# CFerrari_Assignment5

*Charley Ferrari*

*Friday, February 27, 2015*

## Problem Set 1

In this problem set we'll work out some properties of the least squares solution that we reviewed in the weekly readings. Consider the unsolvable system $Ax = b$ as given below:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 8 \\ 19 \end{bmatrix}$$

–Write R markdown script to compute $A^T A$ and $A^T b$

```
A <- matrix(c(1,1,1,1,0,1,3,4),nrow=4)
b <- matrix(c(0,8,8,20), nrow=4)

AtA <- t(A) %*% A
Atb <- t(A) %*% b

AtA
```

```
##      [,1] [,2]
## [1,]    4    8
## [2,]    8   26
```

```
Atb
```

```
##      [,1]
## [1,]   36
## [2,]  112
```

–Solve for $x$ in $R$ using the above two computed matrices.

```
xhat <- solve(AtA) %*% Atb

xhat
```

```
##      [,1]
## [1,]    1
## [2,]    4
```

–What is the squared error of this solution?

$A\hat{x} - b = e$

$||e||^2 = ||A\hat{x} - b||^2$

```
esquared <- t((A %*% xhat) - b) %*% ((A %*% xhat) - b)

esquared
```

```
##      [,1]
## [1,]   44
```

–Instead of b = [0 8 8 20], start with p = [1 5 13 17] and find the exact solution (i.e. show that this system is solvable as all equations are consistent with each other. This should result in an error vector e = 0).

$Ax = p$

```
p <- matrix(c(1,5,13,17),nrow=4)

library(pracma)

rref(cbind(A,p))
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    1
## [2,]    0    1    4
## [3,]    0    0    0
## [4,]    0    0    0
```

As you can see in the RREF form of the matrix, we get $x_1 = 1$ and $x_2 = 4$ looking at the original problem as a system of equations:

$x_1 = 1$

$x_1 + x_2 = 5$ This already gives the solution set $x_1 = 1$ and $x_2 = 4$

$x_1 + 3x_2 = 13$ This is correct

$x_1 + 4x_2 = 17$ This is also correct

– Show that the error $e = b - p = $ [-1 3 -5 2].

```
e <- b - p

e
```

```
##      [,1]
## [1,]   -1
## [2,]    3
## [3,]   -5
## [4,]    3
```

– Show that the error $e$ is orthogonal to $p$ and to each of the columns of $A$.

Lets compute the dot products

```
t(e) %*% p
```

```
##      [,1]
## [1,]    0
```

```r
t(e) %*% A[,1]
```

```
##      [,1]
## [1,]    0
```

```r
t(e) %*% A[,2]
```

```
##      [,1]
## [1,]    0
```

## Problem Set 2

Consider the modified auto-mpg data (obtained from the UC Irvine Machine Learning dataset). This dataset contains 5 columns: displacement, horsepower, weight, acceleration, mpg. We are going to model mpg as a function of the other four variables. Write an R markdown script that takes in the auto-mpg data, extracts an A matrix from the first 4 columns and b vector from the fifth (mpg) column. Using the least squares approach, your code should compute the best fitting solution. That is, find the best fitting equation that expresses mpg in terms of the other 4 variables. Finally, calculate the fitting error between the predicted error of your model and the actual mpg. Your script should be able to load in the 5 column data set, extract A and b, and perform the rest of the calculations. Please have adequate comments in your code to make it easy to follow your work

First, I'll read in the data. It seems to be in a weird format, so I used scan, and coerced it into a matrix:

```r
setwd("E:/Downloads/Courses/CUNY/SPS/Git/IS 605 Fundamentals of Computational Mathematics/Assignment 5")

autodata <- scan("auto-mpg.data")
autodata <- t(matrix(autodata, nrow = 5))
```

Next I'll create my A matrix and b vector:

```r
Adata <- autodata[,1:4]
bdata <- autodata[,5]
```

Now to find my $A^t A$ and $A^t b$:

```r
AtAdata <- t(Adata) %*% Adata
Atbdata <- t(Adata) %*% bdata

AtAdata
```

```
##              [,1]         [,2]        [,3]        [,4]
## [1,]   19097634    9374647.0   259345480   1123011.9
## [2,]    9374647    4857524.0   132989885    607832.3
## [3,]  259345480  132989885.0  3757575489  17758103.6
## [4,]    1123012     607832.3    17758104     97656.9
```

```r
Atbdata
```

```
##             [,1]
## [1,]  1529685.9
## [2,]   868718.8
## [3,] 25209061.4
## [4,]   146401.4
```

And to solve for $\hat{x}$:

```
xhatdata <- solve(AtAdata) %*% Atbdata

xhatdata
```

```
##              [,1]
## [1,] -0.030037938
## [2,]  0.157115685
## [3,] -0.006217883
## [4,]  1.997320955
```

This $\hat{x}$ is a vector of the coefficients to my equation $Ax = b$ that minimizes the square errors.

This equation can be written as:

$$-0.030037938 \times displacement + 0.157115685 \times horsepower-$$
$$0.006217883 \times weight + 1.997320955 \times acceleration = mpg$$

Interestingly, this didn't match what the check I ran using the lm function, and I realized the intercept was missing from this model. Once I coerced a 0 intercept, I was able to get everything to match:

```
lm(autodata[,5] ~ 0 + autodata[,1:4])
```

```
##
## Call:
## lm(formula = autodata[, 5] ~ 0 + autodata[, 1:4])
##
## Coefficients:
## autodata[, 1:4]1  autodata[, 1:4]2  autodata[, 1:4]3  autodata[, 1:4]4
##       -0.030038          0.157116         -0.006218          1.997321
```

Looking into an EViews econometric textbook (having never really understood their linear algebra interpretation of regression), I realized all I needed to do was add a "1" column to my A matrix:

```
AdataInt <- cbind(1,autodata[,1:4])
bdataInt <- autodata[,5]

AtAdataInt <- t(AdataInt) %*% AdataInt
AtbdataInt <- t(AdataInt) %*% bdataInt

xhatdataInt <- solve(AtAdataInt) %*% AtbdataInt

xhatdataInt
```

```
##             [,1]
## [1,] 45.251139699
## [2,] -0.006000871
## [3,] -0.043607731
## [4,] -0.005280508
## [5,] -0.023147999
```

```r
lm(autodata[,5] ~ autodata[,1:4])
```

```
##
## Call:
## lm(formula = autodata[, 5] ~ autodata[, 1:4])
##
## Coefficients:
##      (Intercept)  autodata[, 1:4]1  autodata[, 1:4]2  autodata[, 1:4]3
##        45.251140         -0.006001         -0.043608         -0.005281
## autodata[, 1:4]4
##        -0.023148
```