Homework Assignment #5, Allen Baumgarten (still a Dolphins Fan), STAT5120, Spring 2018

1. For each of the following regression models, write down the X matrix and vector. Assume in both cases that there are four observations.

(a) $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i1} X_{i2} + \varepsilon_i$

(b) $\log Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$

1a. $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i1} X_{i2} + \varepsilon_i$ $(n=4)$

$$Y_i = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} \quad X = \begin{bmatrix} 1 & X_{11} & XX_{12} \\ 1 & X_{21} & XX_{22} \\ 1 & X_{31} & XX_{32} \\ 1 & X_{41} & XX_{42} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{bmatrix}$$

1b. $\log Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$ $(n=4)$

$$Y_i = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = X = \log \begin{bmatrix} 1 & X_{11} & X_{12} \\ 1 & X_{21} & X_{22} \\ 1 & X_{31} & X_{32} \\ 1 & X_{41} & X_{42} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{bmatrix}$$

2. For each of the following regression models, write down the X matrix and vector. Assume in both cases that there are five observations.

(a) $Y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i1}^2 + \varepsilon_i$

(b) $\sqrt{Y_i} = \beta_0 + \beta_1 X_{i1} + \beta_2 \log_{10} X_{i2} + \varepsilon_i$

2a. $Y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i1}^2 + \varepsilon_i \quad (n=5)$

$$Y_i = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{bmatrix} = X \begin{bmatrix} 1 & X_{11} & X_{12} \\ 1 & X_{21} & X_{22} \\ 1 & X_{31} & X_{32} \\ 1 & X_{41} & X_{42} \\ 1 & X_{51} & X_{52} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \\ X_{31} \\ X_{41} \\ X_{51} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

2b. $\sqrt{Y_i} = \beta_0 + \beta_1 X_{i1} + \beta_2 \log_{10} X_{i2} + \varepsilon_i$

$$Y_i = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{bmatrix} = \begin{bmatrix} 1 & X_{11} \\ 1 & X_{21} \\ 1 & X_{31} \\ 1 & X_{41} \\ 1 & X_{51} \end{bmatrix} + \begin{bmatrix} \log_{10} & \log_{10} & \log_{10} & \log_{10} & \log_{10} \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \\ X_{31} \\ X_{41} \\ X_{51} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

3. If adding predictor variables to a regression model never reduces R^2, why not just include all the available predictor variables in the model? Also, remark on the meaning of R^2_{adj}.

Dr. Jones addresses this question in his GLM Lecture #7 where he states, "Including additional variables in the model will always increase R^2, which can be a problem because allowing too many variables to enter into the model can lead to overfitting" (95) Overfitting is the condition in which a model so closely 'predicts' each point that it becomes useless for predicting the more generally unknown behaviors found in real-world data, that is, data points not *already* fitted by the model. Imagine a bivariate scatterplot which has five points. Then imagine a curve that weaves through each of those five points with perfect accuracy. Now imagine a sixth new data point being added that was different from the previous five. The model would not 'predict' that new point at all. It would be an overfitted model.

Out class test states that another reason why too many predictor variables may be troublesome is the problem of extrapolation. The authors say, "A large value of R^2 does not necessarily imply that the fitted model is a useful one. For instance, observations may have been taken at only a few levels of the predictor variables. Despite a high R^2 in this case, the fitted model may not be useful it most predictions require extrapolations outside the region of observations" (227).

Here is where the R^2_{adj} comes into its own when we are faced with an ever-increasing R^2: if R^2 can never decrease by adding variables, it would seem that R^2 has some sort of built-in bias for including more variables. Can we "adjust" for such a state of affairs by somehow evening out the playing field between a model with say two variables vs. one with say, three variables? In our class text, the authors write, "...it is sometimes suggested that a modified measure can be used that adjusts for the number of x variables in the model. [It does this] by dividing each sum of squares by its associated degrees of freedom" (226). Other scholars concur: see Chatterjee (68-69) and Pardoe (95ff). Rawlings, et al, (222-23) comment that, "The adjusted R^2...is a rescaling of R^2 by degrees of freedom so that it involves a ratio of mean squares rather than sums of squares..."

4. Recall the simple correlation coefficient *r* is signed. Why is it not meaningful to include a sign on the coefficient of multiple correlation (or coefficient of determination) R^2?
Chatterjee explains that the R^2 coefficient of determination is shown to be SSR/SST = 1 - SSE/SST (68). When we realize that the the ratio of SSE/SST can be at minimum only a very small positive number, but not less than zero, or at maximum a decimal approaching 1.0, the subtraction of that decimal result from 1 results always in a positive number. This number is the amount of variation in the y-variable explained by the inclusion of the one or more x-variables. Appropriately, we see that this variation can never be negative, given the nature of this coefficient.

The data set called *mtcars* is included in the basic R installation and contains information on 1973-74 model automobiles, including miles per gallon, number of cylinders, displacement (cubic inches), gross horsepower, read axle ratio, weight (in lbs/1000), quarter mile time in seconds, v- or straight (V=0, straight = 1), transmission (coded so 0 = automatic and 1 = manual), number of forward gears, and number of carburetors. This data set came from the 1974 Motor Trend magazine. You can see the dataset by simply typing *mtcars* at the prompt:

```
> mtcars
```

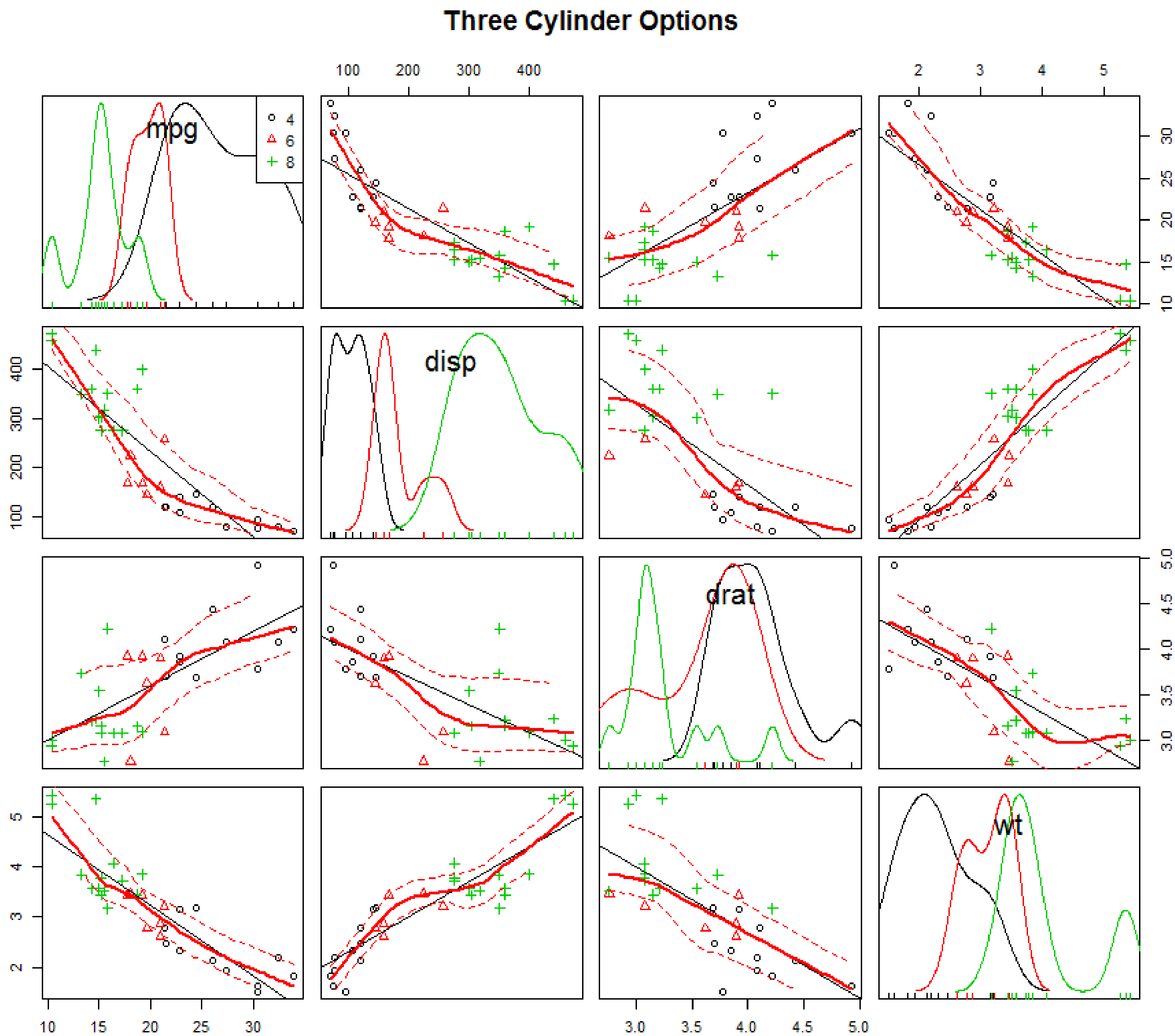
Use this data set to answer the remaining questions. I have included R code to help you.

5. Make a scatterplot matrix of the mpg, dips, drat, and wt variables, indicating the numbers of gears.

The plots along the diagonal are of course not scatterplots (if they were, they would all be straight lines), but show estimated density functions for each variable, color-coded for the number of cylinders. Each of the scatterplots includes a fitted regression line, as well as a lowess-smoothed trend curve (which is a more sophisticated way of estimating trends). There are also rug plots in the cell margins.

Comment generally on a few things that you observe about these five variables based on the scatterplot matrix, specifically about the trends between the quantitative variables and also if the number of cylinders seems to be a factor. Note the legend for the numbers of cylinders is located in one of the plots.

We have below six different combinations of variables, the bottom left scatterplots being reversed mirror images of the scatters in the upper right area. The four probability density curves appear to not be normally distributed, though the drat variable (rear axle ratio) does show some normality. Looking at 'mpg' as the y-variable, it decreases with displacement and weight, and increases with 'drat.' The lowess curves depart from the linear assumption between 'mpg' vs. 'disp,' 'disp' vs. 'drat,' and 'drat' vs. 'wt,' while appearing to be generally straight with 'mpg' (y) vs. the 'drat' and 'wt' variables.



6. Get a sample correlation matrix of the same four quantitative variables from the previous part like this:

Which variable pairs seem most correlated now? Also, explain the magnitude and direction of the correlations.

	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

The 'mpg' variable correlates negatively but strongly with 'disp.' That is, as displacement in cu/in increases, 'mpg' decreases. Most other correlations, in fact, are fairly strong. The weakest correlation is that of 0.681 between 'mpg' and 'drat' (rear axle ratio). We observe, too, that most of the correlation coefficients tend to be negative, indicating that as the x-axis variable increases, the y-axis variable decreases.

7. Fit a GLM that attempts to predict mpg based on disp, wt, and drat. Use a planar model of the form $\hat{mpg} = b_0 + b_1(disp) + b_2(wt) + b_3(drat)$:

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 would appear that the 'wt' (Weight) variable plays a statistically significant role here based on it's small p-value when regressed on 'mpg.' 'disp' and 'drat' appear not so much to.

[Note: statisticians have calculated that the 'wt' p-value is roughly 10 times <u>greater than</u> the probability of the Miami Dolphins NOT trading away valuable players like Jay Ajayi. (Sorry)]

Which (if any) of the three predictor variables seem(s) to be important factors in predicting the mpg?
Based on these preliminary investigations, the 'wt' variable would be most important.

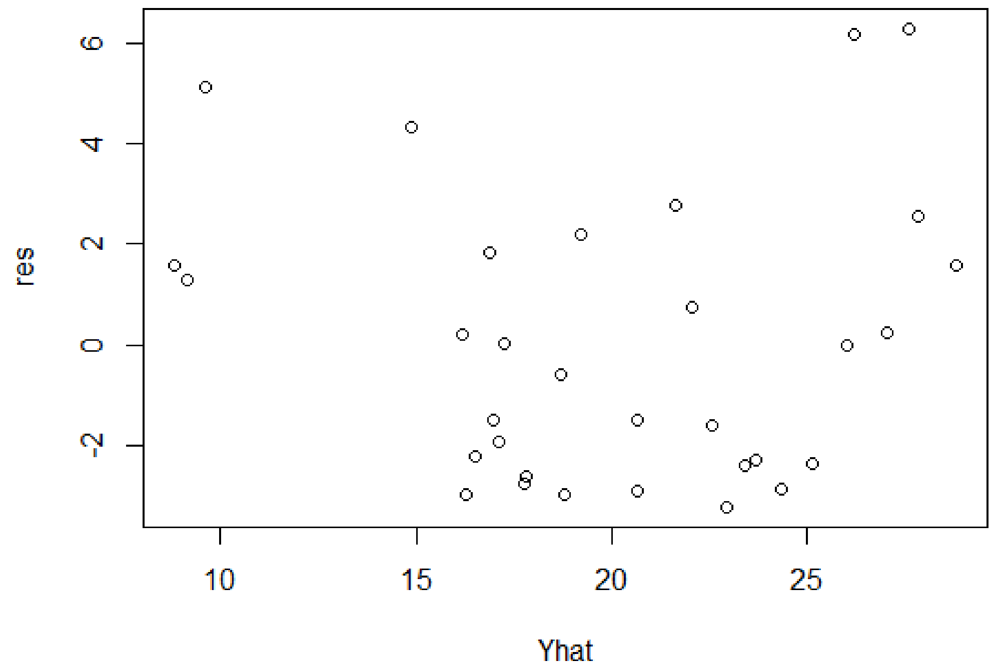
8. Obtain the same GLM as in the previous part, but this time using matrices:

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

etc. Are your model coefficient estimates the same as what you got in the previous part? Given that I am an R neophyte, my calculated estimates were happily equal to those delivered by the lm() function. I had to attach() the mtcars dataset prior to running this.

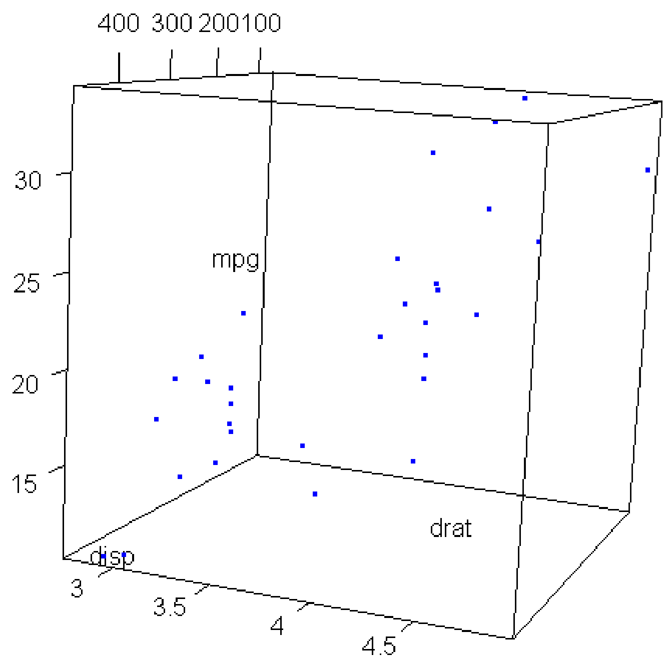
9. Obtain the vector of ts, the hat matrix, and the vector of residuals by using matrix operations in R. Then plot the residuals vs. the fits.

Obtained hat matrix (H), fits, and residuals. Plotted residuals vs. fits below:

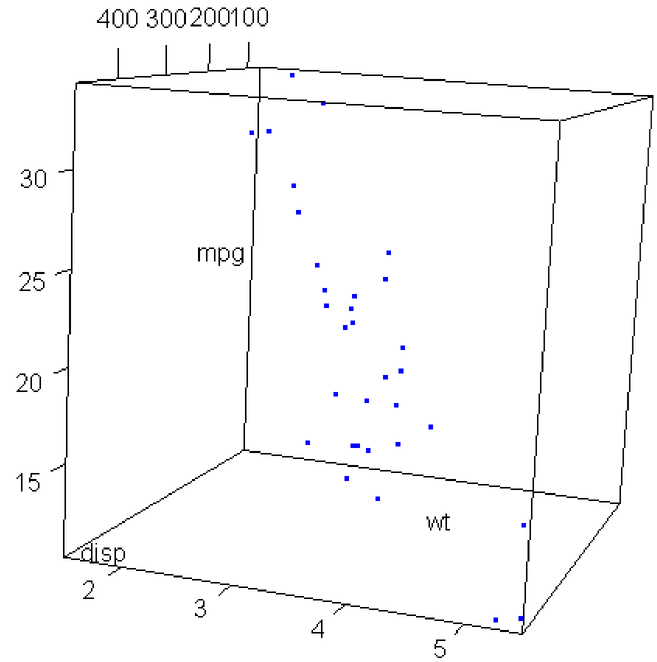


10. Make three, 3D scatterplots. One of them should plot mpg vs. disp and drat; one should plot mpg vs. disp and wt, and a third should plot mpg vs. drat and wt.

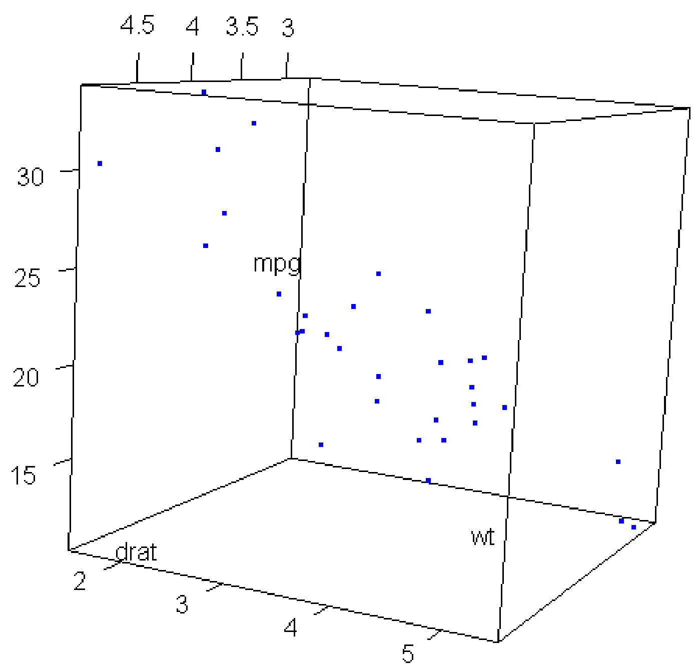
3-D rotation plot of disp, drat, and mpg:



3-D rotation plot of disp, wt, and mpg:



3-D rotation plot of drat, wt, and mpg:



Can you tell from these plots which pair of predictor variables does the best job of predicting mpg? Also, does there seem to be any curvature to the plots? I wonder if a curvilinear model would be better... Regressing 'drat' and 'wt' both seem like good options when reviewing the 3-D scatterplot. They seem tightly clustered and tend to move together with a reasonably straight line. Using 'disp' with 'wt' on the 'mpg' variable of interest, this relationship appears to have some curvature to it, suggesting a curvilinear model. Finally, regressing 'drat' and 'disp' on our variable of interest would seem ill-advised since this 3-D relationship appears to be very loosely related and not very straight at that. We could build a model, to be sure, but the t-values would need to be significant (with significant corresponding p-values).

11. Perform a lack-of-fit F test. Does this test indicate a lack of linear fit? Explain.

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

A lack-of-fit F test was calculated and results show that the model has a statistically significant fit in the disp variable but not the other two. SSE = 243.75 and MSE = 8.71

12. Regardless of your response to the previous question, let's build a polynomial model for mpg vs. the same three quantitative variables (disp, drat, wt)

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

According to the output, what variables and what squared variables seem to be significant? Significant non-squared variables include 'disp' while none of the squared variables appear to have significant p-values. Do some of them (or several of them) suddenly seem less significant than before? Explain. Yes: 'wt' was seen to be signficant in our linear model above but in this model it is not significant. Also, what effect has including the squared terms had on the value of R^2? Does this effect seem intuitive? Explain.

The R2 in our earlier linear model was a modest 0.7835 while in this polynomial model we observe the R2 climbing to 0.8638, albeit with the odd state of affairs of seein our 'wt' variable fall in importance.

13. For this problem, and the remaining ones, refer to the strictly linear model from **problem 8**. Use matrices and R commands to calculate the hat matrix H and use it to get fits. Verify the hat matrix is symmetric and idempotent. Yhat and fits matrices were obtained with R:

```
> head(Yhat)
         [,1]
[1,] 23.40055
[2,] 22.59157
[3,] 25.16234
[4,] 19.21474
[5,] 16.88831
[6,] 18.70825
> head(H)
           [,1]        [,2]        [,3]        [,4]         [,5]       [,6]         [,7]
[1,] 0.04443107  0.041164804 0.048002537 0.022799593  0.020003588 0.01491056  0.019291451
[2,] 0.04116480  0.048964270 0.044793693 0.008729357 -0.010319645 0.01599254 -0.004604134
```

14. Obtain the vector of residuals as well as SSE and MSE. Obtained vectors of residuals and SSE and MSE of original linear model:

```
> head(res)
           [,1]
[1,] -2.4005528
[2,] -1.5915698
[3,] -2.3623354
[4,]  2.1852632
[5,]  1.8116882
[6,] -0.6082518

> SSE
[1] 243.7537

> MSE
[1] 8.70549
```

15. For the model in the previous part, what is the value of R^2? What is the value of R^2_{adj}?
The $R^2 = 0.7835$ and the $R^2_{adj} = 0.7603$

16. Use the confint() function in R to make simultaneous Bonferroni 95% CIs for all four of the parameters (β_0, β_1, β_2, β_3) That is, use 95% as the family error rate.

```
                    2.5 %       97.5 %
(Intercept) 16.49999311  45.586521444
mtcars[, 3] -0.03600944   0.003231128
mtcars[, 5] -2.13657099   3.824501624
mtcars[, 6] -5.66571478  -0.679250218
```

17. Test for evidence that the intercept term differs in a statistically significant way from 0.

Conducted hypothesis test on these three (linear) variables to test if they differed from 0. Outputs shown below; comprehensive R scripts in the appendix (not in my appendix but the paper's)

Test statistic and pvalue for 'disp':

```
> ts.beta_disp
[1] -3.833042
> pval_beta_disp
[1] 0.0006562856
```

Test statistic and pvalue for 'disp':

```
> ts.beta_drat
[1] 0.8515332
> pval_beta_drat
[1] 0.4016976
```

Test statistic and pvalue for 'disp':

```
> ts.beta_wt
[1] -5.857676
> pval_beta_wt
[1] 2.6858e-06
```

18. Use the predict() function in R to make simultaneous Bonferroni 95% CIs for the mean response when the predictors are set to (200, 3.5, 3.1) and (210, 3.75, 3.5). That is, use 95% as the family error rate. The coordinates here are disp, drat, and wt, respectively.

```
> predict(out, new=pts, interval = "confidence")
        fit       lwr      upr
1 23.400553 22.126591 24.67451
2 22.591570 21.254197 23.92894
3 25.162335 23.595257 26.72941
4 19.214737 17.367777 21.06170
5 16.888312 14.547375 19.22925
```

19. **Find 95% Working-Hotelling confidence** intervals for the two points in the previous part.

20. Use the predict() function in R to make 95% simultaneous prediction intervals for new responses based on the predictor levels from the previous two parts.

```
> predict(out, new=pts, interval="prediction", level=.975) # Bonferonni
        fit       lwr      upr
1 23.400553 16.258862 30.54224
2 22.591570 15.434397 29.74874
3 25.162335 17.943121 32.38155
4 19.214737 11.907583 26.52189
5 16.888312  9.394304 24.38232
```

APPENDIX: R SCRIPTS USED (HOPEFULLY CORRECTLY)

Question 5:

Format

A data frame with 32 observations on 11 variables.

[, 1]	mpg	Miles/(US) gallon
[, 2]	cyl	Number of cylinders
[, 3]	disp	Displacement (cu.in.)
[, 4]	hp	Gross horsepower
[, 5]	drat	Rear axle ratio
[, 6]	wt	Weight (1000 lbs)
[, 7]	qsec	1/4 mile time
[, 8]	vs	V/S
[, 9]	am	Transmission (0 = automatic, 1 = manual)
[,10]	gear	Number of forward gears

```
> mtcars
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
```

```
Porsche 914-2        26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa         30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L       15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino         19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora        15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E           21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
> library(car)
Warning message:
package 'car' was built under R version 3.4.3
> scatterplot.matrix(~mpg+disp+drat+wt|cyl,data=mtcars, main="Three Cylinder Options")
Warning message:
'scatterplot.matrix' is deprecated.
Use 'scatterplotMatrix' instead.
See help("Deprecated") and help("car-deprecated").
> scatterplotMatrix(~mpg+disp+drat+wt|cyl,data=mtcars, main="Three Cylinder Options")
```

Question 6:

```
> attach(mtcars)
The following object is masked _by_ .GlobalEnv:

    disp

> d <- data.frame(mpg, disp, drat, wt)
> cor(d)
            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
```

Question 7:

```
> out <- lm(mpg ~ disp + drat + wt)
> 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

Question 8:

```
> ones <- rep(1:1, nrow(mtcars))
> X <- cbind(ones, mtcars$disp, mtcars$drat, mtcars$wt()
```

Question 9:

```
> H <- X %*% solve(t(X) %*% X) %*% t(X)
Y <- as.matrix(mtcars[,1])     ### y variable in mtcars is column 1; be sure sure to pick correct col!
H <- X %*% solve(t(X) %*% X) %*% t(X)     ### hat matrix
Yhat <- H %*% Y     ### Fits
b <- solve(t(X) %*% X) %*% t(X) %*% Y     ### estimates of betas
res <- Y - H %*% Y     ### residuals
> plot(res ~ Yhat)
```

Question 10:

```
> library(car)
> library(rgl)
> open3d()
wgl
  1
> plot3d(disp, drat, mpg, col="blue")
> plot3d(disp, wt, mpg, col="blue")
> plot3d(drat, wt, mpg, col="blue")
```

Question 11:

```
> library(alr3)
> pureErrorAnova(out)
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
```

Question 12:

```
> out2 <- lm(mpg ~ disp + drat + wt + I(disp^2) + I(drat^2) + I(wt^2))
> summary(out2)

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
```

Question 13 and 14 comprehensive R scripts:

```
ones <- rep(1:1, nrow(mtcars))
X <- cbind(ones, mtcars$disp, mtcars$drat, mtcars$wt)
X <- as.matrix(X)
Y <- as.matrix(mtcars[,1])  ### y variable in mtcars is column 1; be sure sure to pick correct col!
H <- X %*% solve(t(X) %*% X) %*% t(X)  ### hat matrix
Yhat <- H %*% Y  ### Fits
b <- solve(t(X) %*% X) %*% t(X) %*% Y  ### estimates of betas
res <- Y - H %*% Y ### residuals
dim(H)  ### gets the dimensions of H
# [1] 32, 32  ### run the dim(H) to get the matrix row-col count
I <- diag(32)  ### plug in the row-col count here in this function
SSE <- sum(res^2)
MSE <- SSE/28  ### this number should be df = n - p's
# names(data) <- c("colname", "colname2")  ### use the names() function if necessary
covest <- MSE * diag(32) - H  ### plug in same number used in the I matrix above
summary(out <- lm(mtcars[,1] ~ mtcars[,3] + mtcars[,5] + mtcars[,6]))
out$fit
out$residual
SSE
MSE
anova(out)
```

Question 14:

```
> SSE
[1] 243.7537
> MSE
[1] 8.70549
> anova(out)
Analysis of Variance Table

Response: mtcars[, 1]
            Df Sum Sq Mean Sq F value    Pr(>F)
mtcars[, 3]  1 808.89  808.89 92.9171 2.152e-10 ***
mtcars[, 5]  1  14.26   14.26  1.6383    0.2111
```

```
mtcars[, 6]  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
```

Question 15:

```
> summary(out)

Call:
lm(formula = mtcars[, 1] ~ mtcars[, 3] + mtcars[, 5] + mtcars[,
    6])

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 ***
mtcars[, 3]  -0.016389   0.009578  -1.711      0.098127 .
mtcars[, 5]   0.843965   1.455051   0.580      0.566537
mtcars[, 6]  -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
```

Question 16:

```
> confint(out, level = 0.95)
                   2.5 %           97.5 %
(Intercept) 16.49999311    45.586521444
mtcars[, 3] -0.03600944     0.003231128
mtcars[, 5] -2.13657099     3.824501624
mtcars[, 6] -5.66571478    -0.679250218
```

Question 17:

```
> sd.beta_disp <- sqrt(MSE/sum((mtcars$disp-mean(mtcars$disp))^2))
> hyp_beta_disp <- 0
> ts.beta_disp <- (-0.016389 - hyp_beta_disp)/sd.beta_disp
> pval_beta_disp <- 2*pt(-abs((-0.016389-hyp_beta_disp)/sd.beta_disp),28)

> sd.beta_drat <- sqrt(MSE/sum((mtcars$drat-mean(mtcars$drat))^2))
> hyp_beta_drat <- 0
> ts.beta_drat <- (0.843965 - hyp_beta_drat)/sd.beta_drat
> pval_beta_drat <- 2*pt(-abs((0.843965-hyp_beta_drat)/sd.beta_drat),28)

> sd.beta_wt <- sqrt(MSE/sum((mtcars$wt-mean(mtcars$wt))^2))
> hyp_beta_wt <- 0
> ts.beta_wt <- (-3.172482 - hyp_beta_wt)/sd.beta_wt
> pval_beta_wt <- 2*pt(-abs((-3.172482-hyp_beta_wt)/sd.beta_wt),28)
```

Question 18:

```
> ptone <- c(5,5,100)
> pttwo <- c(10,10,100)
> rbind(ptone,pttwo)
      [,1] [,2] [,3]
ptone    5    5  100
pttwo   10   10  100
> pts <- rbind(ptone,pttwo)
> class(pts)
[1] "matrix"
> pts <- data.frame(pts)
> class(pts)
[1] "data.frame"
> predict(out, new=pts, interval = "confidence")
        fit       lwr      upr
1  23.400553 22.126591 24.67451
2  22.591570 21.254197 23.92894
3  25.162335 23.595257 26.72941
4  19.214737 17.367777 21.06170
5  16.888312 14.547375 19.22925
```

Question 20:

```
> predict(out, new=pts, interval="prediction", level=.975) # Bonferonni
        fit       lwr      upr
1  23.400553 16.258862 30.54224
2  22.591570 15.434397 29.74874
3  25.162335 17.943121 32.38155
4  19.214737 11.907583 26.52189
5  16.888312  9.394304 24.38232
```