# 36-617: Applied Linear Models
# Fall 2022
# HW01 – Due Weds Sept 07, 11:59pm

- Please turn the homework in online in our course webspace at canvas.cmu.edu.
  - There is a link to Gradescope in the description of this assignment on Canvas.
  - You should submit a single pdf to Gradescope. If you need help with this, please see https://www.cmu.edu/teaching/gradescope/index.html. Also, allow yourself some extra time to create the pdf & upload it in Gradescope.
  - Gradescope allows the TA to grade all the problem 1's together, then all the problem 2's, and so forth. This leads to more consistent grading and better comments for you.
- Reading:
  - For this week: Sheather Ch 5 (supplemental: ISLR 3.1, 3.2; G&H Ch 3)
  - For next week: ISLR 3.1, 3.2, 3.3.1, 3.3.2 (supplemental: G&H Ch 3)
- There are four exercises below.

## Exercises

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

   - Please be sure you are reading and using the *second edition* of the ISLR text. I mistakenly put the first edition on canvas earlier, but now the second edition is there.
   - The `Auto` data set is one of the data sets for the ISLR text. The data sets for the ISLR book are listed briefly on pp. 13–14. To get them, you must install the ISLR2 package from `https://cran.r-project.org/`. In Rstudio this is easiest to do by going to the **Tools** menu, selecting the first option (**Install packages...**) and following your nose. You can also do this at the R command prompt, with a command like `install.packages("ISLR2")`. You only have to install a package once, but to use the package (i.e. to have access to the data sets, in this case) you must execute the command `library(ISLR2)` at the beginning of your R or Rstudio session.

2. ISLR second Ed. p. 123–124, #9.

   - Notes for part (e): The syntax and semantics for interactions in lm() and other modeling functions take a little getting used to. To include the product of two variables, say $weight \times horsepower$, you would include the term `weight:horsepower` on the right hand side of the regression model, so

     ```
     lm(mpg ~ weight:horsepower,data=Auto)
     ```

     would fit the model

     $$(mpg) = \beta_0 + \beta_1 \cdot (weight) \cdot (horsepower) + \epsilon$$

     On the other hand, the R notation `weight*horsepower` includes the product *and all lower-order terms* (in this case, the main effects). So for example either of the following two commands

     ```
     lm(mpg ~ weight*horsepower,data=Auto)
     lm(mpg ~ weight + horsepower + weight:horsepower,data=Auto)
     ```

fit the same model

$$(mpg) = \beta_0 + \beta_1 \cdot (weight) + \beta_2 \cdot (horsepower) + \beta_3 \cdot (weight) \cdot (horsepower) + \epsilon$$

You should almost always prefer * rather than `:`. Whenever you include an interaction in the model you should also include all lower-order terms from that interaction.

(Note also that the intercept is included by default in all models. To get rid of the intercept, you have to add -1 to your model formula.)

- Notes for part (f): You can apply a function like `log(mpg)` or `sqrt(horsepower)` directly to any variable in an R model statement. Powers are a little trickier; because of the semantics of the modeling language in R (see notes for part (e) above), the command `lm(y ~ x + x^2)` produces the same model as `lm(y ~ x)` (try it!). To get powers, it is safest to construct new variables, e.g.

```
x2 <- x^2
lm(y ~ x + x2)
```

You can accomplish the same thing with the `I()` function: `lm(y ~ x + I(x^2))`, though constructing a separate variable and putting it in the data frame with the other variable(s) is usually better for using the `predict()` function, etc.

3. This exercise introduces some base-R notation for working with matrices, and also illustrates that the matrix formulation that we talked about in class really does produce the usual least-squares estimates for regression coefficients, standard errors, and so forth. Here are some of the R commands and notation that you will use:

- `cbind(x,y,z, ...)` creates a matrix with columns x,y,z, etc. If one of the varaibles is shorter than the others, its elements are repeated in order until it is the right length. (cbind can also do other things, like join two matrices, two data frames, etc.)
- `dim(X)` returns a vector with the dimensions of the matrix (or data frame) X: `dim(X)[1]` is the number of rows, `dim(X)[2]` is the number of columns.
- `%*%` does matrix multiplication. (* does scalar and elementwise multiplication)
- For any matrix X, `t(X)` is the transpose of X.
- For any square matrix X, `diag(X)` produces the diagonal elements of X.
- If X is an invertible matrix, `solve(X)` produces $X^{-1}$, the inverse matrix.
- `round(X,d)` rounds all the numbers in X to d decimal places.
- If X is a data frame with named columns, the `$` operator selects columns by name, for example:

```
> X <- as.data.frame(matrix(rnorm(9),ncol=3))
> names(X) <- c("fred","bobby","sue")
> X
       fred      bobby        sue
1  0.977772 -0.5336657 -0.2353455
2  0.522910  0.4758267  1.5450506
3 -2.988834  0.7029876  0.4422550
> X$fred
[1]  0.977772  0.522910 -2.988834
> X$bobby
[1] -0.5336657  0.4758267  0.7029876
> X$sue
[1] -0.2353455  1.5450506  0.4422550
```

Now, on to the exercise!

(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()` commansd to print out the first few rows of `y` and `X`, and turn the results of the `dim()` and `head()` commands in.

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

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

(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 $s^2$.

(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 $SE(\hat{\beta}_2)$.

(f) Compare your values for $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$ (from part 3c), and their standard errors (from part 3e) with the result of running

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

Comment on any similarities or differences.

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.

(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).

(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!