    Min      1Q  Median      3Q     Max
-3.9112 -1.5269 -0.3124  1.3489  5.3946

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.583e+01  2.209e+00  16.221 4.39e-16 ***
disp        -1.053e-01  2.028e-02  -5.192 1.49e-05 ***
I(disp^2)    1.255e-04  3.891e-05   3.226   0.0031 **
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.837 on 29 degrees of freedom
Multiple R-squared:  0.7927,    Adjusted R-squared:  0.7784
F-statistic: 55.46 on 2 and 29 DF,  p-value: 1.229e-10
```

Here, both the displacement and square of the displacement seem statistically significant, and the r^2 value is fairly high (almost 80%). Note that r^2 is less than what it was with the larger model, as is expected. Finally, let's see how much we can improve r^2 and/or r^2_adj by allowing the weight variable to enter:

```
> out5 <- lm(mpg ~ disp + I(disp^2) + wt)
> summary(out5)

Call:
lm(formula = mpg ~ disp + I(disp^2) + wt)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4225 -1.6596 -0.4234  1.7265  4.2112

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 41.4019837  2.4266906  17.061  2.5e-16 ***
disp        -0.0823950  0.0182460  -4.516 0.000104 ***
I(disp^2)    0.0001277  0.0000328   3.892 0.000561 ***
wt          -3.4179165  0.9545642  -3.581 0.001278 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.391 on 28 degrees of freedom
Multiple R-squared:  0.8578,    Adjusted R-squared:  0.8426
F-statistic: 56.32 on 3 and 28 DF,  p-value: 5.563e-12
```

Ah… This looks very promising. Both r^2 and r^2_adj have improved some (at least 5% pts each), and also all three variables in the model (displacement, the square of the displacement, and the weight) seem statistically significant. I think this is probably a good model. It would be cool to superimpose this model onto a plot.

13. We’ve actually already found the hat matrix and the fits. Let’s do a spot check for idempotency:

```
> head(H)
       [,1]       [,2]       [,3]       [,4]       [,5]       [,6]       [,7]       [,8]       [,9]       [,10]       [,11]
[1,] 0.04443107  0.041164804 0.048002537 0.022799593  0.020003588 0.01491056  0.019291451
0.03392821  0.03815314  0.034169024  0.034169024
[2,] 0.04116480  0.048964270 0.044793693 0.008729357 -0.010319645 0.01599254 -0.004604134
0.05615709  0.06321830  0.064741910  0.064741910
[3,] 0.04800254  0.044793693 0.067228820 0.040383243  0.008134076 0.05624594  0.004595774
0.05103475  0.04581872  0.035023219  0.035023219
[4,] 0.02279959  0.008729357 0.040383243 0.093387622  0.082187202 0.11374468  0.068099439
0.01433968 -0.01049646 -0.024059199 -0.024059199
[5,] 0.02000359 -0.010319645 0.008134076 0.082187202  0.150021490 0.03654540  0.131718501 -
0.05058991 -0.06239897 -0.071159324 -0.071159324
[6,] 0.01491056  0.015992540 0.056245945 0.113744682  0.036545398 0.19751155  0.025506284
0.06632404  0.02546174  0.009710623  0.009710623
        [,12]       [,13]       [,14]       [,15]       [,16]       [,17]       [,18]       [,19]       [,20]       [,21]       [,22]
[1,]  0.011510145 0.01586516 0.01522472 -0.007801307 -0.0087975088 -0.003931228  0.05348752
0.074511952 0.060462898 0.043641013  0.013206682
[2,]  0.029711314 0.01931203 0.02084133  0.007962531  0.0186031377  0.028336563  0.05582239
0.063392101 0.050800141 0.041563571 -0.007665306
[3,]  0.025194872 0.02947333 0.02884415 -0.037531727 -0.0387409761 -0.039706468  0.06925911
0.050469564 0.071439720 0.066933765  0.030692433
[4,]  0.048983136 0.06774345 0.06498458  0.017861631 -0.0008983455 -0.024865617  0.01783198 -
0.048120507 0.021145849 0.050770332  0.118896715
[5,] -0.001924862 0.03850612 0.03256038  0.052721335  0.0171934777 -0.003512279 -0.01662310
0.009771087 0.012849419 0.009616267  0.118679745
[6,]  0.108025463 0.10658282 0.10679497  0.015756735  0.0105762333 -0.021517196  0.02987142 -
0.134911612 0.006093382 0.078170904  0.138837437
       [,23]       [,24]       [,25]       [,26]       [,27]       [,28]       [,29]       [,30]       [,31]       [,32]
[1,] 0.020630850  0.024193108  0.013301842 0.05687885  0.05939692 0.05719835  0.04054797
0.038213186  0.0251264332  0.046109446
[2,] 0.004878057  0.021469344 -0.010926192 0.04763483  0.05272988 0.02132772  0.01490689
0.041747713  0.0211350682  0.062836797
[3,] 0.023129419 -0.012622967 -0.005408849 0.07251379  0.04782402 0.08336022 -0.01999497
0.058992702  0.0098648355  0.049730039
[4,] 0.077367283 -0.007635836  0.071547369 0.03248129 -0.01540768 0.09295715 -0.02704605
0.045426642  0.0246997857 -0.013779110
[5,] 0.096862551  0.065363961  0.143574528 0.01517724  0.01385245 0.09550780  0.12276995
0.001040532  0.0594428628 -0.046412537
[6,] 0.072008130 -0.067464512  0.026440662 0.02855691 -0.06435516 0.07645005 -0.16559850
0.079141639 -0.0008577625 -0.000266011
> HH <- H%*% H
> head(HH)
       [,1]       [,2]       [,3]       [,4]       [,5]       [,6]       [,7]       [,8]       [,9]       [,10]       [,11]
[1,] 0.04443107  0.041164804 0.048002537 0.022799593  0.020003588 0.01491056  0.019291451
0.03392821  0.03815314  0.034169024  0.034169024
```

```
[2,] 0.04116480  0.048964270 0.044793693 0.008729357 -0.010319645 0.01599254 -0.004604134
0.05615709  0.06321830  0.064741910  0.064741910
[3,] 0.04800254  0.044793693 0.067228820 0.040383243  0.008134076 0.05624594  0.004595774
0.05103475  0.04581872  0.035023219  0.035023219
[4,] 0.02279959  0.008729357 0.040383243 0.093387622  0.082187202 0.11374468  0.068099439
0.01433968 -0.01049646 -0.024059199 -0.024059199
[5,] 0.02000359 -0.010319645 0.008134076 0.082187202  0.150021490 0.03654540  0.131718501 -
0.05058991 -0.06239897 -0.071159324 -0.071159324
[6,] 0.01491056  0.015992540 0.056245945 0.113744682  0.036545398 0.19751155  0.025506284
0.06632404  0.02546174  0.009710623  0.009710623
           [,12]      [,13]      [,14]        [,15]         [,16]        [,17]       [,18]        [,19]       [,20]       [,21]        [,22]
[1,]  0.011510145 0.01586516 0.01522472 -0.007801307 -0.0087975088 -0.003931228  0.05348752
0.074511952 0.060462898 0.043641013  0.013206682
[2,]  0.029711314 0.01931203 0.02084133  0.007962531  0.0186031377  0.028336563  0.05582239
0.063392101 0.050800141 0.041563571 -0.007665306
[3,]  0.025194872 0.02947333 0.02884415 -0.037531727 -0.0387409761 -0.039706468  0.06925911
0.050469564 0.071439720 0.066933765  0.030692433
[4,]  0.048983136 0.06774345 0.06498458  0.017861631 -0.0008983455 -0.024865617  0.01783198 -
0.048120507 0.021145849 0.050770332  0.118896715
[5,] -0.001924862 0.03850612 0.03256038  0.052721335  0.0171934777 -0.003512279 -0.01662310
0.009771087 0.012849419 0.009616267  0.118679745
[6,]  0.108025463 0.10658282 0.10679497  0.015756735  0.0105762333 -0.021517196  0.02987142 -
0.134911612 0.006093382 0.078170904  0.138837437
          [,23]        [,24]        [,25]      [,26]       [,27]      [,28]       [,29]       [,30]         [,31]        [,32]
[1,] 0.020630850  0.024193108  0.013301842 0.05687885  0.05939692 0.05719835  0.04054797
0.038213186  0.0251264332  0.046109446
[2,] 0.004878057  0.021469344 -0.010926192 0.04763483  0.05272988 0.02132772  0.01490689
0.041747713  0.0211350682  0.062836797
[3,] 0.023129419 -0.012622967 -0.005408849 0.07251379  0.04782402 0.08336022 -0.01999497
0.058992702  0.0098648355  0.049730039
[4,] 0.077367283 -0.007635836  0.071547369 0.03248129 -0.01540768 0.09295715 -0.02704605
0.045426642  0.0246997857 -0.013779110
[5,] 0.096862551  0.065363961  0.143574528 0.01517724  0.01385245 0.09550780  0.12276995
0.001040532  0.0594428628 -0.046412537
[6,] 0.072008130 -0.067464512  0.026440662 0.02855691 -0.06435516 0.07645005 -0.16559850
0.079141639 -0.0008577625 -0.000266011
```

Yep, looks like H = H%*%H alright.

14. So we actually already got the vector of residuals, too. We stored them in "res":

```
> SSE <- t(res) %*% res
> MSE <- SSE / 28
> SSE
         [,1]
[1,] 243.7537
> MSE
        [,1]
[1,] 8.70549
```

We can check these values against those that we get with the anova() output:

```
> anova(out)
Analysis of Variance Table

Response: mpg
          Df Sum Sq Mean Sq F value    Pr(>F)    
disp       1 808.89  808.89 92.9171 2.152e-10 ***
drat       1  14.26   14.26  1.6383    0.2111    
wt         1  59.14   59.14  6.7937    0.0145 *  
Residuals 28 243.75    8.71                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yep, you can see the SSE and MSE values in the last line of the table there (not the line with the significance codes). Remember the degrees of freedom for SSE will be n – 1 – (# of parameters in model) = 32 – 1 – 3 = 28.

15. R^2 and r^2 adj are, respectively,

```
Multiple R-squared:  0.7835,    Adjusted R-squared:  0.7603
```

16. Since we're going to make simultaneous intervals for four parameters (intercept, disp, drat, and wt), we need an individual error rate of 1.25%, so that the individual CIs have confidence levels of 98.75%:

```
> confint(out, level =.9875)
              0.625 %     99.375 %
(Intercept) 12.09050982 49.996004740
disp        -0.04195827  0.009179952
drat        -3.04026255  4.728193182
wt          -6.42165691  0.076691912
```

So we can be about 95% confidence that the actual intercept is between 12.09 and 49.996, the displacement coefficient is between -.0419 and .0092, the drat coefficient is between -3.0403 and 4.7292, and the weight coefficient is between -6.4217 and .0767. Notice I've rounded the lower interval endpoints down in each case, and the upper ones up. This is good practice- to err on the side of caution. You want to be *at least* xyz% confident that the parameter value appears in some interval…

17. Just take a look at the p-value for the intercept coefficient when we run summary(out):

```
> summary(out)

Call:
lm(formula = mpg ~ disp + drat + wt)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2342 -2.3719 -0.3148  1.6315  6.2820 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 31.043257   7.099792   4.372 0.000154 ***
disp        -0.016389   0.009578  -1.711 0.098127 .  
drat         0.843965   1.455051   0.580 0.566537    
wt          -3.172482   1.217157  -2.606 0.014495 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.951 on 28 degrees of freedom
Multiple R-squared:  0.7835,    Adjusted R-squared:  0.7603 
F-statistic: 33.78 on 3 and 28 DF,  p-value: 1.92e-09
```

It's pretty small… 0.000154.

18. Since we want to make two CIs, set the individual error rates to 2.5%, so that each individual confidence level is 97.5%.

```
> ptone <- c(200, 3.5, 3.1)
> pttwo <- c(210, 3.75, 3.5)
> pts <- rbind(ptone, pttwo)
> class(pts)
[1] "matrix"
> pts <- data.frame(pts)
> class(pts)
[1] "data.frame"
> names(pts) <- c("disp", "drat", "wt")
> predict(out, new=pts, interval="confidence", level = .975)
         fit      lwr      upr
ptone 20.88461 19.45604 22.31318
pttwo 19.66272 17.82373 21.50170
```

We can be about 95% confident that the mean responses at those two predictor variable settings are between 19.45 and 22.32, and 17.82 and 21.51, respectively. Note again that I've rounded in such a way that the CIs are a little larger (lower end point goes down, and upper one up).

19. We have p=4 parameters.

```
> p=4
> W = sqrt(p * qf(.95, p, nrow(data) - p))
> x <- cbind(c(1,1),pts)
> xx <- as.matrix(x)
> coefs <- as.matrix(coef(out))
> xx
> coefs
> yhhats <- xx %*% coefs
> MSE <- MSE[1,1]
> covmatb <- MSE * solve(t(X) %*% X)
> ssq.yhat.h <- xx %*% covmatb %*% t(xx)
> yhhats[1,1] - W * sqrt(ssq.yhat.h[1,1])
   ptone
18.89724
> yhhats[1,1] + W * sqrt(ssq.yhat.h[1,1])
   ptone
22.87198
>
>
>
> yhhats[2,1] - W * sqrt(ssq.yhat.h[2,2])
   pttwo
17.10439
> yhhats[2,1] + W * sqrt(ssq.yhat.h[2,2])
   pttwo
22.22104
```

20.

```
> predict(out, new=pts, interval = "prediction", level = .975)
          fit      lwr      upr
ptone 20.88461 13.75195 28.01727
pttwo 19.66272 12.43666 26.88877
```