Homework 01 Solutions

2022-09-02

36-617: Applied Linear Models
Fall 2022
Solutions

```
library(arm)  ## includes lme4, MASS, Matrix
library(ggplot2); theme_set(theme_bw())
library(gridExtra) ## to arrange ggplots...
```

Problem 1: ISLR second Ed. p. 123 #8.

```
## You can use
##
##           getCRANmirrors(all = FALSE, local.only = FALSE)
##
## to get a listing of all the CRAN mirrors from which you can get
## packages to install...

## install.packages("ISLR2",repos="https://cran.case.edu/")  # only have to do this once...

library(ISLR2)  ## do this once in every R session where you want to use this library.
```

```
## Warning: package 'ISLR2' was built under R version 4.1.3
```

1(a)

Use the `lm()` function to perform a simple linear regression with `mpg` as the response and `horsepower` as the predictor. Use the `summary()` function to print the results. Comment on the output. For example:

i. Is there a relationship between the predictor and the response?

ii. How strong is the relationship between the predictor and the response?

iii. Is the relationship between the predictor and the response positive or negative?

iv. What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals?

```
data(Auto)
str(Auto) ## get a quick "look" at the data...
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : int  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
```

```
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 1
##  - attr(*, "na.action")= 'omit' Named int [1:5] 33 127 331 337 355
##   ..- attr(*, "names")= chr [1:5] "33" "127" "331" "337" ...
```

```
lm.0 <- lm(mpg ~ horsepower,data=Auto)

summary(lm.0)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

The estimated coefficient on `horsepower`, $\hat{\beta}_1$, *is highly significantly different from zero (p-value* $< 2 \times 10^{-16}$ *is very small!!), and the* R^2 *value is moderately high,* $R^2 = 0.6059$, *so there appears to be a moderately strong relationship between the variables. Since* $\hat{\beta}_1 = -0.157845 < 0$, *the relationship is negative, i.e.* `mpg` *appears to decrease as* `horsepower` *increases.*

Using the code

```
new.data <- data.frame(mpg=0,horsepower=98)

predict(lm.0,newdata=new.data,interval="confidence")
```

```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```
predict(lm.0,newdata=new.data,interval="prediction")
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

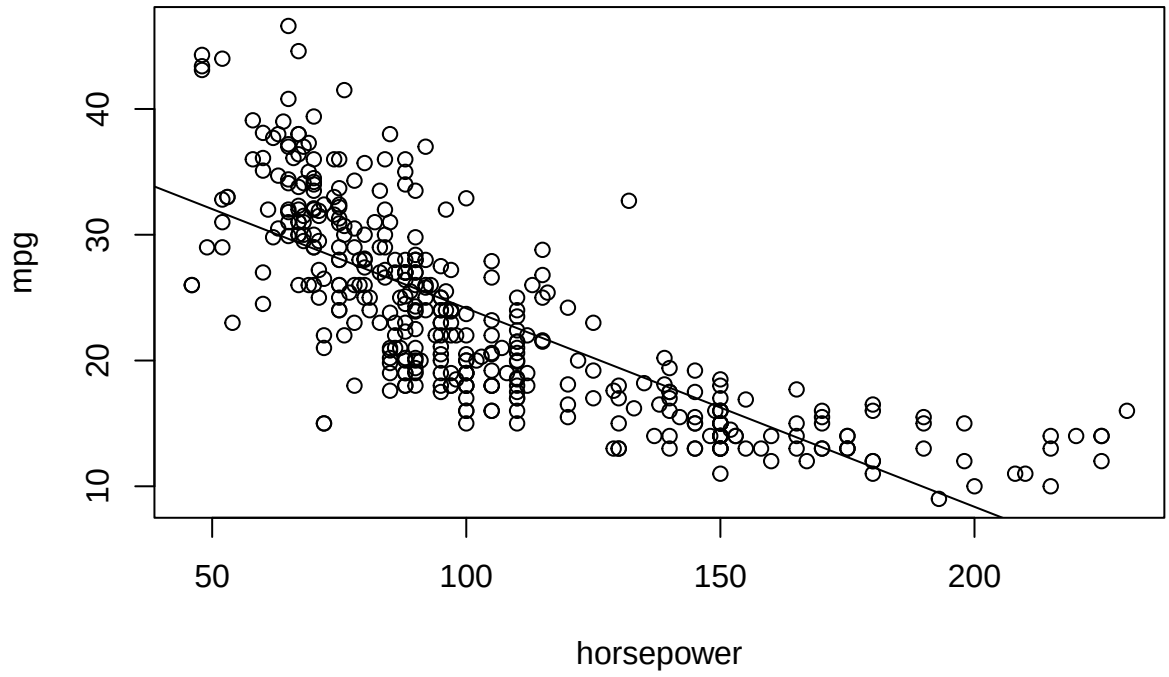
we see that when `horsepower` $= 98$,

- *The fitted* `mpg`, $\hat{y}$, *is 24.46708*
- *The 95% CI is* $(23.97308, 24.96108)$
- *The 95% PI is* $(14.8094, 34.12476)$

1(b)

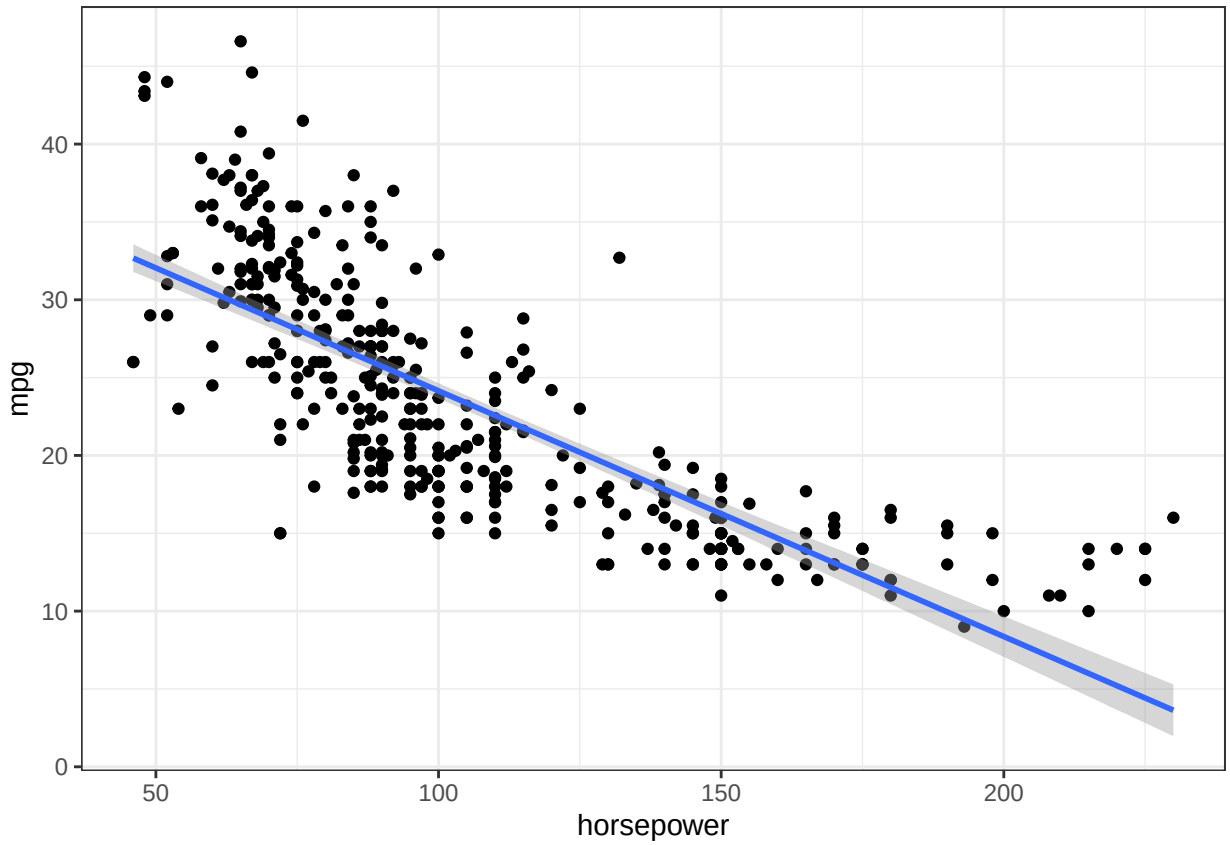
Plot the response and the predictor. Use the `abline()` function to display the least squares regression line.

```
plot(mpg ~ horsepower,data=Auto)
abline(lm.0)
```



Note that you can get a "prettier" plot with ggplot...

```
ggplot(data=Auto,aes(x=horsepower,y=mpg)) +
  geom_point() +
  geom_smooth(method='lm')
```



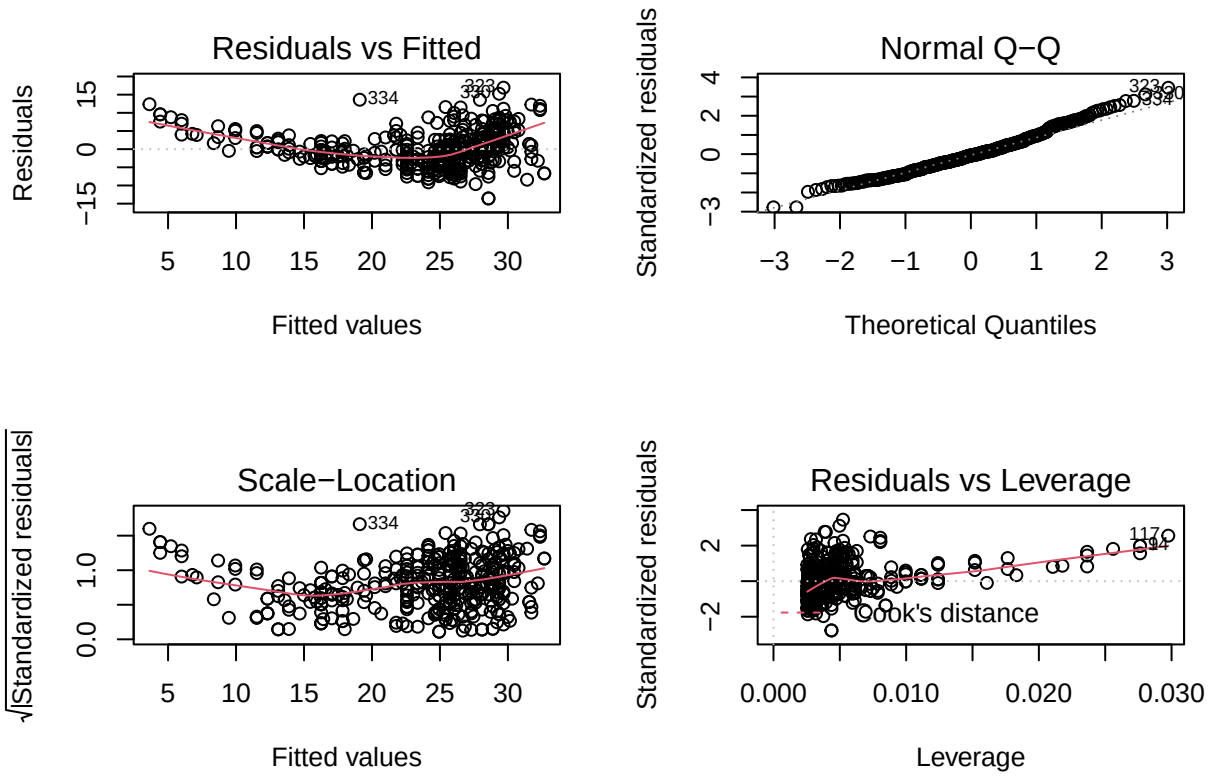
(The shaded region around the regression line in the ggplot version is made by computing the 95% confidence interval for `mpg`*,* $\hat{y}$*, for every* `horsepower`*,* x*, in the figure. If you don't want that in the figure, you can specify* `se=FALSE` *as another argument to the* `geom_smooth` *function.)*

It is always good to look at the data and compare it to the fitted model somehow. In this case, the plots confirm that there is a reasonably strong negative relationship between `horsepower` *and* `mpg`*, but it does not in fact appear to be a linear relationship.*

1(c)

Use the `plot()` function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

```
par(mfrow=c(2,2))

plot(lm.0)
```



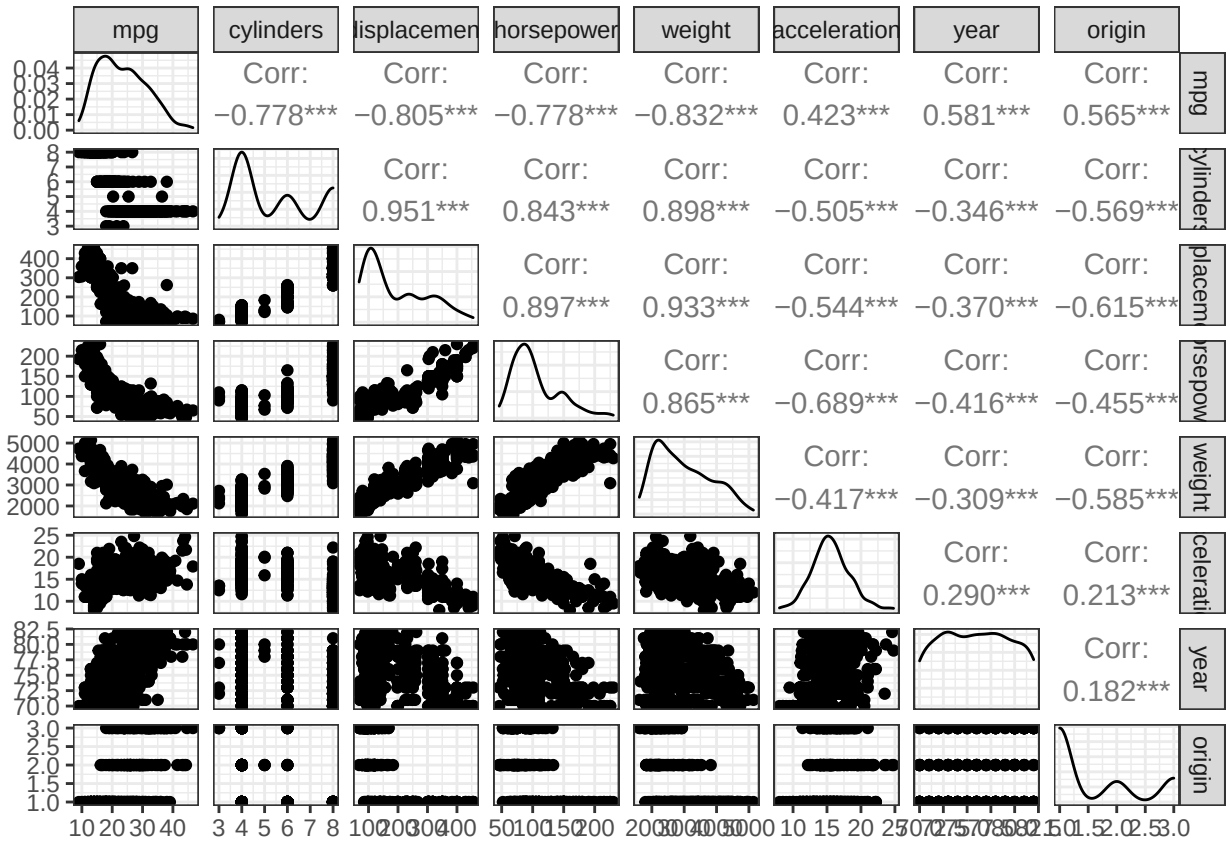
- *In the Residual vs. Fitted plot we see that the data cloud of residuals is curved, reflecting the curvature in the raw data in the plot we made before. We also see some fanning as $\hat{y}$ increases, suggesting perhaps that the error variance is not constant across all observations.*
- *The normal qq plot looks reasonably good, but we might like to see a blown-up version of it to investigate possible outliers (obs 334 looks like an outlier in the residual vs. fitted plot for example).*
- *The scale-location plot is exhibiting some of the same curvature as the residual vs. fitted plot, and the nonparametric smooth (the red like) suggests perhaps some nonconstant variance. That's not unusual: if the functional form is wrong, it sometimes manifests itself as non-constant variance (since one part of the true nonlinear relationship may be better fitted with a straight line than some other part).*
- *The Residuals vs. Leverage plot does not show any particularly bad points. Although there are some data point in the NE corner of the plot, the residual is still modest so the leverage doesn't come much into play. No data points have Cook's Distance above 0.5 (in fact, the level curve for Cook's distance = 0.50 is outside the plotting range of the graph).*

Problem 2. ISLR second Ed. p. 123–124, #9.

2(a)

Produce a scatterplot matrix which includes all of the variables in the data set.

```
## Note that I should remove the column "name" before making the scatterplot matrix,
## because it is not a numeric variable.

library(GGally)
ggpairs(Auto[,-grep("name",names(Auto))])
```




```
## you can also make a simpler plot with
##
##    pairs(Auto)
##
```

Note that this plot also gives us a nonparametric marginal density estimate (i.e. a smooth histogram) for each variable, the correlations between the variables, and a crude set of p-values for the test of whether each correlation is significantly different from zero.

2(b)

Compute the matrix of correlations between the variables using the function `cor()`. You will need to exclude the name variable, `cor()` which is qualitative.

The correlations are already calculated above, but let's do it again with the `cor()` *function.*

```
options(width=999)
round(cor(Auto[,-grep("name",names(Auto))]),2)
```

```
##                mpg cylinders displacement horsepower weight acceleration  year origin
## mpg           1.00     -0.78        -0.81      -0.78  -0.83         0.42  0.58   0.57
## cylinders    -0.78      1.00         0.95       0.84   0.90        -0.50 -0.35  -0.57
## displacement -0.81      0.95         1.00       0.90   0.93        -0.54 -0.37  -0.61
## horsepower   -0.78      0.84         0.90       1.00   0.86        -0.69 -0.42  -0.46
## weight       -0.83      0.90         0.93       0.86   1.00        -0.42 -0.31  -0.59
## acceleration  0.42     -0.50        -0.54      -0.69  -0.42         1.00  0.29   0.21
## year          0.58     -0.35        -0.37      -0.42  -0.31         0.29  1.00   0.18
## origin        0.57     -0.57        -0.61      -0.46  -0.59         0.21  0.18   1.00
```

2(c)

Use the `lm()` function to perform a multiple linear regression with `mpg` as the response and all other variables except name as the predictors. Use the `summary()` function to print the results. Comment on the output. For instance:

i. Is there a relationship between the predictors and the response?

ii. Which predictors appear to have a statistically significant relationship to the response?

iii. What does the coefficient for the year variable suggest?

```
Nameless_Auto <- Auto[,-grep("name",names(Auto))]

lm.1 <- lm(mpg ~ . , data=Nameless_Auto)
## Note the use of "." to stand for "all the variables in the data frame
## except for the y-variable (mpg, in this case)"...

summary(lm.1)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = Nameless_Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

There appears to be a relationship between the predictors and the response, `mpg`*: 4 of the 7 predictors have coefficients that appear to be significantly different from zero, and the* R^2 *is a more respectable 0.8215.*

The four predictors with an apparently statistically significant relationship with the response are:

- `displacement`
- `weight`
- `year`
- `origin`

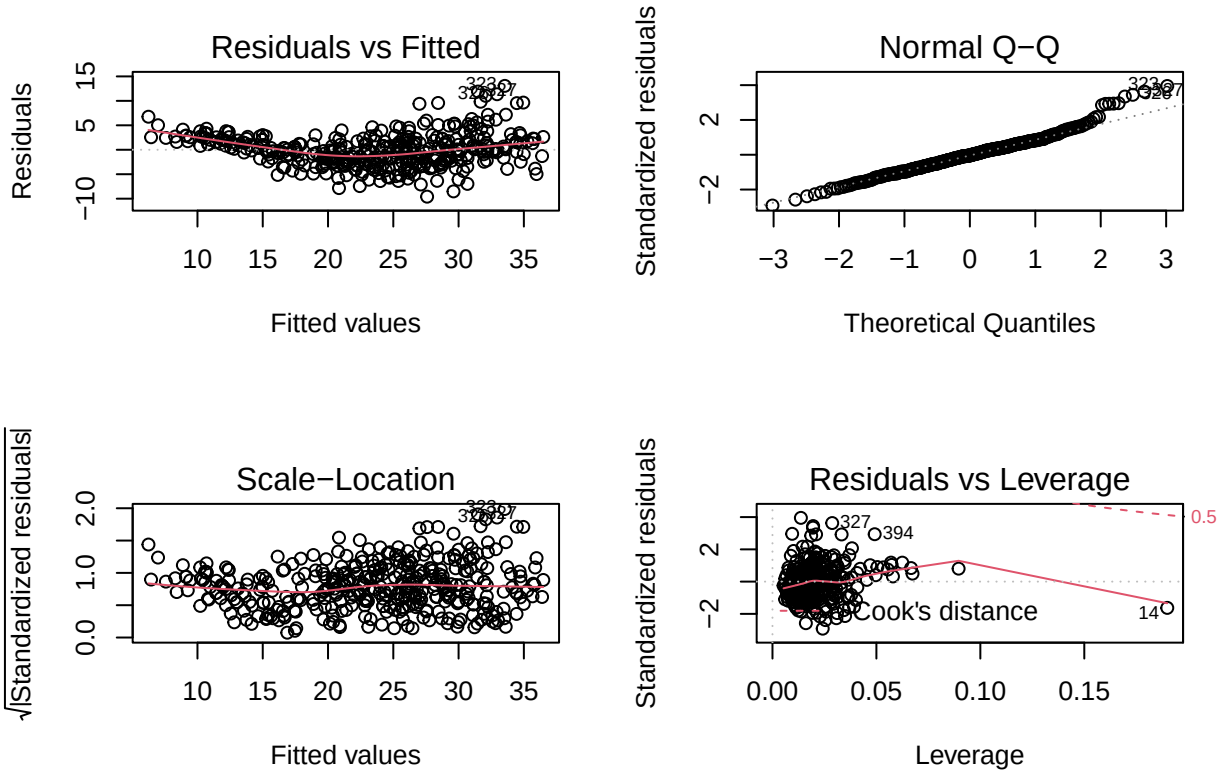
The coefficient estimate 0.750773 for `year` *is significantly different from zero, so it makes sense to try to interpret it:*

The value 0.750773 suggests that the average miles per gallon (`mpg`) for cars has been going up by about 3/4 mile per gallon per year, over the years covered by this data set.

2(d)

Use the `plot()` function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```
par(mfrow=c(2,2))

plot(lm.1)
```



- *The Residual vs. Fitted plot shows that the data cloud of residuals is still curved, suggesting that there is some nonlinearity that simply adding more different variables to the regression did not fix. There is still some fanning out of the data as $\hat{y}$ increases, suggesting non-constant error variance. At least obs. #334 no longer looks like an outlier.*
- *The normal qq plot looks reasonably good, much as it did before. There's some suggestion thsat the upper tail might be a bit long, but nothing really concerning.*
- *The scale-location plot still has some curvature in it, again suggesting some nonlinearity that we haven't fixed in our model, but the nonparametric smooth is fairly flat now, so we might not worry too much about the fanning we saw in the residuals vs. fitted plot.*
- *The Residuals vs. Leverage plot looks fairly good. There is one data point #14, with relatively high leverage, but a very modest residual so it probably is not affecting the fitted regression function very much. Although one of the level curves for Cook's Distance = 0.50 appears in the plot, there are no points with Cook's Distance above 0.5.*

2(e)

Use the `*` and `:` symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

We will learn about / review variable selection methods soon, but for now it suffices to fit a bunch of interactions and see which ones have significant t-statistics. Here are a few useful things to know about the notation, before we begin fitting things:

- *We put the* two-way interaction*, or product, x1×x2 in the model with the model notation* `x1:x2`
- *We almost always want to include all lower-order terms with an interaction (this is called the "hierarchy principle"), so it is better to say* `x1*x2`*, which expands to* `1 + x1 + x2 + x1:x2`
- *If we try this with three variables, we get a* three-way interaction*, with all lower order terms:* `x1*x2*x3` *expands to* `1 + x1 + x2 + x3 + x1:x2 + x1:x3 + x2:x3 + x1:x2:x3`*.*
- *The notation* `(x1+x2+x3)^3` *is equivalent to* `x1*x2*x3`*. The notation* `(x1+x2+x3)^2` *is equivalent to* `x1*x2 + x1*x3 + x2*x3`*, which is in turn equivalent to* `1 + x1 + x2 + x3 + x1:x2 + x1:x3 + x2:x3` *(i.e., all the two-way interactions, along with the corresponding lower-order terms.)*
- *Recall that* `.` *represents the additive model with all variables in the data frame except for the y-variable. Hence* `.^2` *is the model with all 2-way interactions (and all lower order terms),* `.^3` *is the model with all 3-way interactions (and all lower order terms), and similarly with* `.^4`*,* `.^5`*, etc.*
- *It is rare to see 4-way interactions in real data, and 3-way interactions are fairly uncommon. Often, it suffices to look at main effects and 2-way interactions.*

Let's fit

```
lm.2 <- lm(mpg ~ .^2, data=Nameless_Auto)

lm.3 <- lm(mpg ~ .^3, data=Nameless_Auto)
```

From the summary for `lm.2`

```
summary(lm.2)
```

```
##
## Call:
## lm(formula = mpg ~ .^2, data = Nameless_Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6303 -1.4481  0.0596  1.2739 11.1386
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 3.548e+01  5.314e+01   0.668  0.50475
## cylinders                   6.989e+00  8.248e+00   0.847  0.39738
## displacement               -4.785e-01  1.894e-01  -2.527  0.01192 *
## horsepower                  5.034e-01  3.470e-01   1.451  0.14769
## weight                      4.133e-03  1.759e-02   0.235  0.81442
## acceleration               -5.859e+00  2.174e+00  -2.696  0.00735 **
## year                        6.974e-01  6.097e-01   1.144  0.25340
## origin                     -2.090e+01  7.097e+00  -2.944  0.00345 **
## cylinders:displacement     -3.383e-03  6.455e-03  -0.524  0.60051
## cylinders:horsepower        1.161e-02  2.420e-02   0.480  0.63157
## cylinders:weight            3.575e-04  8.955e-04   0.399  0.69000
## cylinders:acceleration      2.779e-01  1.664e-01   1.670  0.09584 .
```

```
## cylinders:year            -1.741e-01  9.714e-02  -1.793  0.07389 .
## cylinders:origin           4.022e-01  4.926e-01   0.816  0.41482
## displacement:horsepower   -8.491e-05  2.885e-04  -0.294  0.76867
## displacement:weight        2.472e-05  1.470e-05   1.682  0.09342 .
## displacement:acceleration -3.479e-03  3.342e-03  -1.041  0.29853
## displacement:year          5.934e-03  2.391e-03   2.482  0.01352 *
## displacement:origin        2.398e-02  1.947e-02   1.232  0.21875
## horsepower:weight         -1.968e-05  2.924e-05  -0.673  0.50124
## horsepower:acceleration   -7.213e-03  3.719e-03  -1.939  0.05325 .
## horsepower:year           -5.838e-03  3.938e-03  -1.482  0.13916
## horsepower:origin          2.233e-03  2.930e-02   0.076  0.93931
## weight:acceleration        2.346e-04  2.289e-04   1.025  0.30596
## weight:year               -2.245e-04  2.127e-04  -1.056  0.29182
## weight:origin             -5.789e-04  1.591e-03  -0.364  0.71623
## acceleration:year          5.562e-02  2.558e-02   2.174  0.03033 *
## acceleration:origin        4.583e-01  1.567e-01   2.926  0.00365 **
## year:origin                1.393e-01  7.399e-02   1.882  0.06062 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.695 on 363 degrees of freedom
## Multiple R-squared:  0.8893, Adjusted R-squared:  0.8808
## F-statistic: 104.2 on 28 and 363 DF,  p-value: < 2.2e-16
```

we see (ignoring predictors that are only significant at the 0.10 level):

- `displacement`, `accelleration` *and* `origin` *are all significant main effects.*
- `displacement:year`, `accelleration:year` *and* `accelleration:origin` *are all significant 2-way interactions.*

So initially the model we would think of would be

```
mpg ~ displacement + accelleration + origin + displacement:year + accelleration:year +
        accelleration:origin
```

but, following the "hierarchy principle" that any interaction should bring with it all of its lower-order interactions, we should really be looking at

```
mpg ~ displacement + accelleration + origin + year + displacement:year +
        accelleration:year + accelleration:origin
```

or, more succinctly,

```
mpg ~ displacement*year + accelleration*year + accelleration*origin              (1)
```

From the summary for `lm.3`

```
summary(lm.3)
```

```
##
## Call:
## lm(formula = mpg ~ .^3, data = Nameless_Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3263 -1.3084 -0.0916  1.2427 10.3972
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                          -2.151e+02  4.506e+02  -0.477 0.633414
## cylinders                             3.804e+01  1.132e+02   0.336 0.737101
## displacement                         -2.159e-01  2.595e+00  -0.083 0.933729
## horsepower                           -5.629e+00  3.435e+00  -1.638 0.102305
## weight                                3.354e-01  2.376e-01   1.412 0.159037
## acceleration                         -1.253e+01  1.937e+01  -0.647 0.518227
## year                                  5.025e+00  5.202e+00   0.966 0.334697
## origin                               -9.470e+01  9.183e+01  -1.031 0.303204
## cylinders:displacement               -8.647e-03  1.680e-01  -0.051 0.958977
## cylinders:horsepower                 -6.153e-01  6.333e-01  -0.971 0.332017
## cylinders:weight                      1.888e-02  2.613e-02   0.723 0.470493
## cylinders:acceleration               -9.031e-01  5.191e+00  -0.174 0.861998
## cylinders:year                       -6.801e-01  1.290e+00  -0.527 0.598503
## cylinders:origin                      8.173e+00  1.591e+01   0.514 0.607746
## displacement:horsepower               1.929e-02  9.712e-03   1.986 0.047894 *
## displacement:weight                  -5.486e-04  4.648e-04  -1.180 0.238746
## displacement:acceleration            -5.727e-02  1.038e-01  -0.552 0.581427
## displacement:year                     1.027e-03  3.065e-02   0.034 0.973292
## displacement:origin                   5.925e-01  5.692e-01   1.041 0.298704
## horsepower:weight                    -5.419e-04  1.008e-03  -0.537 0.591308
## horsepower:acceleration               3.994e-01  9.552e-02   4.181 3.73e-05 ***
## horsepower:year                       4.337e-02  4.213e-02   1.030 0.303972
## horsepower:origin                     2.643e+00  7.805e-01   3.386 0.000795 ***
## weight:acceleration                  -8.123e-03  8.288e-03  -0.980 0.327759
## weight:year                          -4.186e-03  2.823e-03  -1.483 0.139001
## weight:origin                        -1.260e-01  4.344e-02  -2.901 0.003965 **
## acceleration:year                     1.348e-01  2.307e-01   0.584 0.559511
## acceleration:origin                   7.782e+00  3.860e+00   2.016 0.044604 *
## year:origin                           3.895e-01  1.024e+00   0.380 0.703859
## cylinders:displacement:horsepower     1.047e-04  3.498e-04   0.299 0.764857
## cylinders:displacement:weight        -3.581e-06  7.703e-06  -0.465 0.642339
## cylinders:displacement:acceleration   2.952e-03  2.793e-03   1.057 0.291447
## cylinders:displacement:year           3.492e-04  2.309e-03   0.151 0.879857
## cylinders:displacement:origin        -6.383e-02  2.110e-02  -3.025 0.002684 **
## cylinders:horsepower:weight          -1.572e-05  4.080e-05  -0.385 0.700276
## cylinders:horsepower:acceleration    -1.197e-02  7.002e-03  -1.710 0.088301 .
## cylinders:horsepower:year             1.231e-02  8.431e-03   1.461 0.145102
## cylinders:horsepower:origin          -5.544e-02  5.930e-02  -0.935 0.350541
## cylinders:weight:acceleration         2.805e-04  3.855e-04   0.728 0.467370
## cylinders:weight:year                -3.314e-04  3.417e-04  -0.970 0.332837
## cylinders:weight:origin               2.992e-03  2.797e-03   1.070 0.285530
## cylinders:acceleration:year           1.158e-02  6.050e-02   0.191 0.848333
## cylinders:acceleration:origin         7.960e-02  4.358e-01   0.183 0.855180
## cylinders:year:origin                -3.052e-02  1.636e-01  -0.187 0.852118
## displacement:horsepower:weight        8.808e-08  3.359e-07   0.262 0.793332
## displacement:horsepower:acceleration  3.820e-05  1.079e-04   0.354 0.723541
## displacement:horsepower:year         -2.941e-04  1.242e-04  -2.368 0.018460 *
## displacement:horsepower:origin        3.409e-04  1.859e-03   0.183 0.854575
## displacement:weight:acceleration     -3.748e-06  6.354e-06  -0.590 0.555689
## displacement:weight:year              8.862e-06  6.045e-06   1.466 0.143610
## displacement:weight:origin           -1.919e-05  4.168e-05  -0.460 0.645500
## displacement:acceleration:year        6.991e-04  1.236e-03   0.566 0.571951
## displacement:acceleration:origin     -1.184e-02  1.288e-02  -0.919 0.358578
## displacement:year:origin             -2.183e-03  6.107e-03  -0.357 0.720972
```

```
## horsepower:weight:acceleration        2.583e-06  1.158e-05   0.223 0.823662
## horsepower:weight:year                9.295e-06  1.195e-05   0.778 0.437395
## horsepower:weight:origin             -6.136e-05  9.824e-05  -0.625 0.532667
## horsepower:acceleration:year         -3.915e-03  1.154e-03  -3.394 0.000774 ***
## horsepower:acceleration:origin       -4.246e-02  1.074e-02  -3.952 9.49e-05 ***
## horsepower:year:origin               -2.164e-02  9.107e-03  -2.376 0.018086 *
## weight:acceleration:year              7.719e-05  9.701e-05   0.796 0.426779
## weight:acceleration:origin            1.143e-03  7.779e-04   1.470 0.142583
## weight:year:origin                    1.369e-03  4.798e-04   2.854 0.004589 **
## acceleration:year:origin             -8.235e-02  4.289e-02  -1.920 0.055720 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.441 on 328 degrees of freedom
## Multiple R-squared:  0.9179, Adjusted R-squared:  0.9022
## F-statistic: 58.22 on 63 and 328 DF,  p-value: < 2.2e-16
```

we see

- *No significant main effects*
- `displacement:horsepower`, `horsepower:accelleration`, `horsepower:year`, `weight:origin`, *and* `accelleration:origin` *are significant two-way interactions*
- `cylinder:horsepower:weight`, `displacement:horsepower:year`, `horsepower:accelleration:year`, `horsepower:accelleration:origin`, *and* `weight:year:origin` *are all significant three-way interactions.*

which we could write (using `*` *to make sure we are following the "hierarchy principle") as*

```
mpg ~ cylinder*horsepower*weight + displacement*horsepower*year +
        horsepower*accelleration*year + horsepower*accelleration*origin +
        weight*year*origin                                                    (2)
```

Note that all the two-way interactions listed above (and a few more) are included in this model, and all the main effects `cylinder`, `horsepower`, `weight`, `displacement`, `year`, *and* `accelleration` *are also included in the model (even though they weren't by themselves significant).*

Just for kicks, I also looked at

```
summary(lm(mpg ~ .^4, data=Nameless_Auto))
```

and it turns out that none of the 4-way interactions appear to be significant, so we will not pursue this, or any higher-order models, any further.

Note that models (1) and (2) are rather different. This is in part because interactions share some variability with their lower order terms, and the t-statistics measure the importance of a variable or interaction after *all other variables and interactions have been included in the model. If the t-statistic is for a variable that has similar variability to something that is already in the model, then the t-statistic will undervalue its variable.*

In general, t-statistics are good for an initial quick skim of the variables in a model, but they do not say anything about overall fit, predictive power, etc., of the model.

2(f)

Try a few different transformations of the variables, such as `log(X)`, `sqrt(X)`, `X^2`, etc. Comment on your findings.

We haven't talked about a principled way to proceed with this in class yet (though you may remember some principled approaches from your earlier work with regression models!). For now we will just try to build a

little on the little bit of EDA we did in problem 2(a).

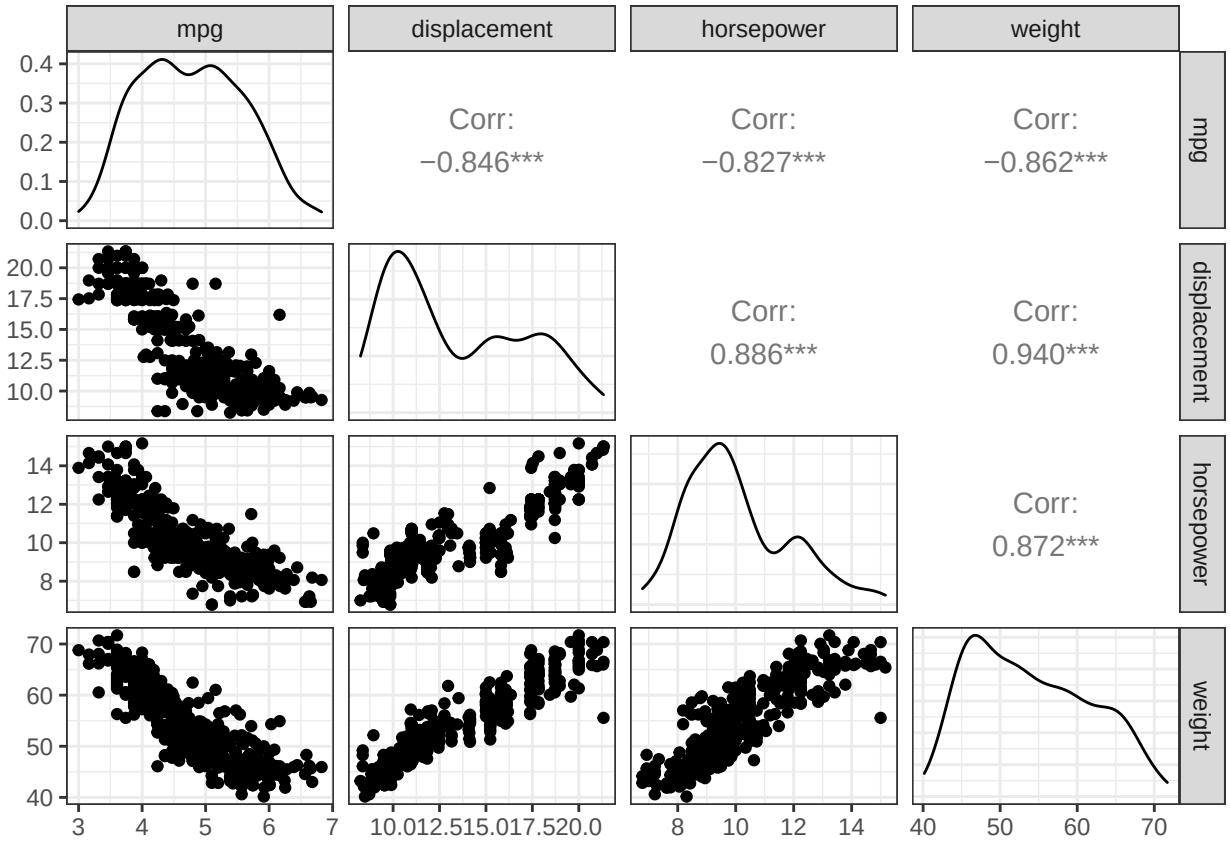
I will try two different ideas to guide transformations here:

- *Remove asymmetry (skewing) in all variables that exhibit it, and see if that improves the casewise diagnostic plots.*
- *Leave the y variable,* `mpg`*, alone, and try to transform some of the X variables so that the distribution of $\hat{y}$ more closely follows the distribution of y.*

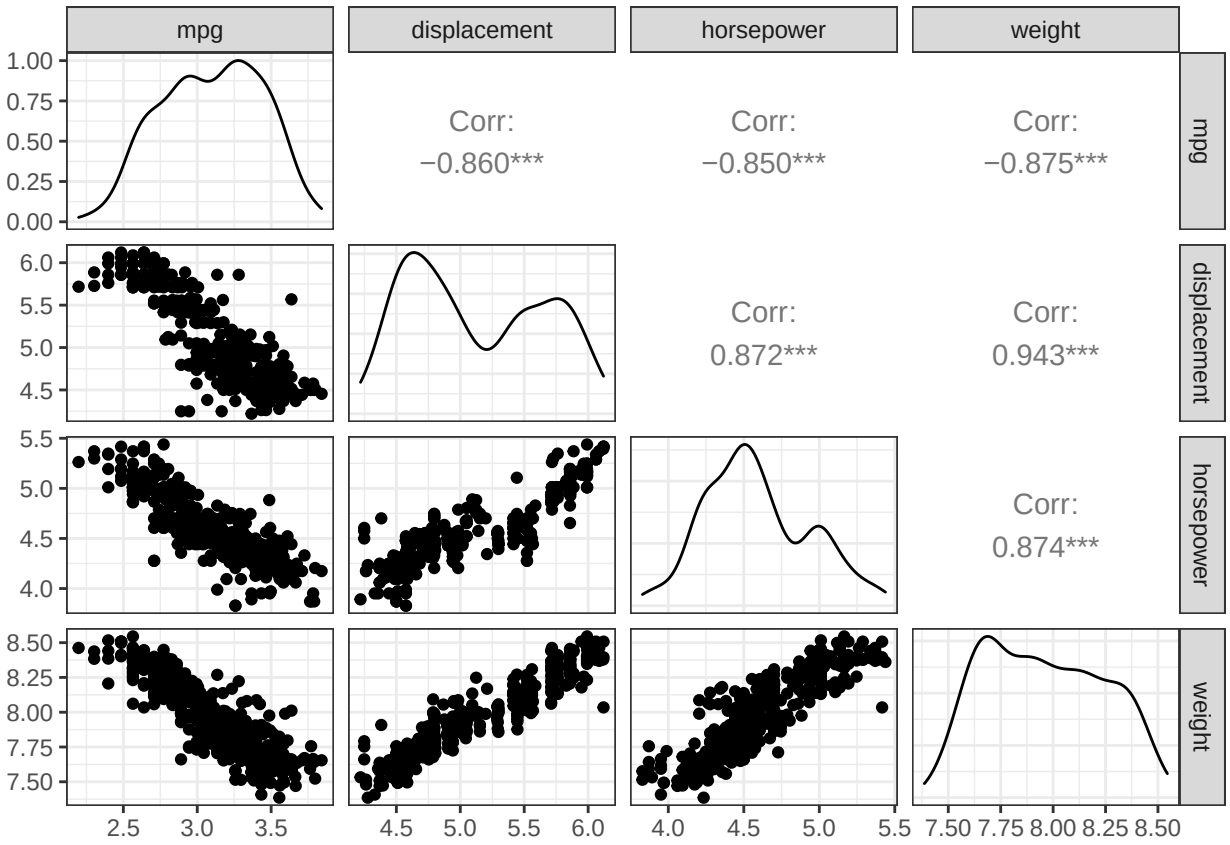
Here's the first idea in action:

Scanning the density plots from problem 2(a), it looks like `mpg`*,* `displacement`*,* `horsepower` *and* `weight` *are the continuous variables with least symmetric distributions, and all of them are right skewed. (The other variables are either not very skewed, or are discrete and so transformations are not appropriate.) We'll quickly try* `log()` *and* `sqrt()` *on each of them to see what improves their distributions more...*

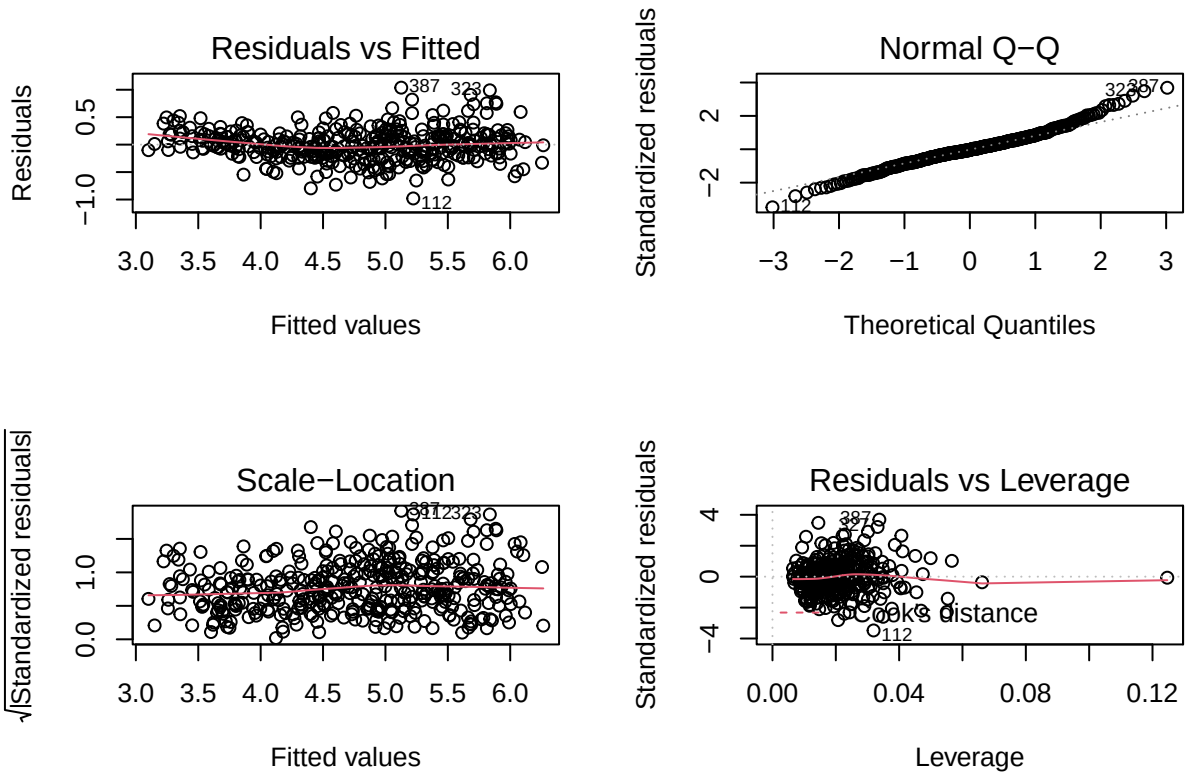
```
ggpairs(sqrt(Nameless_Auto[,c("mpg","displacement","horsepower","weight")]))
```



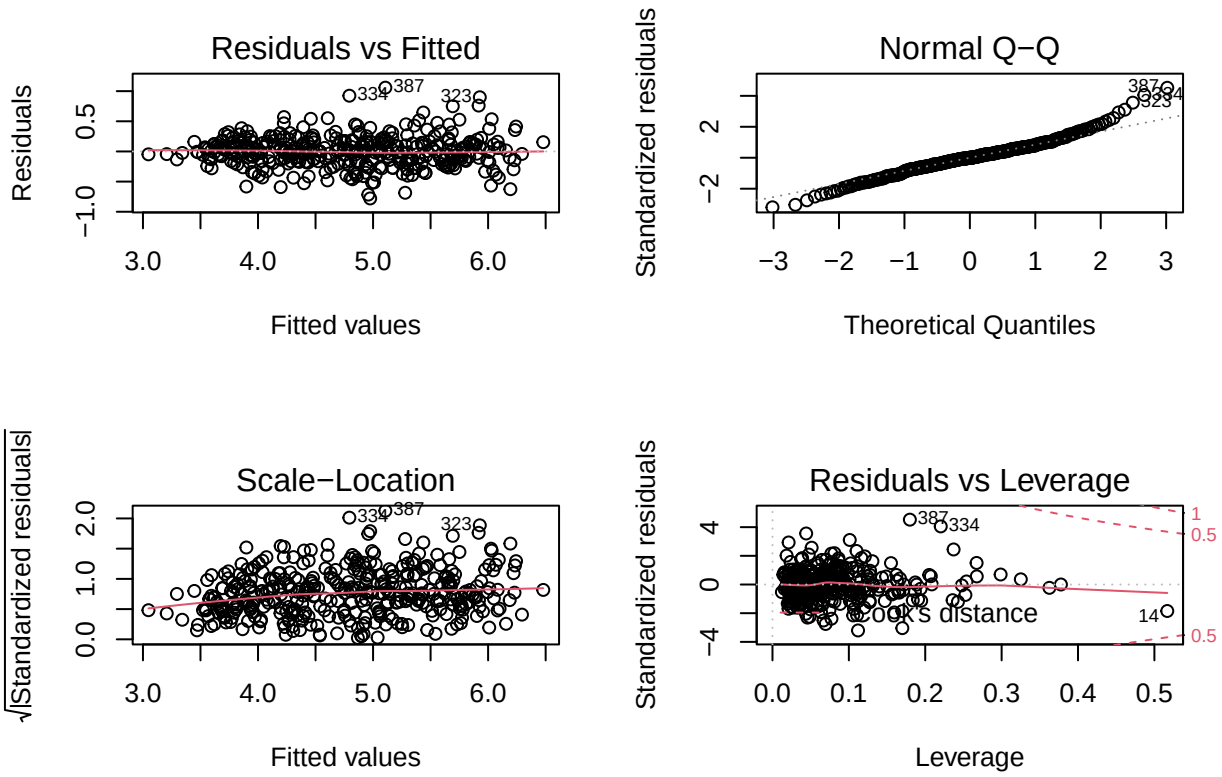

```
ggpairs(log(Nameless_Auto[,c("mpg","displacement","horsepower","weight")]))
```



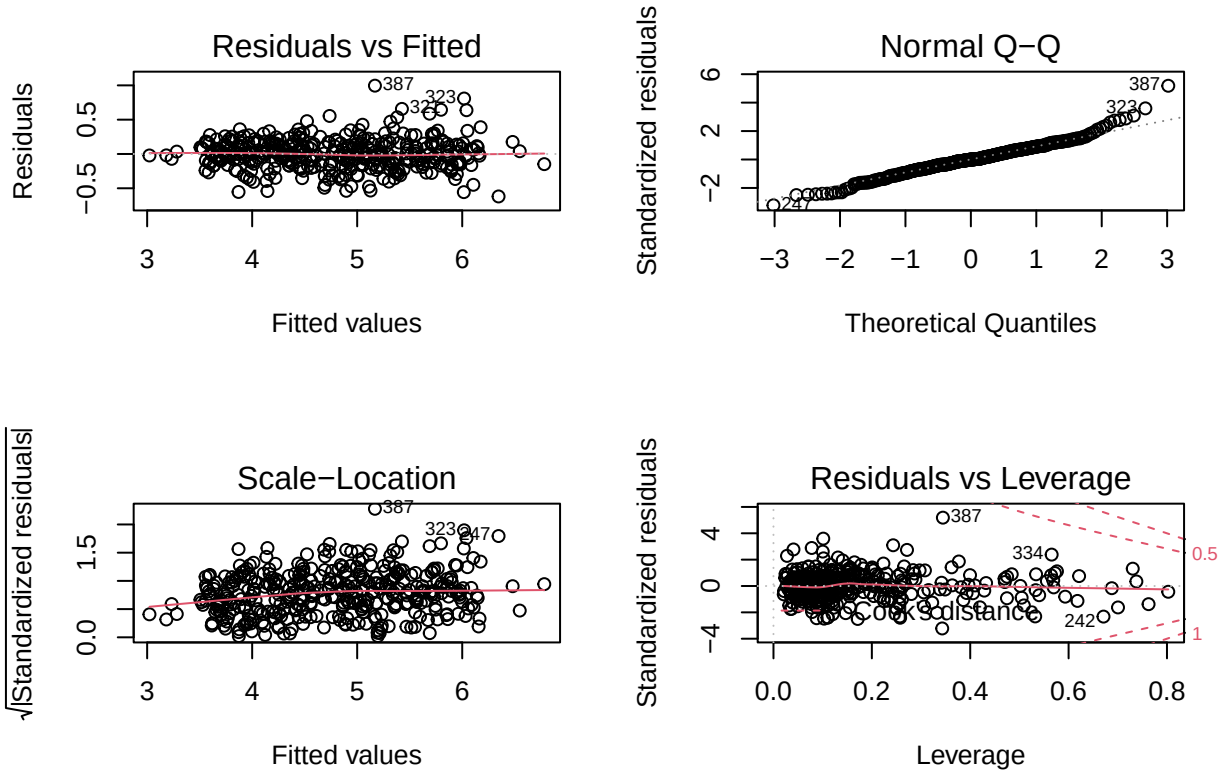

```
## it looks like sqrt() works well for mpg, and log() works reasonably well
## for displacement, horsepower and weight.

## it is possible to be fussier about the transformations, but then you have to
## figure out how to explain a very fussy transformation to your co-worker or
## client.

data_p1 <- Nameless_Auto

data_p1 <-data_p1[,-grep("mpg",names(data_p1))]
data_p1$sqrt_mpg <- sqrt(Nameless_Auto$mpg)

data_p1 <-data_p1[,-grep("displacement",names(data_p1))]
data_p1$log_displacement <- log(Nameless_Auto$displacement)

data_p1 <-data_p1[,-grep("horsepower",names(data_p1))]
data_p1$log_horsepower <- log(Nameless_Auto$horsepower)

data_p1 <-data_p1[,-grep("weight",names(data_p1))]
data_p1$log_weight <- log(Nameless_Auto$weight)

lm.p1.1 <- lm(sqrt_mpg ~ ., data=data_p1)
par(mfrow=c(2,2))
plot(lm.p1.1)
```




```
lm.p1.2 <- lm(sqrt_mpg ~ .^2, data=data_p1)
par(mfrow=c(2,2))
plot(lm.p1.2)
```




```
lm.p1.3 <- lm(sqrt_mpg ~ .^3, data=data_p1)
par(mfrow=c(2,2))
plot(lm.p1.3)
```



There's not a lot to distinguish these residual plots, and so I will invoke the "principle of parsimony" and just choose to examine the first model more closely (although there are indeed slight improvments in the residuals for the other models, perhaps especially the third one).

From the summary

```
summary(lm.p1.1)
```

```
##
## Call:
## lm(formula = sqrt_mpg ~ ., data = data_p1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98033 -0.16427 -0.01022  0.15266  1.03773
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      13.826816   0.946114  14.614  < 2e-16 ***
## cylinders         0.005869   0.026620   0.220  0.82563
## acceleration     -0.019796   0.009561  -2.070  0.03908 *
## year              0.072729   0.004384  16.588  < 2e-16 ***
## origin            0.069286   0.026064   2.658  0.00818 **
## log_displacement -0.071752   0.132478  -0.542  0.58840
## log_horsepower   -0.587390   0.145739  -4.030 6.71e-05 ***
## log_weight       -1.425817   0.204366  -6.977 1.33e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.2865 on 384 degrees of freedom
## Multiple R-squared:  0.875,  Adjusted R-squared:  0.8728
## F-statistic: 384.1 on 7 and 384 DF,  p-value: < 2.2e-16
```

we see that all of the predictors except for `cylinder` *and* `year` *and* `log_displacement` *are significant. The model we would end up with, then, is*

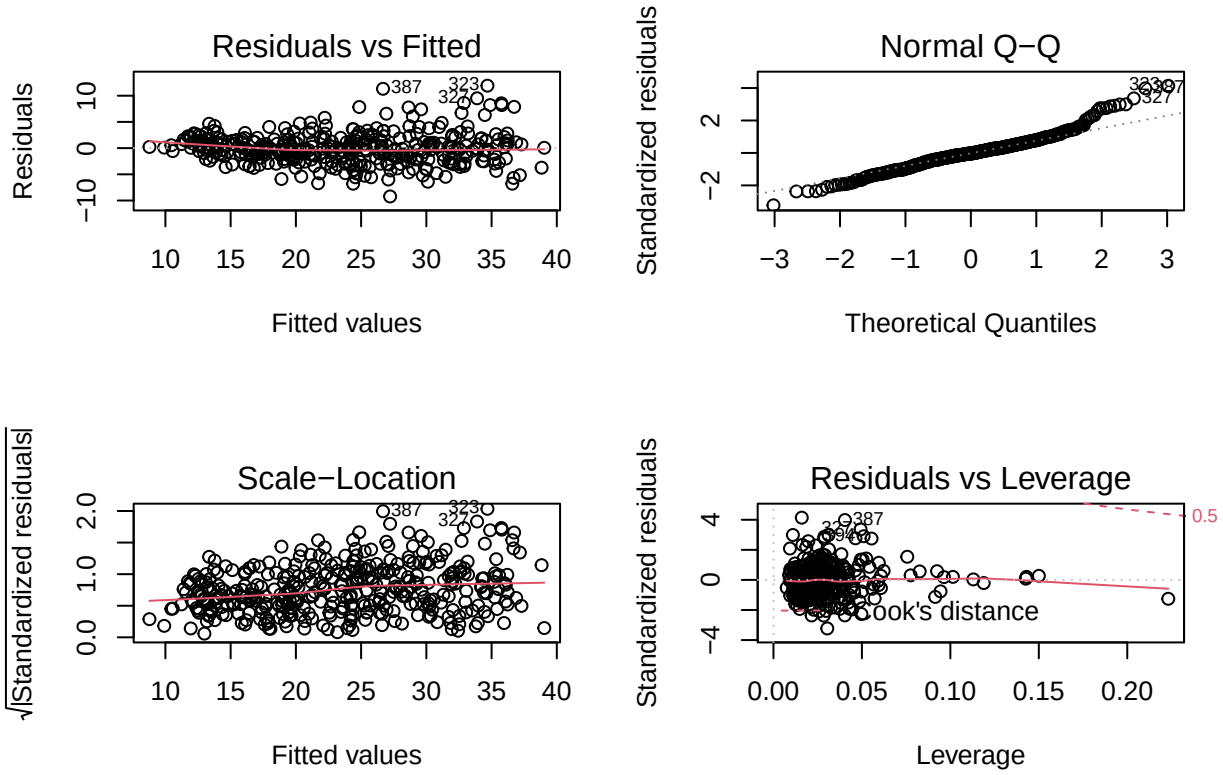
```
sqrt_mpg ~ accelleration + year + origin + log_horsepower + log_weight          (3)
```

Here's the second idea in action:

- *Looking back at the scatter plots in problem 2(a), it looks like* `mpg` *has the most nonlinear relationships with* `displacement`*,* `horsepower`*, and* `weight`*.*
- *Since* `sqrt(mpg)` *worked well before, I'm guessing that when we leave* `mpg` *alone, we should square the predictors.*
- *A variation of the "hierarchy priniple" says that when you include a power of a predictor in a model you should include all lower-order powers as well.*

Here's an initial model that implements these ideas:

```
data_p2 <- Nameless_Auto
data_p2$displacement2 <- (data_p2$displacement)^2
data_p2$horsepower2 <- (data_p2$horsepower)^2
data_p2$weight2 <- (data_p2$weight)^2

lm.p2.1 <- lm(mpg ~ ., data=data_p2)
par(mfrow=c(2,2))
plot(lm.p2.1)
```



These residual plots look about the same to me as the plots for `lm.p1.1`*, so let's see what predictors look significant here:*

```
summary(lm.p2.1)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = data_p2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2232 -1.5534 -0.0931  1.4304 11.9162
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.509e+00  4.789e+00   0.733 0.464247
## cylinders      4.113e-01  3.275e-01   1.256 0.209886
## displacement  -3.513e-02  2.005e-02  -1.752 0.080556 .
## horsepower    -1.915e-01  4.096e-02  -4.675 4.09e-06 ***
## weight        -1.067e-02  2.590e-03  -4.122 4.62e-05 ***
## acceleration  -1.735e-01  1.004e-01  -1.728 0.084870 .
## year           7.692e-01  4.512e-02  17.048  < 2e-16 ***
## origin         5.788e-01  2.668e-01   2.170 0.030643 *
## displacement2  6.324e-05  3.463e-05   1.826 0.068661 .
## horsepower2    5.268e-04  1.384e-04   3.807 0.000164 ***
## weight2        1.047e-06  3.488e-07   3.002 0.002862 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.902 on 381 degrees of freedom
## Multiple R-squared:  0.8653, Adjusted R-squared:  0.8618
## F-statistic: 244.8 on 10 and 381 DF,  p-value: < 2.2e-16
```

The significant predictors here are `horsepower`, `weight`, `year`, `origin`, `horsepower2` *and* `weight2`, *so the model we would get is*

```
mpg ~ horsepower + weight + year + origin + horsepower2 + weight2                (4)
```

Aside:

The models (1), (2), (3) and (4) seem really different. How could we choose between them? There are basically three perspectives on model choice:

Fit/Validity. Sheather's textbook is dominated by this perspective. The two main questions are: Does the model fit the data, and can we verify that the assumption of the model actually hold?

Predictive Accuracy. The ISLR book is dominated by this perspective. The main question is, regardless of how well the model fits or how valid the assumptions are, does the model provide a good "machine" for predicting future observations? This is where issues like the bias-variance tradeoff for mean-squared prediction error come into play.

Scientific/Policy Usefulness. The Gelman-Hill book is dominated by this perspective. Does the model agree with or improve on existing theory for how the data is generated? Is it useful for making new scientific inferences and/or new social policy decisions, etc.?

Notice that I say each book is "dominated" by a particular perspective. No good statistician only approaches modeling and inference from just one perspective, and neither do these books. You can find some of each perspective in all three books, and you will find it useful to approach your work from combinations of these perspectives as well. But each book tends to emphasize one perspective over the others, as I have indicated above.

Problem 3.

Problem 3(a)

The `kidiq.csv` file is in the same directory as this assigment. Read it into a data frame in R with a command like `kidiq <- read.csv("kidiq.csv",header=TRUE)`. Use the `cbind()` command to create:

- `y` = a column vector (matrix with one column), from the column `kid.score` in `kidiq`.
- `X` = a matrix with (a) the first column containing all 1's; (b) the second column containing the column `mom.hs` from `kidiq`; and (c) the third columnn containing the column `mom.iq` from `kidiq`.

Use `dim()` to verify that `y` is 434×1, and `X` is 434×3. Use the `head()` command to print out the first few rows of `y` and `X`, and turn the results of the `dim()` and `head()` commands in.

```
kidiq <- read.csv("kidiq.csv",header=TRUE)
str(kidiq)
```

```
## 'data.frame':    434 obs. of  6 variables:
##  $ X        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ kid.score: int  65 98 85 83 115 98 69 106 102 95 ...
##  $ mom.hs   : int  1 1 1 1 1 0 1 1 1 1 ...
##  $ mom.iq   : num  121.1 89.4 115.4 99.4 92.7 ...
##  $ mom.work : int  4 4 4 3 4 1 4 3 1 1 ...
##  $ mom.age  : int  27 25 27 25 27 18 20 23 24 19 ...
```

```
y <- cbind(kidiq$kid.score)
X <- cbind(1, with(kidiq, cbind(mom.hs,mom.iq)))

dim(y)
```

```
## [1] 434   1
```

```
dim(X)
```

```
## [1] 434   3
```

```
head(y)
```

```
##      [,1]
## [1,]   65
## [2,]   98
## [3,]   85
## [4,]   83
## [5,]  115
## [6,]   98
```

```
head(X)
```

```
##        mom.hs    mom.iq
## [1,] 1      1 121.11753
## [2,] 1      1  89.36188
## [3,] 1      1 115.44316
## [4,] 1      1  99.44964
## [5,] 1      1  92.74571
## [6,] 1      0 107.90184
```

Problem 3(b)

Compute $V = (X^T X)^{-1}$ in R, and show the result (V should be a 3×3 matrix; why?).

```
V <- solve(t(X) %*% X)
print(V)
```

```
##                              mom.hs        mom.iq
##         0.1049491626 -0.0001705848 -1.025110e-03
## mom.hs -0.0001705848  0.0148740410 -1.151616e-04
## mom.iq -0.0010251098 -0.0001151616  1.115594e-05
```

`X` *is* 434×3*, so* `t(X)` *is* 3×434*. Therefore* `t(X)%*%X` *multiplies a* 3×434 *matrix with a* 434×3 *matrix; the inner dimensions cancel and the result has the outer dimensions,* 3×3*. Finally the inverse of a square matrix has the same dimensions as the original matrix,* 3×3 *in this case.*

Problem 3(c)

Compute $\hat{\beta} = (X^T X)^{-1} X^T y$ in R, and show the result.

```
beta.hat <- V %*% t(X) %*% y
print(beta.hat)
```

```
##             [,1]
##        25.731538
## mom.hs  5.950117
## mom.iq  0.563906
```

Problem 3(d)

Calculate the residual vector $y - X\hat{\beta}$ in R, and use the result to compute the residual variance s^2 (don't forget to divide by $n - k$; what are n and k here?). Show the resulting s^2.

```
n <- dim(X)[1]
k <- dim(X)[2]
res.var <- t(y - X%*%beta.hat) %*% (y - X%*%beta.hat) / (n-k)
print(res.var)
```

```
##          [,1]
## [1,] 328.9028
```

```
## this is s^2
```

Problem 3(e)

Calculate the matrix $\text{Var}(\hat{\beta}) = (X^T X)^{-1} s^2$ in R, and extract the square roots of the diagonal elements of this matrix. These are the standard errors $SE(\hat{\beta}_0)$, $SE(\hat{\beta}_1)$, and $SE(\hat{\beta}_2)$.

```
var.beta <- V * c(res.var)
beta0.sd <- sqrt(var.beta[1,1])
beta1.sd <- sqrt(var.beta[2,2])
beta2.sd <- sqrt(var.beta[3,3])

cbind(beta.hat, c(beta0.sd, beta1.sd, beta2.sd))
```

```
##             [,1]       [,2]
##        25.731538 5.87520802
## mom.hs  5.950117 2.21181218
## mom.iq  0.563906 0.06057408
```

```
res.sd <- sqrt(res.var)
print(res.sd)
```

```
##          [,1]
## [1,] 18.13568
```

```
## This is s, the residual SE.
```

Problem 3(f)

Compare your values for $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$ (from part), and their standard errors (from part) with the result of running

```
summary(lm(kid.score ~ mom.hs + mom.iq, data=kidiq))
```

Comment on any similarities or differences.

```
summary(lm(kid.score ~ mom.hs + mom.iq, data=kidiq))
```

```
##
## Call:
## lm(formula = kid.score ~ mom.hs + mom.iq, data = kidiq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.873 -12.663   2.404  11.356  49.545
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.73154    5.87521   4.380 1.49e-05 ***
## mom.hs       5.95012    2.21181   2.690  0.00742 **
## mom.iq       0.56391    0.06057   9.309  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.14 on 431 degrees of freedom
## Multiple R-squared:  0.2141, Adjusted R-squared:  0.2105
## F-statistic: 58.72 on 2 and 431 DF,  p-value: < 2.2e-16
```

The estimated $\hat{\beta}$'s, $SE(\hat{\beta})$'s, and s, are identical with the results of the raw matrix calculations, to as many displayed decimal places as they have in common.

Problem 4.

In the folder for this hw assignment you will find a pdf called "An IMRAD paper on wine", based on Example 1.2.4 in Sheather. This paper is based only on EDA, not on any more sophisticated methods.

Problem 4(a)

Read the slides "IMRAD: What goes into each section", a pdf of which is in the same folder as this assignment. (There is nothing to turn in for this part).

Problem 4(b)

Does the paper appropriately address each of the parts of an IMRAD paper as described the "IMRAD: What goes into each section" pdf?

For each section below, either say "yes this section has the right content", or say "no" and describe what is missing and/or what needs to be moved to another section of the paper or deleted.

- Abstract
- Introduction
- Methods
- Results
- Discussion

Later we will see (and write) more complex versions of IMRAD and IDMRAD papers; this is just a first taste!

Here are some possible "good" answers. Your answers may be a bit different, but as long as they are thoughtful and take into account some of the important features of each section of an IMRAD paper (as described in the slides I asked you to read), you should be fine.

- *Abstract*
  Yes, this section has the right content.

  The slides on IMRAD don't discuss Abstracts, but a good rule of thumb is that the abstract should have about as many sentences as there are main sections in the paper; each sentence gives the "highlight" for the corresponding section of the paper. In this case, the first sentence summarizes the intrduction; the second and third sentences summarize the methods section; Sentences 4-7 summarize the results; and sentence 8 summarizes the discussion. (It is a little awkward because the answer to the main question of the paper, that Parker's ratings have a bigger impact than Coates', comes in the middle of the abstract instead of toward the end where the summary of the discussion is, but otherwise it seems fine.)

- *Introduction*
  Yes, it mostly does a good job.
  - + *Provides a rationale and aim for the study ("determine whether Parker or Coates has a grweater effect on wine prices")*
  - + *Supplies sufficient background (pretty much the whole first paragraph)*
  - − *Doesn't do much lit review (What would previous literature be like, for this article? Would it be about wine? Would it be about the reliability and accuracy of wine critics? Would it be on what makes a critic or "influencer" believable?)*
  - + *Brief, clear to the point, and in present tense*
  - *Doesn't address some of the "optional" items, like study design, main results, etc.*
- *Methods*
  Yes, this is not too bad, and is fairly typical of the methods section of a "data analysis" paper.
  - + It adequately addresses "Who? What? When? Where? How? Why?".
  - + It says where the data came from.
  - − It does not describe study design (how each wine was selected for the study, how it was extracted from Parker's and Coates' publications, etc.).
  - + It does describe all the variables in the study, and therefore also describes how the effect (Parker vs Coates) was measured.
  - + It describes the data analysis method (pure EDA in this case)!
  - − It would probably be better if the graphs themselves appeared in the Results section, since the Methods section should not contain any Results or Discussion material.
  - For this paper there do not appear to be any ethical considerations to discuss, but for other papers there would be, such as ethical aspects of data collection & study design, confidentiality of individuals' data, etc.
- *Results*
  Yes this section has the right content, for such a short simple paper.

  Because the methods are so simple and the EDA was already shown in the Methods section (it really didn't belong there, it should have been here in the Methods section), so what we have here is a description of how the EDA was used to get results for each research question, and the reasoning behind each of the results.

  There is one paragraph for each research question announced in the Introduction (in a paper with more involved analyses, there might be one subsection for each research question, rather than just one paragraph, with corresponding subsections back in the Methods section). The paragraphs summarize the analyses and give appropriate conclusions (in a paper with more involved analyses, e.g. regression analysis etc., there might also be tables listing coefficient estimates & standard errors, graphs showing predicted values, etc.).
- *Discussion*
  Not quite the right content.

  The first paragraph should summarize the main results of the paper. The first paragraph here is a mixture of new results (should be in the results section!) and study limitations (should come later in the discussion!). The first sentence of the last paragraph would be better as the first sentence of the Discussion. The remaining text describes further limitations and other considerations, which is fine.