# Chapter 10

# The Least Squares Problem

## 10.1 Introduction to Least Squares

A very popular problem in the sciences and engineering is to translate a collection of data points into a corresponding mathematical model that accurately describes the underlying phenomenon being measured. Suppose we are running an experiment in which we measure certain dependent physical quantitates as a function of some other independent physical quantity. For different values of our independent variable $x_i$ we make the measurement $y_i$ of the corresponding measurement for our dependent variable. In this way, we obtain a set of $m$ data points given by

$$(x_1, y_1), \qquad (x_2, y_2), \qquad \ldots \qquad (x_m, y_m).$$

We can then use Microsoft Excel, MATLAB, Google Docs, or even our TI-84 calculators to graph our given data. Given the shape of the graph, we can guess what type of function might model our data. We might use the following function types to model our data:

General Models Used for Data Fitting

| Model Type | General Equation |
|---|---|
| Linear polynomials: | $y(x) = a_0 + a_1 x$ |
| Quadratic polynomials: | $y(x) = a_0 + a_1 x + a_2 x^2$ |
| General $n$th degree polynomial: | $y(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ |
| Exponential Function: | $y(x) = a_0 e^{\,a_1 \, x}$ |
| Periodic (Sinusoidal) Function: | $y(x) = a_0 + a_1 \cos(x) + a_2 \sin(x)$ |
| Any linear combinations of Functions: | $y(x) = a_1 f_1(x) + a_2 f_2(x) + \cdots + a_n f_n(x)$ |

We can then generate a possible model as an $m \times n$ linear system.

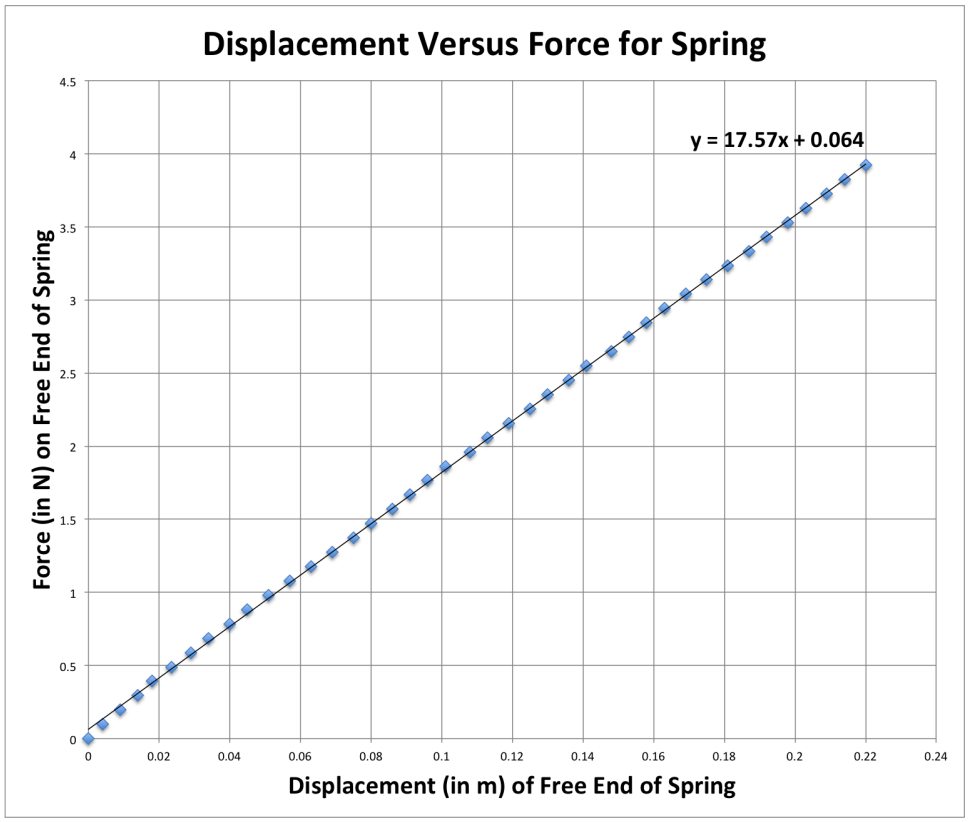
**EXAMPLE 10.1.1**

Let's look back at Example 2.2.6 from our early discussion of vectors. Recall, in this example, we studied the physical properties of mass-spring systems to verify **Hooke's law** via experiment. We saw that the internal force in a spring was directly proportional to the elongation of the spring. We can state this in vector form as follows:

$$\mathbf{f} = k\mathbf{u}$$

where $k$ is the specific spring constant for the spring we use in our experiment, $\mathbf{u}$ is the calculated displacement vector from Example 2.2.5 and $\mathbf{f}$ is the calculated force vector from Example 2.2.1.

Note that the spring constant of a spring is a measurement of stiffness. The higher the spring constant, the harder it is to pull the spring apart. We can use excel's trend line chart option to get a formula for the value of $k$ in this experiment (see the figure below):




Consider our data points $\{(x_i, y_i)\}_{i=1}^{m}$ above (in the Hooke's law example, we see $m = 41$ since we collected 41 data points). In theory, we should be able to find $a_0, a_1 \in \mathbb{R}$ such that all of these data points lie on line

$$y(x) = a_0 + a_1 x.$$

As we notice in our graph above however, are data points cannot be fit to a line exactly. This is because experimental error has been introduced when the measurements were made do to inaccuracy of measuring instruments, human error and other

related inaccuracies. Thus, it is impossible to find $a_0, a_1 \in \mathbb{R}$ that models all our data exactly (in other words, are resulting linear system is going to be inconsistent).

Assuming we choose two values $a_0, a_1 \in \mathbb{R}$ to model our data, then we can measure the *error* between our chosen model $y = a_0 + a_1 x$ and each data point $(x_i, y_i)$ from our experiment. The individual errors for each of our $m$ data points can be quantified:

$$e_i = y_i - (a_0 + a_1 x_i)$$

where $i = 1, 2, ..., m$. We can write the resulting system of $m$ equations in matrix form:

$$\mathbf{e} = \mathbf{y} - \mathbf{A}\mathbf{x}$$

where

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}.$$

We call the vector $\mathbf{e}$ the *error vector* and the vector $\mathbf{y}$ the *data vector*. We know, from our study of linear systems, that if we can fit our data exactly, so that $y_i = a_0 + a_1 x_i$ for each $i$, we have $e_i = 0$. In the language of linear algebra, all of our data points lie on a straight line if and only if $\mathbf{y} \in \text{Col}(A)$.

In the Hooke's law experiment presented above, our collected data is not collinear (cannot be measured exactly by one line). One way we can redefine our search for a line is to try to find a line that minimizes the euclidean norm of our error vector:

$$\text{Error} = \|\mathbf{e}\|_2 = \sqrt{e_1^2 + e_2^2 + \cdots + e_m^2}.$$

In this paradigm, we attempt to minimize the square roots of the sum of the squares of the individual errors, hence the term least squares. In short, when trying to find $a_0, a_1 \in \mathbb{R}$ to model our data, we want to find vector $\mathbf{x}$ that minimizes the euclidean norm of the error vector:

$$\|\mathbf{e}\|_2 = \|A\mathbf{x} - \mathbf{y}\|_2.$$

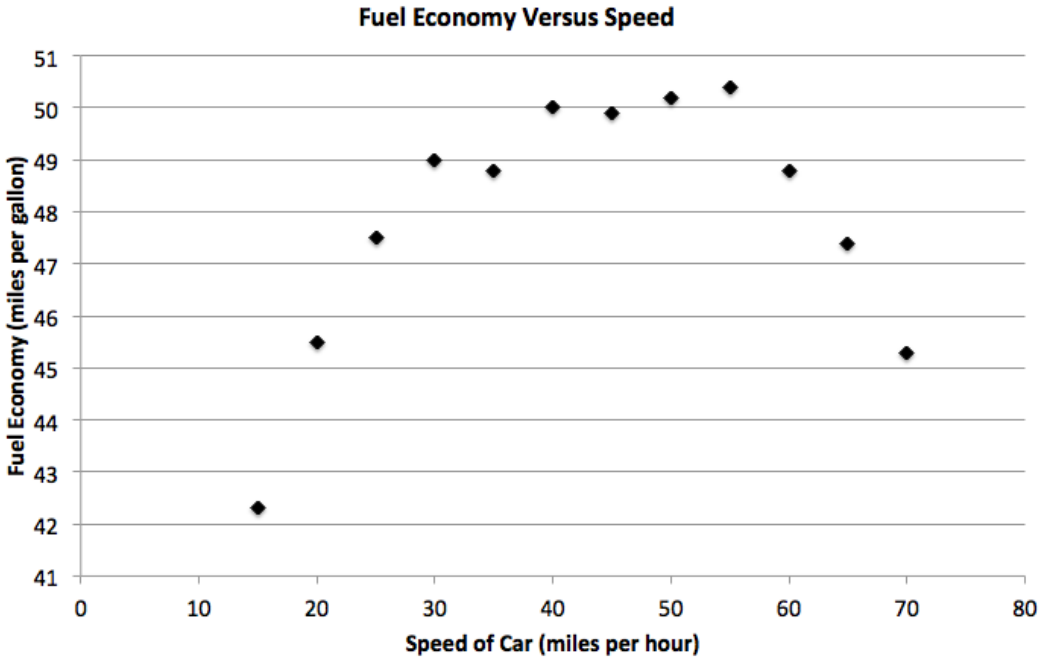
This problem is now right in the center of study of linear algebra and has an elegant solution.

**EXAMPLE 10.1.2**

Suppose you are an automotive engineer. You are building a new hybrid car that runs on both electricity and gasoline. As part of your design process, you want to study the relationship of the speed of your car to the fuel economy of the car. You design an experiment to discover this relationship and track the following data:

| Speed (mph) | Fuel Economy |
|---|---|
| 15 | 42.3 |
| 20 | 45.5 |
| 25 | 47.5 |
| 30 | 49.0 |
| 35 | 48.8 |
| 40 | 50.00 |
| 45 | 49.9 |
| 50 | 50.2 |
| 55 | 50.4 |
| 60 | 48.8 |
| 65 | 47.4 |
| 70 | 45.3 |

In order to set up the least-squares problem related to this data that will allow us to model our vehicle's fuel economy, lets first plot our data:




We notice that this data seems to be best described by a quadratic polynomial:

$$y(x) = a_0 + a_1 x + a_2 x^2.$$

Let us now generate the corresponding linear system using the Vandermonde matrix for our data:

$$\begin{bmatrix} 1 & 15 & 225 \\ 1 & 20 & 400 \\ 1 & 25 & 625 \\ 1 & 30 & 900 \\ 1 & 35 & 1225 \\ 1 & 40 & 1600 \\ 1 & 45 & 2025 \\ 1 & 50 & 2500 \\ 1 & 55 & 3025 \\ 1 & 60 & 3600 \\ 1 & 65 & 4225 \\ 1 & 70 & 4900 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 42.3 \\ 45.5 \\ 47.5 \\ 49.0 \\ 48.8 \\ 50.0 \\ 49.9 \\ 50.2 \\ 50.4 \\ 48.8 \\ 47.4 \\ 45.3 \end{bmatrix}$$

This can be stated as the matrix equation

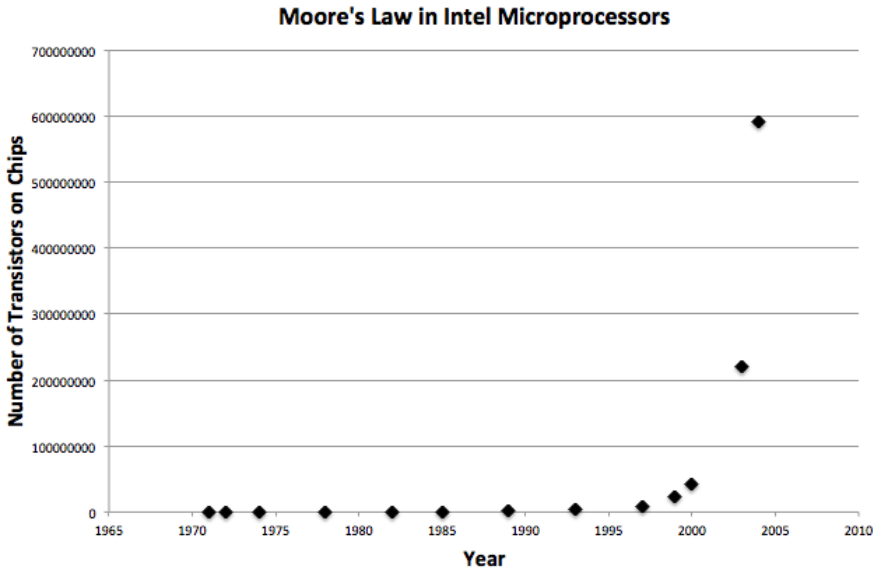
$$A\mathbf{x} = \mathbf{b}$$

**EXAMPLE 10.1.3**

"Moore's law is the observation that, over the history of computing hardware, the number of transistors in a dense integrated circuit doubles approximately every two years. The law is named after Gordon E. Moore, co-founder of Intel Corporation, who described the trend in his 1965 paper." (See Wikipedia Article on Moore's Law). We can track the size of integrated circuits over a number of years. Below you'll find a data table that does exactly this task.

| Processor | Year of release | Number of transistors (1000's) |
|---|---|---|
| *4004* | 1971 | 2.3 |
| *8008* | 1972 | 2.5 |
| *8080* | 1974 | 5 |
| *8086* | 1978 | 29 |
| *80286* | 1982 | 120 |
| *80386* | 1985 | 275 |
| *80486* | 1989 | 1,180 |
| *Pentium* | 1993 | 3,100 |
| *Pentium II* | 1997 | 7,500 |
| *Pentium III* | 1999 | 24,000 |
| *Pentium 4* | 2000 | 42,000 |
| *Itanium 2* | 2003 | 220,000 |
| *Itanium 2 (9MB cache)* | 2004 | 592,000 |

We can graph this data and we notice that the growth in numbers of transistors on Intel chips seems to fit an exponential curve:




In this case, we want to model our data using an **exponential model**

$$y(x) = a_0 e^{a_1 x}.$$

However, we cannot achieve such a model directly through least squares because the two unknowns $a_0$ and $a_1$ are not linearly related in our equation. One way to deal with this problem is to use the rules of logarithms that we know and love to

linearize our problem:

$$\begin{aligned}\ln(y) &= \ln\left(a_0 e^{\,a_1\,x}\right), \\ &= \ln(a_0) + a_1 x, \\ &= k + a_1 x,\end{aligned}$$

where we've introduced a new constant $k = \ln(a_0)$. Now, both unknown coefficients $k, a_1$ are in a linear model and were back to solving problems using the linear regression. We can solve the related matrix equation

$$A\mathbf{x} = \mathbf{b}$$

to find paramaters $k$ and $a_1$. To translate our solution into the corresponding exponential model, we will simply use the translation $a_0 = e^k$.

**Definition 10.1: Vandermonde Matrix for Polynomial Interpolation**

Given the $m$ data points collected above $\{x_i, y_i\}_{i=1}^m$, we can approximate this data using an $n$th degree polynomial in the form

$$y(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

by solving a least squares problem. The total least squares error between the data and the sample values of the function is equal to

$$\|\mathbf{e}\|_2^2 = \sum_{i=1}^{m} [y_i - y(x_i)]^2 = \|\mathbf{y} - A\mathbf{x}\|_2^2$$

where

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Here, the matrix $A$ is known as the $m \times n+1$ **Vandermonde matrix**. If $m = n+1$, then $A$ is square. Assuming $A$ is invertible in this case, we can solve $A\mathbf{x} = \mathbf{y}$ exactly. In the case where $m > n+1$, we will not get an exact interpolating polynomial. Instead, we will get the best fit polynomial.

**Definition 10.2: Generalized Vandermonde Matrix**

Given the $m$ data points collected above $\{x_i, y_i\}_{i=1}^m$, we do not have to try to fit our data to a polynomial. Instead, suppose we want to fit our data using a linear combinations of functions of our choosing. In other words, suppose we choose functions $h_1(x), h_2(x), ..., h_n(x)$ and we want to model our data in the form

$$y(x) = a_1 f_1 x + a_2 f_2(x) + \cdots + a_n f_n(x).$$

Again, we can do this by solving a least squares problem. The total least squares error between the data and the sample values of the function is equal to

$$\|\mathbf{e}\|_2^2 = \sum_{i=1}^{m} [y_i - y(x_i)]^2 = \|\mathbf{y} - A\mathbf{x}\|_2^2$$

where

$$A = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_m) & f_2(x_m) & \cdots & f_n(x_m) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

A particularly important case is provided by the $2n+1$ trigonometric functions

$$f_1(x) = 1,$$

$$f_2(x) = \cos(x), \qquad f_3(x) = \sin(x),$$

$$f_4(x) = \cos(2x), \qquad f_5(x) = \sin(2x),$$

$$\vdots \qquad \vdots$$

$$f_{2n}(x) = \cos(nx), \qquad f_{2n+1}(x) = \sin(nx).$$

Interpolation on $2n+1$ equally spaced data points on the interval $[0, 2\pi]$ leads to the famous Discrete Fourier Transform, used in signal processing, data transmission and compression and many other application areas.

**Definition 10.3: The Least Squares Problem**

Suppose $A \in \mathbb{R}^{m \times n}$ with $m > n$ and $\mathbf{b} \in \mathbb{R}^m$. The **least squares problem** is to find the vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ such that $\|A\hat{\mathbf{x}} - \mathbf{b}\|_2$ achieves a minimum value. In other words, we want to find $\hat{\mathbf{x}} \in \mathbb{R}^n$ such that

$$\|A\hat{\mathbf{x}} - \mathbf{b}\|_2 \leq \|A\mathbf{x} - \mathbf{b}\|_2$$

for all $\mathbf{x} \in \mathbb{R}^n$. Assuming we can find such a vector $\hat{\mathbf{x}}$, we call this vector the **least squares solution** of $A\mathbf{x} = \mathbf{b}$.

**Theorem 41: The Normal Equation to Solve the Least Squares Problem**

The set of least-squares solutions of $A\mathbf{x} = \mathbf{b}$ coincides with the nonempty set of solutions of the normal equations:

$$A^T A\mathbf{x} = A^T \mathbf{b}.$$

**Theorem 42: The Solution to the Normal Equation**

Let $A \in \mathbb{R}^{m \times n}$. The following statements are equivalent:

a. Equation $A\mathbf{x} = \mathbf{b}$ has a unique least-squares solution for all $\mathbf{b} \in \mathbb{R}^m$.

b. The columns of $A$ are linearly independent.

c. The matrix $A^T A$ is invertible.

When these statements are true, the least-squares solution $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}.$$

# Normal Equations to Solve Least-Squares

Section 6.5 p. 360 - 368

Example 2 p. 354

Let $A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{4\times 3}$ and $\vec{b} = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 2 \end{bmatrix}_{4\times 1}$

Today → (i) Solve least-squares problem

$$\min_{x \in \mathbb{R}^3} \| \vec{b} - A\vec{x} \|_2$$

using Normal Equations [TI-84 Calc]

i. Solve least-squares problem $\min_{x \in \mathbb{R}^n} \| \vec{b} - A\vec{x} \|_2$ using normal equatio

Solution: For $A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 3}$ and $\vec{b} = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 2 \end{bmatrix} \in \mathbb{R}^{4\times 1}$

we see $A \in \mathbb{R}^{m\times n}$ with $m=4$ and $n=3$.

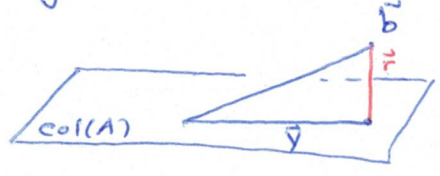
We know the theoretic solution to the problem

$$\min_{\vec{x} \in \mathbb{R}^n} \| \vec{b} - A\vec{x} \|_2$$

is based on projecting $\vec{b}$ orthogonally onto Col(A)

Side Note:

A great visual for this is below




Here, we want to choose $\vec{y} \in Col(A)$ such that $\vec{r} \perp Col(A)$.

To choose $\vec{y} \in Col(A)$ such that $\vec{r} = \vec{b} - \vec{y}$ is perpendicular to Col(A), we first realize

If $\vec{y} \in Col(A)$, then there is $x^* \in \mathbb{R}^n$ such that

$$A\vec{x}^* = \vec{y}$$

Then, we want to choose $\vec{x}^*$ such that

$$\vec{r} = \vec{b} - \vec{y} = \vec{b} - A\vec{x}^* \in [\text{Col}(A)]^\perp$$

We know by theorem 3 p. 335 $\text{Nul}(A^T) = [\text{Col}(A)]^\perp$

$\Rightarrow$ We want to choose $\vec{r} = b - A\vec{x}^* \in \text{Nul}(A^T)$

$\Rightarrow$ We want to find $\vec{x}^* \in \mathbb{R}^n$ s.t. $A^T\vec{r} = \vec{0}$

$\Rightarrow$ We want to find $\vec{x}^* \in \mathbb{R}^n$ s.t. $A^T(\vec{b} - A\vec{x}^*) = \vec{0}$

$\Rightarrow$ We want to find $\vec{x}^* \in \mathbb{R}^n$ s.t. $A^T\vec{b} - A^TA\vec{x}^* = \vec{0}$

$\Rightarrow$ We want $\vec{x}^* \in \mathbb{R}^n$ s.t. $A^TA\vec{x} = A^T\vec{b}$

By theorem 14 p. 363, we know

$$(A^TA)^{-1} \text{ exists} \Leftrightarrow \text{rank}(A) = 3 = n$$

Side note:

For $A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, we see

$$\text{RREF}(A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$\Rightarrow \text{rank}(A) = 3$ ✓

Because we know $\text{rank}(A) = 3 = n = \#$ columns of $A$, we can conclude $(A^TA)^{-1}$ exists and the solution to our least-squares problem is given by

$$\vec{x}^* = (A^TA)^{-1} A^T \vec{b}$$

Let's consider each component separately

$$A^TA = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{3\times 4} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{4\times 3}$$

inner dimensions agree :)

$$= \begin{bmatrix} 4 & 3 & 2 \\ 3 & 3 & 2 \\ 2 & 2 & 2 \end{bmatrix}_{3\times 3}$$

$$A^Tb = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}_{3\times 4} \begin{bmatrix} 1 \\ 3 \\ 1 \\ 2 \end{bmatrix}_{4\times 1}$$

inner dimensions agree :)

$$= \begin{bmatrix} 7 \\ 6 \\ 3 \end{bmatrix}$$

Now, to solve the original least-squares problem

$$\min_{\vec{x} \in \mathbb{R}^n} \| \vec{b} - A\vec{x} \|$$

we can instead solve the linear system problem

$$A^T A \vec{x} = A^T \vec{b}$$

$$\Leftrightarrow \begin{bmatrix} 4 & 3 & 2 \\ 3 & 3 & 2 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 6 \\ 3 \end{bmatrix}_{3\times 1}$$

$3\times 3$ and $3\times 1$: inner dimensions agree !!

To solve this linear system, we can

1. Perform Gaussian elimination by hand !!
2. Use a TI-84 Calculator
3. Use MATLAB

Let's use a TI-84 Calculator to ~~perform this~~ solve this linear system:

By using our TI-84 calculator, we see

$$\vec{x} = \left[A^T A\right]^{-1} A^T b = \begin{bmatrix} 1 \\ 2 \\ -1.5 \end{bmatrix}$$

is the solution to the least-squares problem

$$\min_{x \in \mathbb{R}^n} \| \vec{b} - A\vec{x} \|_2 .$$

Reflections: Using Normal Equations, we've found

- $\text{Proj}_{\text{Col}(A)}[\vec{b}] = A \cdot x^*$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -1.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 1.5 \\ 1.5 \end{bmatrix}$$

- $\text{Proj}_{[\text{Col}(A)]^\perp}(\vec{b}) = \vec{r} = \vec{b} - Ax^* = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 3 \\ 1.5 \\ 1.5 \end{bmatrix}$

$$= \begin{bmatrix} 0 \\ 0 \\ -0.5 \\ 0.5 \end{bmatrix}$$

~~Transform least squares~~

- Using normal equations, we've transformed the least-squares problem

$$\min_{x \in \mathbb{R}^n} \|\vec{b} - A\vec{x}\|_2$$

into equivalent linear system problem

$$A^T A \vec{x} = A^T \vec{b}$$

which can be solved using a matrix factorization iff $\text{rank}(A) = n$