CS 237: Probability in Computing

Wayne Snyder
Computer Science Department
Boston University

Lecture 25: Logistic Regression

- Motivation: Why Logistic Regression?
- Sigmoid functions and the logit transformation
- Cost functions
- Optimization by Gradient Descent

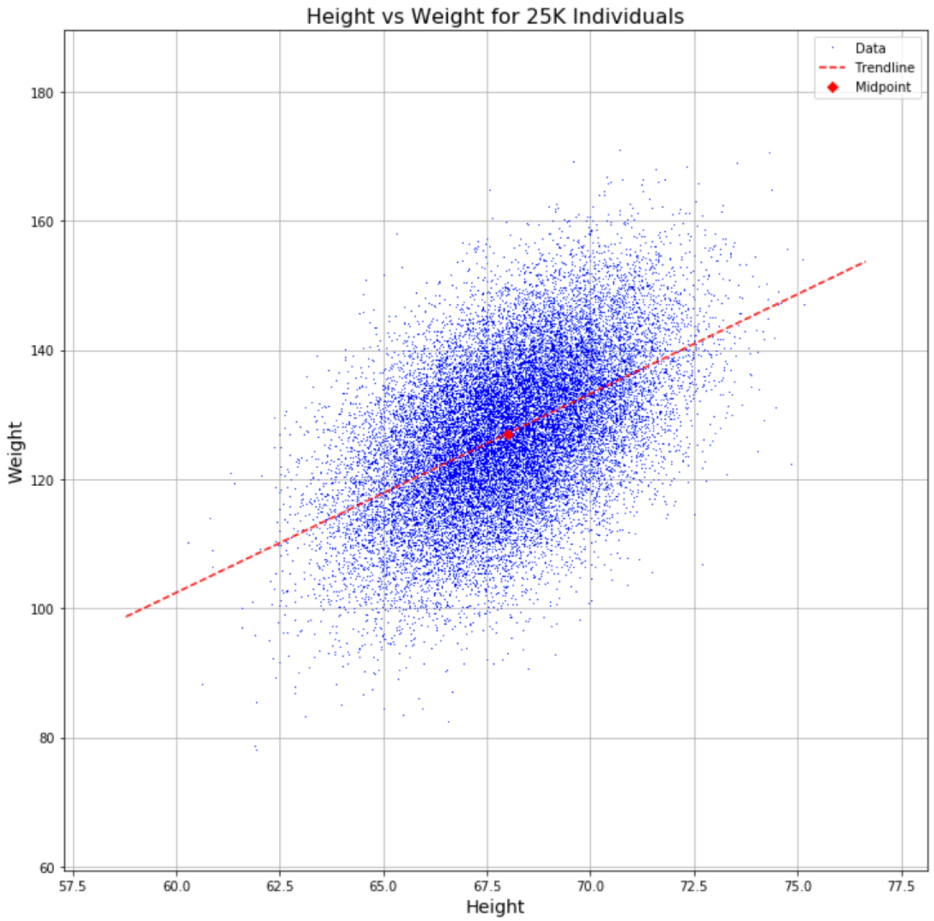
Next time: Putting it all together!

Linear Regression and Data Types

Linear Regression relates some number of independent variables

$$X_1,\ X_2,\ \ldots,\ X_n$$

with a dependent or response variable Y. All are assumed to be **real numbers:**



Linear Regression and Data Types

But probability and statistics deal with many different kinds of data, continuous and discrete, and we have dealt with at least the following during the semester:

- Real Numbers -- Height, weight, time, $X \sim N(\mu, \sigma^2)$, ... — Continuous
- Integers -- Number of emails, Cards, $X \sim B(N,p)$, ...
- Binary – Heads/Tails, Red/Black, $X \sim Bern(p)$, ...

But there are other kinds of (discrete) data which statisticians must consider:

Categorical -- A finite list of unordered categories or labels, e.g.,

- Political parties (Republican, Democrat, Independent, Green. ...)
- Blood type (A, B, AB, O, ...)
- State lived in (MA, VT, ...)

Ordinal – Like categorical, but with an explicit ordering, e.g.,

- Class year (freshman, sophomore, ...)
- Grades (A, A-, B+, ...)
- Likert Scale (disagree strongly, disagree, neutral, agree, agree strongly)

Discrete

Logistic Regression: The Basic Idea

The regression framework has been adapted to all these kinds of statistical information, but we will consider only the simplest: **binary or Bernoulli data**.

Logistic Regression is a modification of linear regression to deal with binary categories or binary outcomes. It relates some number of independent variables

$$X_1,\ X_2,\ \ldots,\ X_n$$

with a Bernoulli dependent or response variable Y , i.e., $R_Y = \{ 0, 1 \}$. It returns the probability p for Y ~ Bernoulli(p), i.e., the probability P(Y = 1).

Logistic regression can be used to provide the following kinds of binary outcomes:

- Given the results of medical tests A, B, C and D, does this patient have cancer?
- Given the credit score, annual salary, gender, and state of residency, will this customer default on the loan he/she is applying for?
- Is this email spam or not?
- Given a student's homework and midterm grades, will he/she get an A in CS 237?
- Given a person's height, what is their gender?

Logistic Regression: A Motivating Example

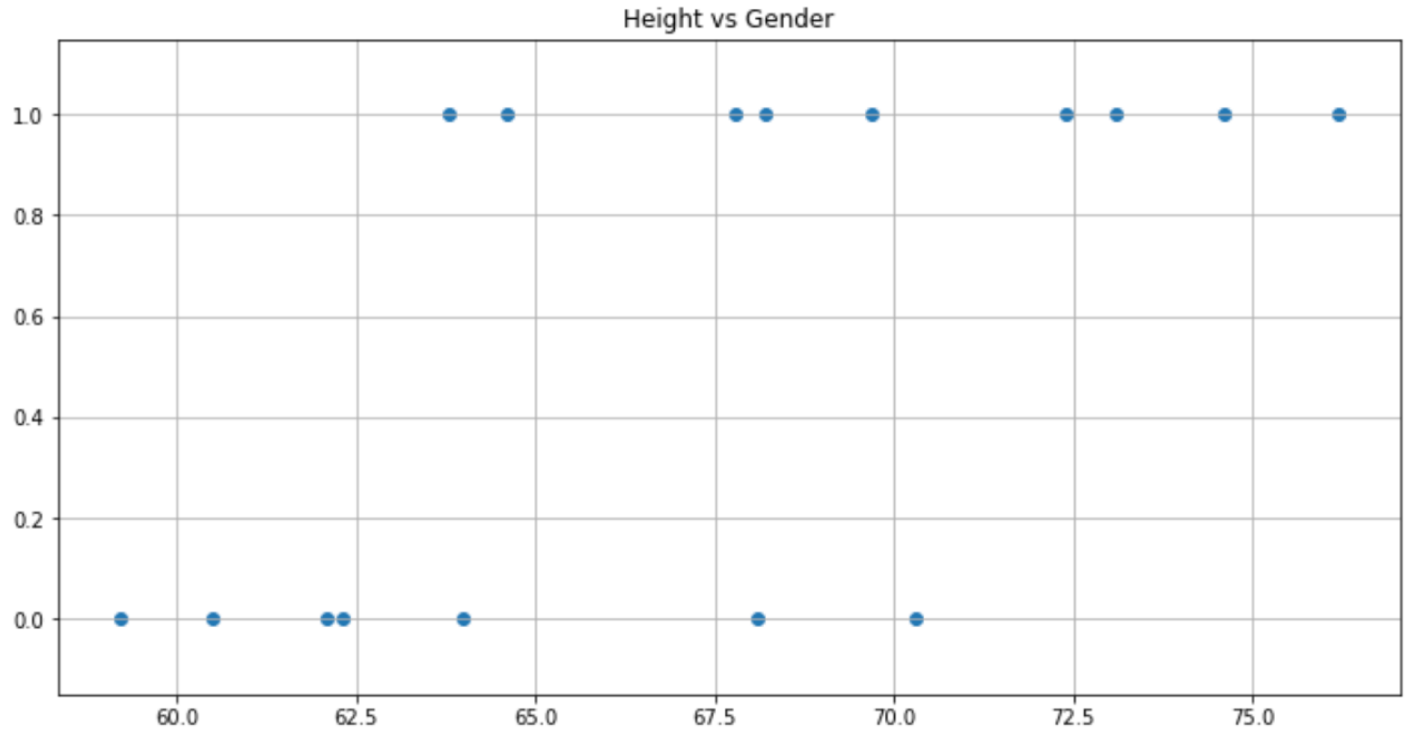
Suppose we consider the last example. Men in general are taller than women (the average height of an American man is 5' 9" and for women 5' 4"), but can we use this datum to predict a person's gender, or give the probability they are one or the other?

Let's try it: we proceed to collect data from a random sample of 16 people, and plot X = height against Y = gender (1 for male, 0 for female):

```
Heights: [59.2, 60.5, 62.1, 62.3, 63.8, 64.0, 64.6, 67.8, 68.1, 68.2, 69.7, 70.3, 72.4, 73.1, 74.6, 76.2]
Gender: [0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1]
```

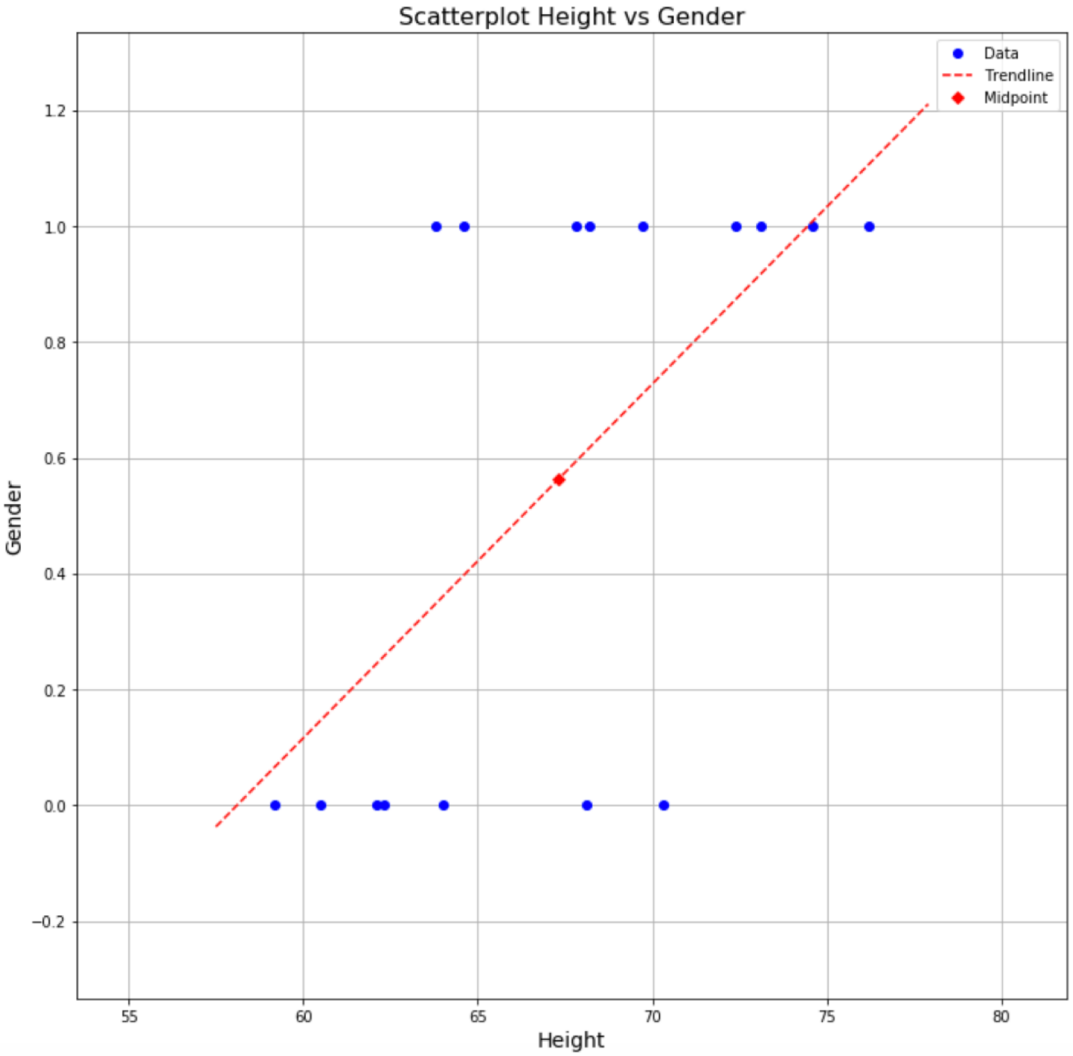
Male:

Female:



Logistic Regression: Motivating Example

If we plug this into the linear regression algorithm, we get the following:



rho: 0.6187 r^2: 0.3828

Residual SS: 2.4303 Explained SS: 1.5072 Total SS: 3.9375

Regression Line: y = 0.0612 * x - 3.554

There are many issues with this:

How can we use this to predict someone's gender from their height?

How to give the probability of their gender?

The R^2 value is bound to be very low (here, 0.3828), so how useful is this?

There is clearly no linear trend, so what does the line even mean?

In addition, there are technical and mathematical reasons why linear regression is not appropriate here.

Logistic Regression: The Logit Transformation

In order to solve this, we will use the same idea that we used for non-linear models: we will transform the scale of Y into a new domain, in this case into the real interval [0..1].

This is called the **Logit Transformation**, and is based on the notion of a **sigmoid function** $s : \mathcal{R} \rightarrow [0..1]$ of the form

$$s(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

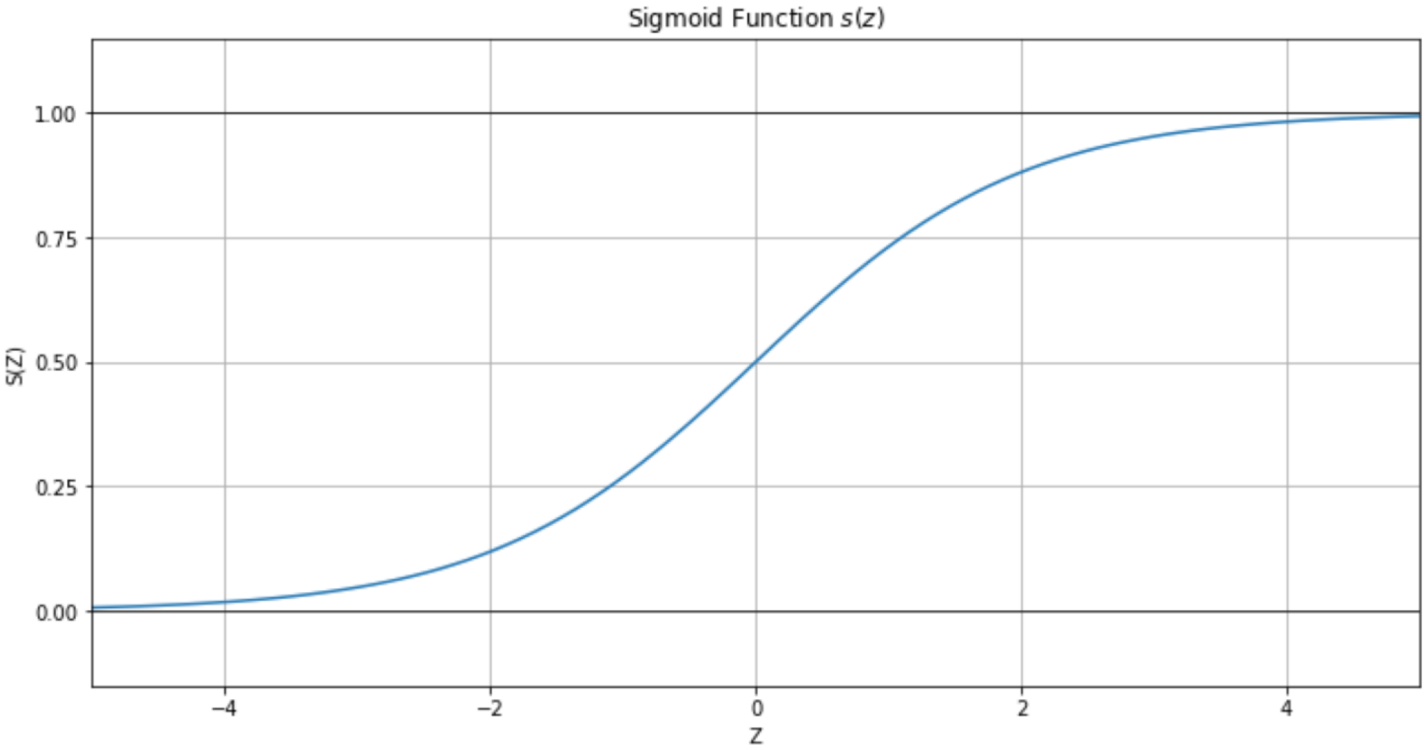
```
def s(z):
    return 1/(1+np.exp(-z))
```

Notice that:

$$\lim_{z \to \infty} \frac{1}{1 + e^{-z}} = 1$$

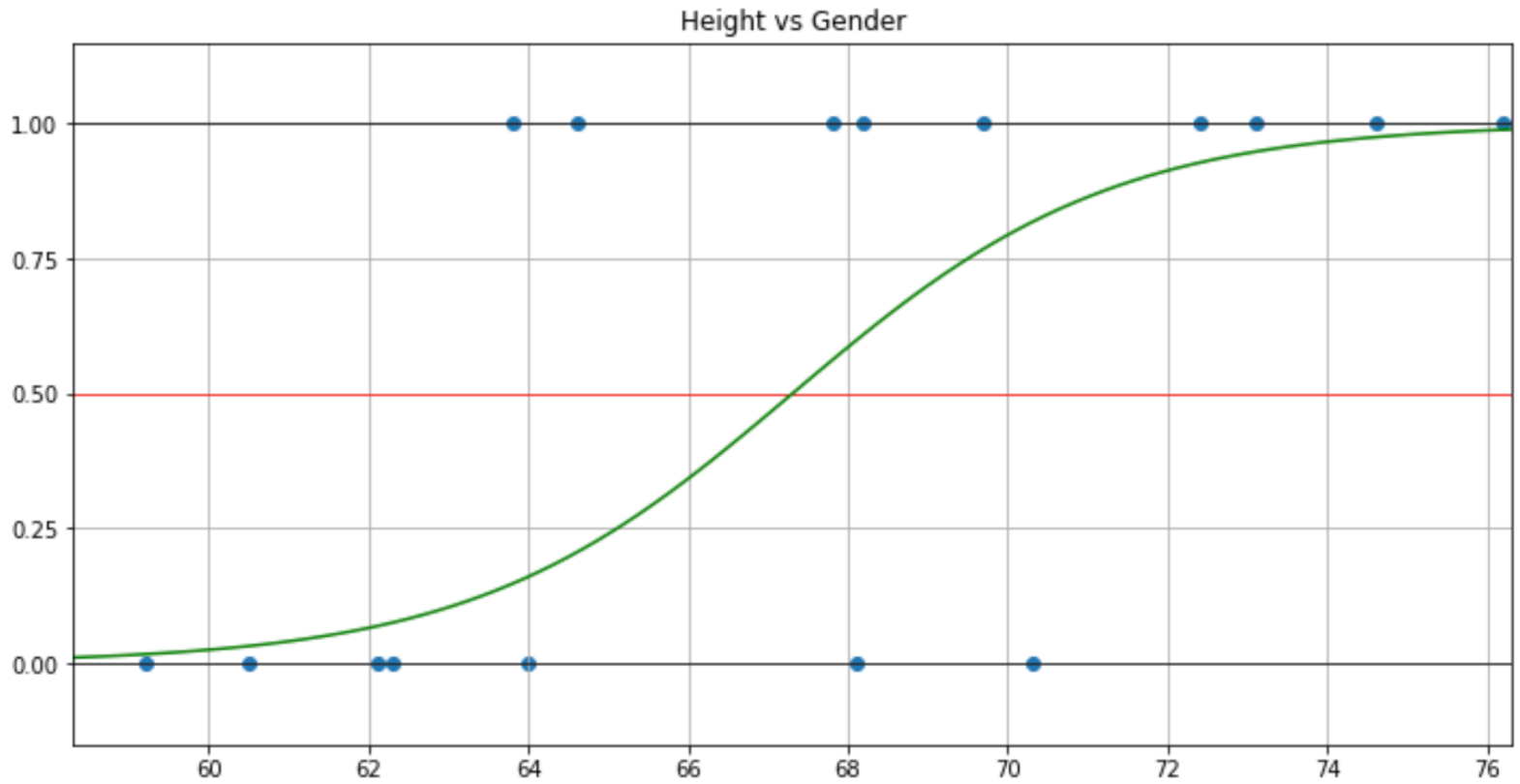
$$\lim_{z \to -\infty} \frac{1}{1 + e^{-z}} = 0$$

$$s(0) = \frac{1}{1 + e^0} = \frac{1}{1 + 1} = 0.5$$



Logistic Regression: The Logit Transformation

The punchline here is that we will transform the regression line into a sigmoid, and use it to give us the probability that a given individual is male, and then define as a **decision boundary** a threshold (typically 0.5) by which we will decide if the binary output is 1 or 0:

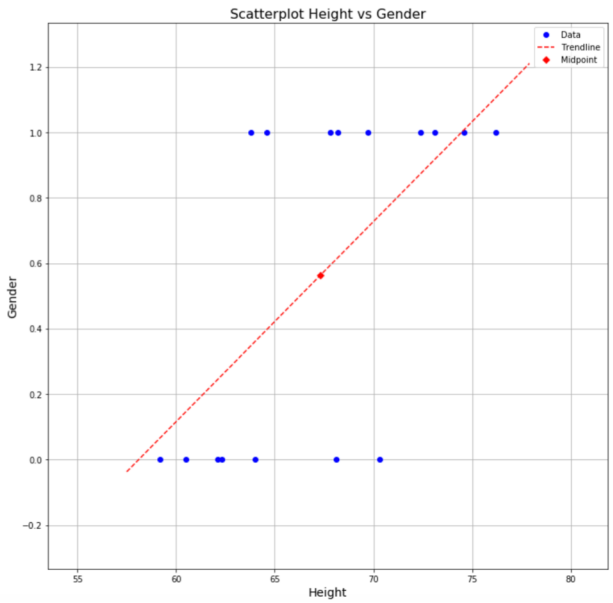


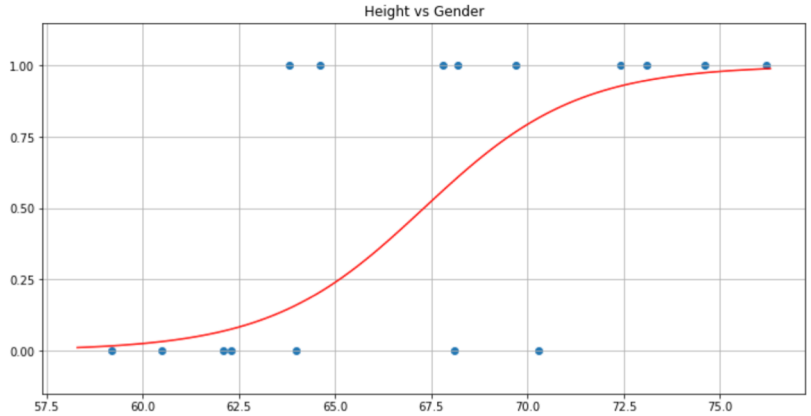
But in fact it is not that simple, because the **least squares technique does not work** any more, and we will have to recast the regression framework around the sigmoid function.....

Logistic Regression: The Logit Transformation

Linear Regression: $-\infty < Y < \infty$

Logistic Regression: $0 \leq Y \leq 1$





$$s(z) = \frac{1}{1+e^{-z}}$$

$$s : \mathcal{R} \rightarrow [0..1]$$

$$s^{-1} : [0..1] \rightarrow \mathcal{R}$$

$$s^{-1}(p) = ln\left(\frac{p}{1-p}\right)$$

This is called the "logit" or the "log odds ratio."

$$\begin{aligned}
y &= \frac{e^x}{1+e^x} \\
\Leftrightarrow \quad y + ye^x &= e^x \\
\Leftrightarrow \quad y &= e^x - ye^x \\
\Leftrightarrow \quad y &= (1-y)e^x \\
\Leftrightarrow \quad \frac{y}{(1-y)} &= e^x \\
\Leftrightarrow \quad ln\left(\frac{y}{1-y}\right) &= x \\
\Leftrightarrow \quad s^{-1}(p) &= ln\left(\frac{p}{1-p}\right)
\end{aligned}$$

Logistic Regression: The Case of One Independent Variable

We will first consider this in the case of one independent variable X (as in our motivating example), and then generalize to multiple X_1, ..., X_n, as we did for linear regression.

The basic equation gives us the probability for Y to be 1 given some value of the independent random variable X:

$$P(Y = 1) = s(\hat{\theta}_0 + \hat{\theta}_1 X) = \frac{1}{1 + e^{-(\hat{\theta}_0 + \hat{\theta}_1 X)}}$$

So far so good.... but we can't use the linear regression formulae to determine the parameters $\hat{\theta}_0$ and $\hat{\theta}_1$ and, even worse, there **exists no formula to calculate them.**

Question: So how do we find them?

Answer: We will use an **iterative approximation algorithm**, searching for the values which minimize the overall errors (though they are not defined the same way as in linear regression).

To see how to do this, let's go back and see how to apply this idea in linear regression....

Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

In linear regression, we have explicit formulae for finding the parameters for the slope and y-intercept of the regression line which minimizes the mean square error (MSE):

$$\hat{\theta}_1 = \rho(X, Y)\frac{\sigma^Y}{\sigma_X}$$

$$\hat{\theta}_0 = \mu_Y - \hat{\theta}_1\mu_X$$

But what if we didn't? We could then use an iterative approximation algorithm called **Gradient Descent** to find an approximation of the values which minimize the MSE.

Basic idea: Define a **cost or loss function** J(...) which gives the cost or penalty measuring how well the model parameters fit the actual data (high cost = bad fit), and then search for the parameters which minimize this cost.

So let's pretend we didn't have the formulae at the upper right, and suppose we needed to find them by gradient descent. In linear regression this would mean finding values for $\hat{\theta}_0$ and $\hat{\theta}_1$ which minimize the MSE, in other words minimize the cost function:

$$J(\hat{\theta}_0, \hat{\theta}_1) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (\hat{\theta}_0 + \hat{\theta}_1 x_i))^2$$

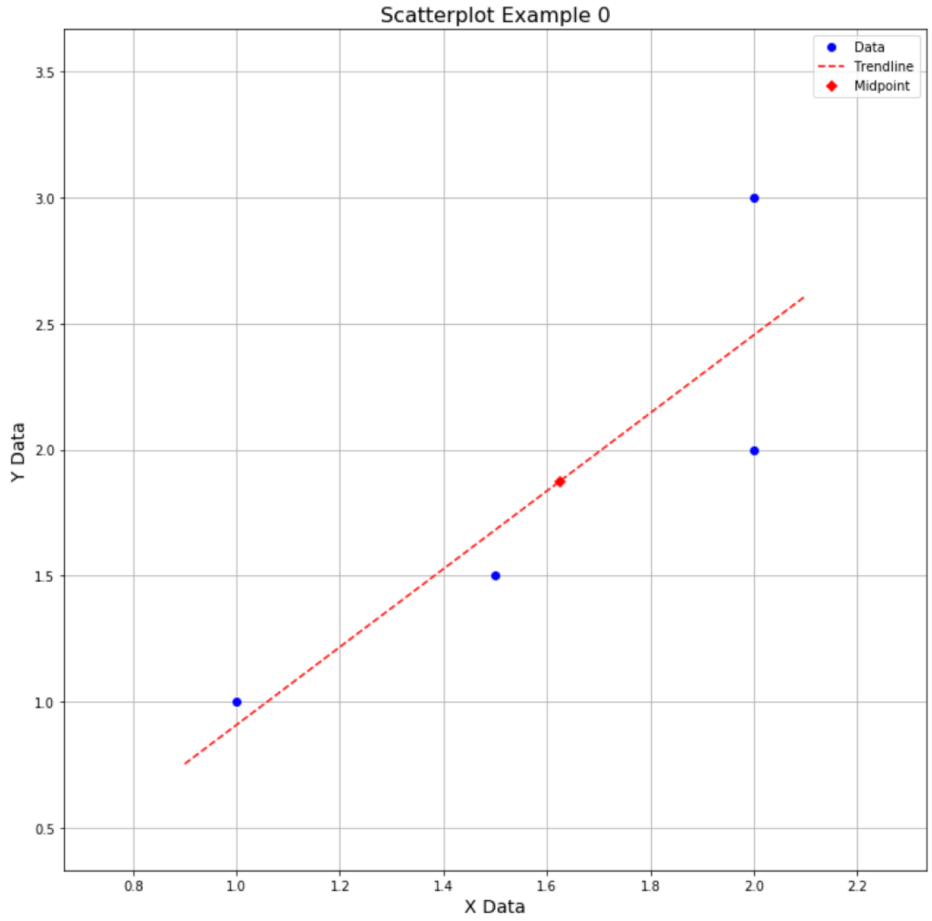
Cost Function
= MSE

The J in the cost function is used in machine learning and refers to the Jacobian Matrix.

Reference: https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html

Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Let's consider what the space of possible values for $\hat{\theta}_0$ and $\hat{\theta}_1$ looks like for this example of a linear regression line:



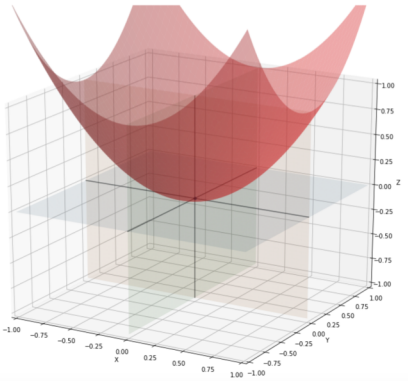

```
mean(x):        1.625   std(x): 0.4146
mean(y):        1.875   std(y): 0.7395

rho:   0.8664   r^2:   0.7506

Residual SS:   0.5455   Explained SS: 1.642     Total SS:   2.1875

Regression Line: y = 1.5455 * x - 0.6364
```

Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

3D Paraboloid:

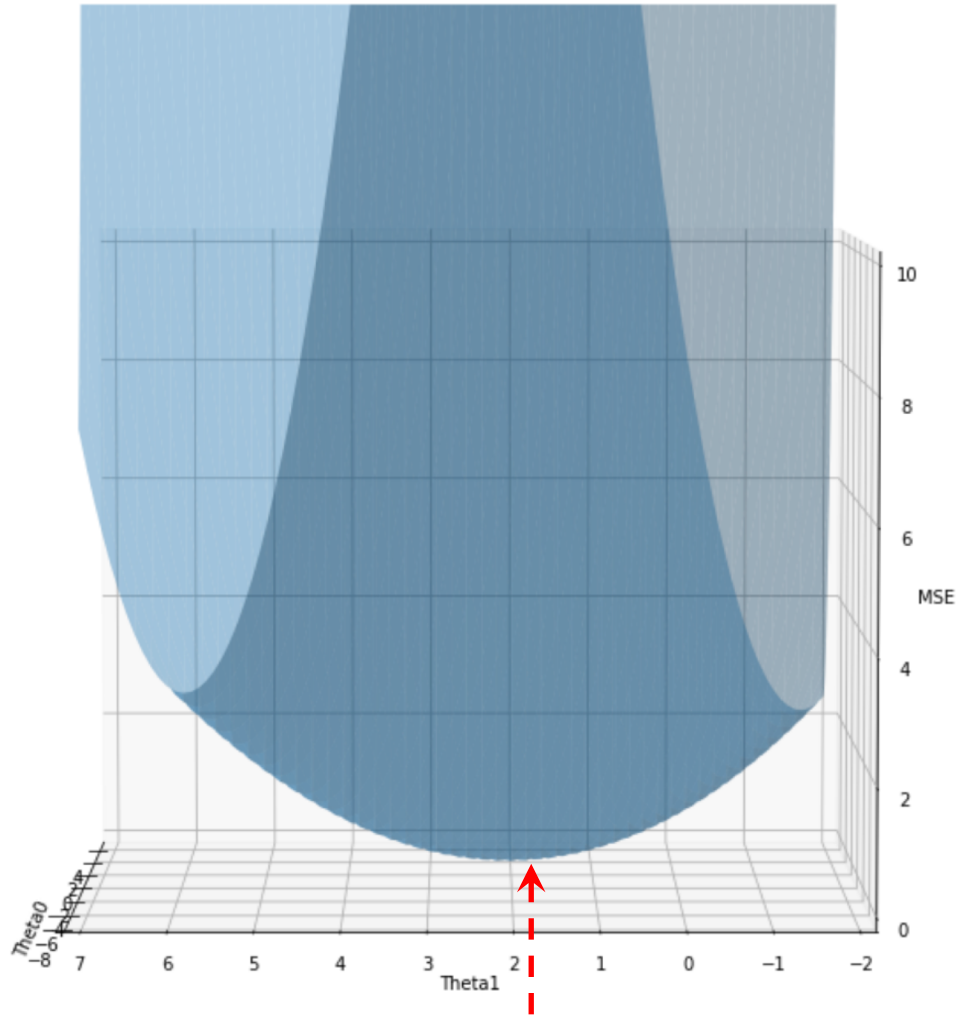
$z = x^2 + y^2$



Minimum values from the formulae are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$



Decreasing cost

$J(\hat{\theta}_0, \hat{\theta}_1)$

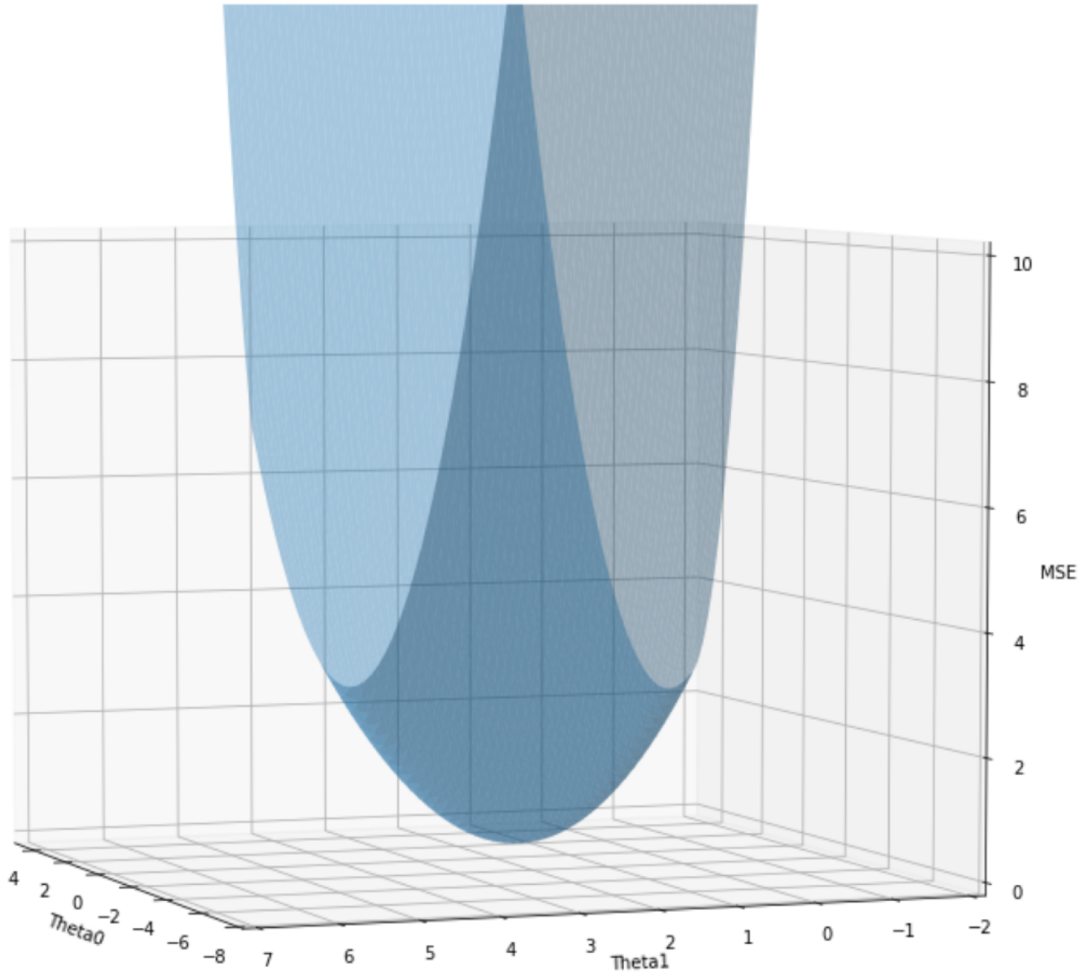
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

Minimum values are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$



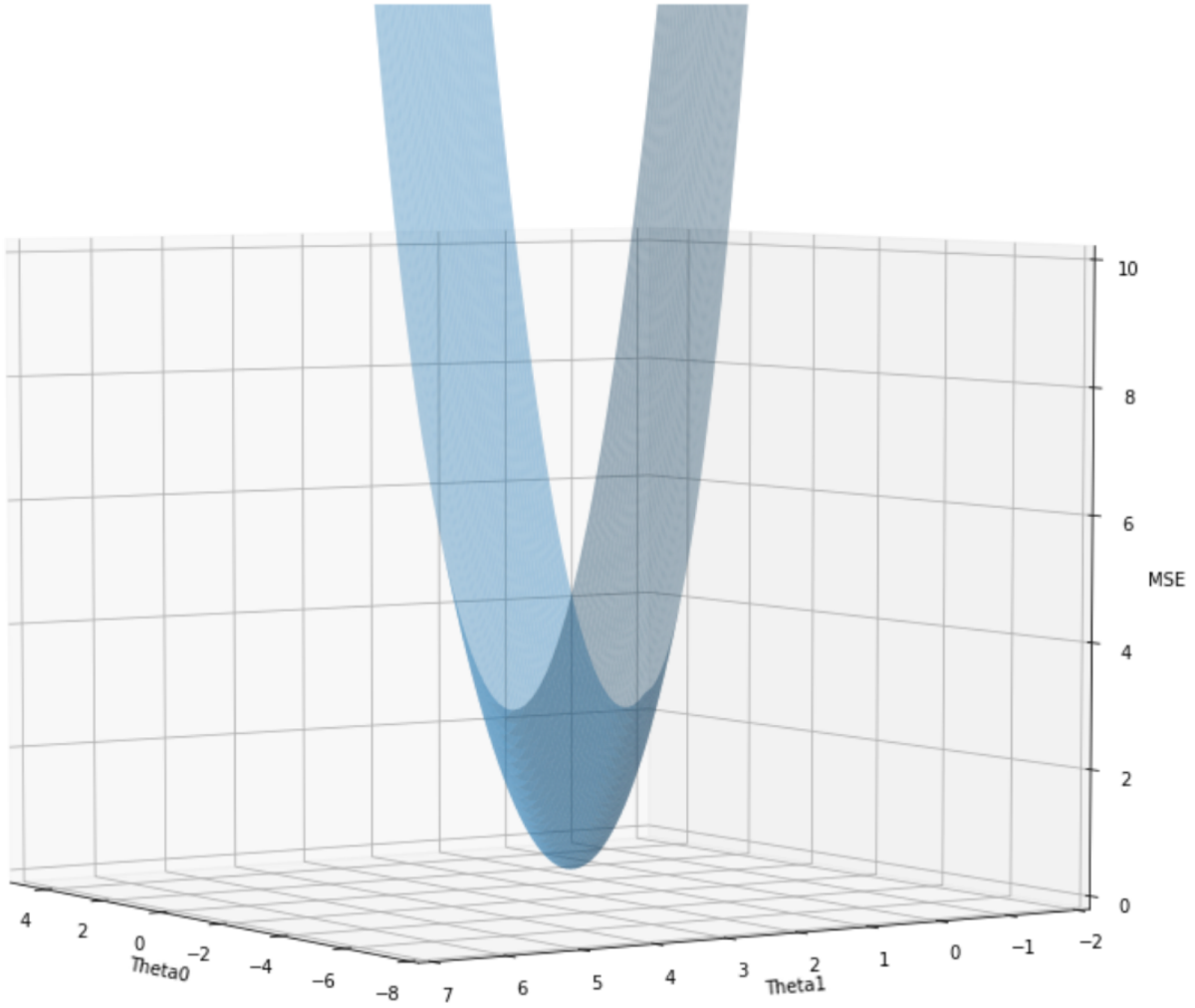
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

Minimum values are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$



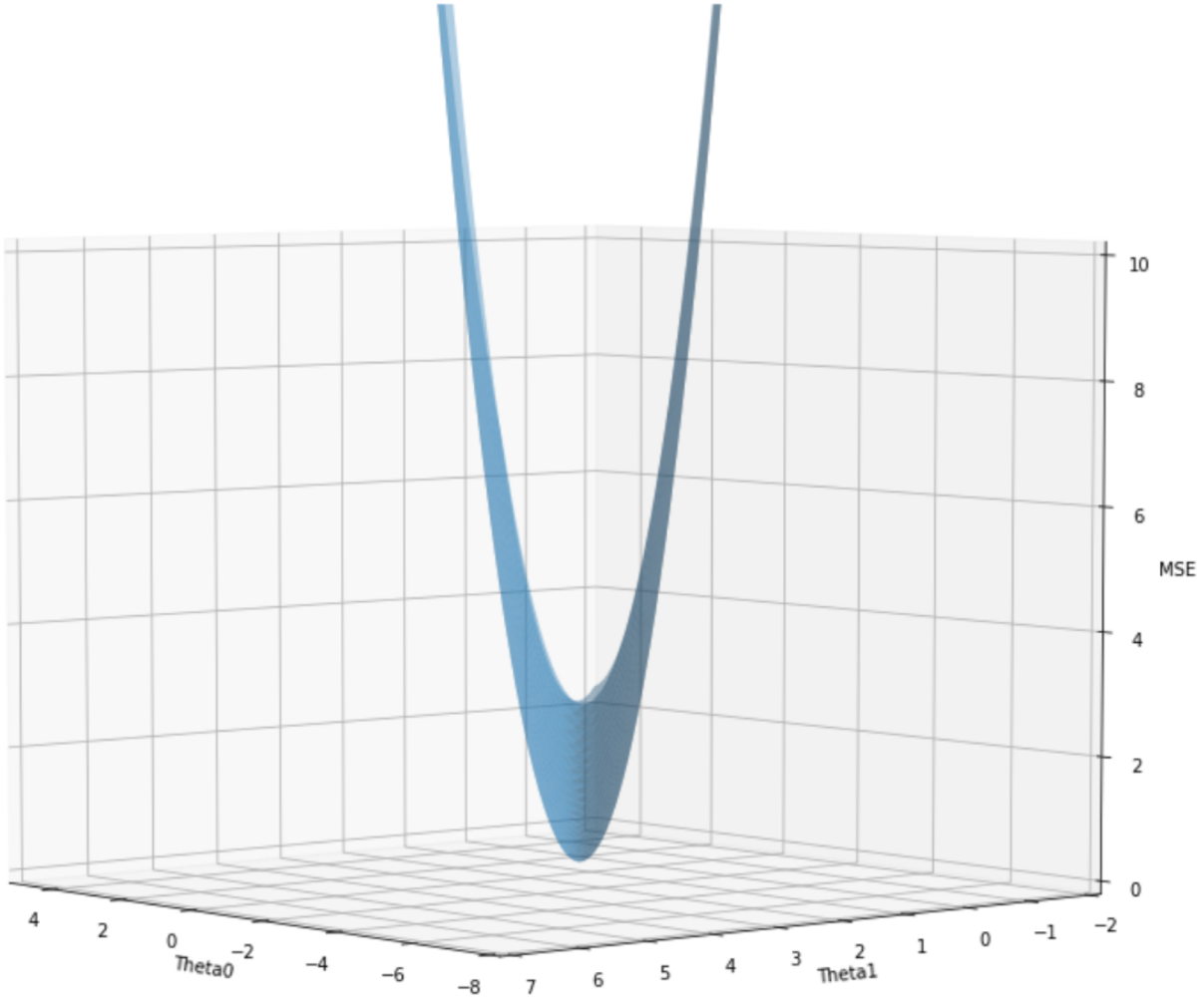
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

Minimum values are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$



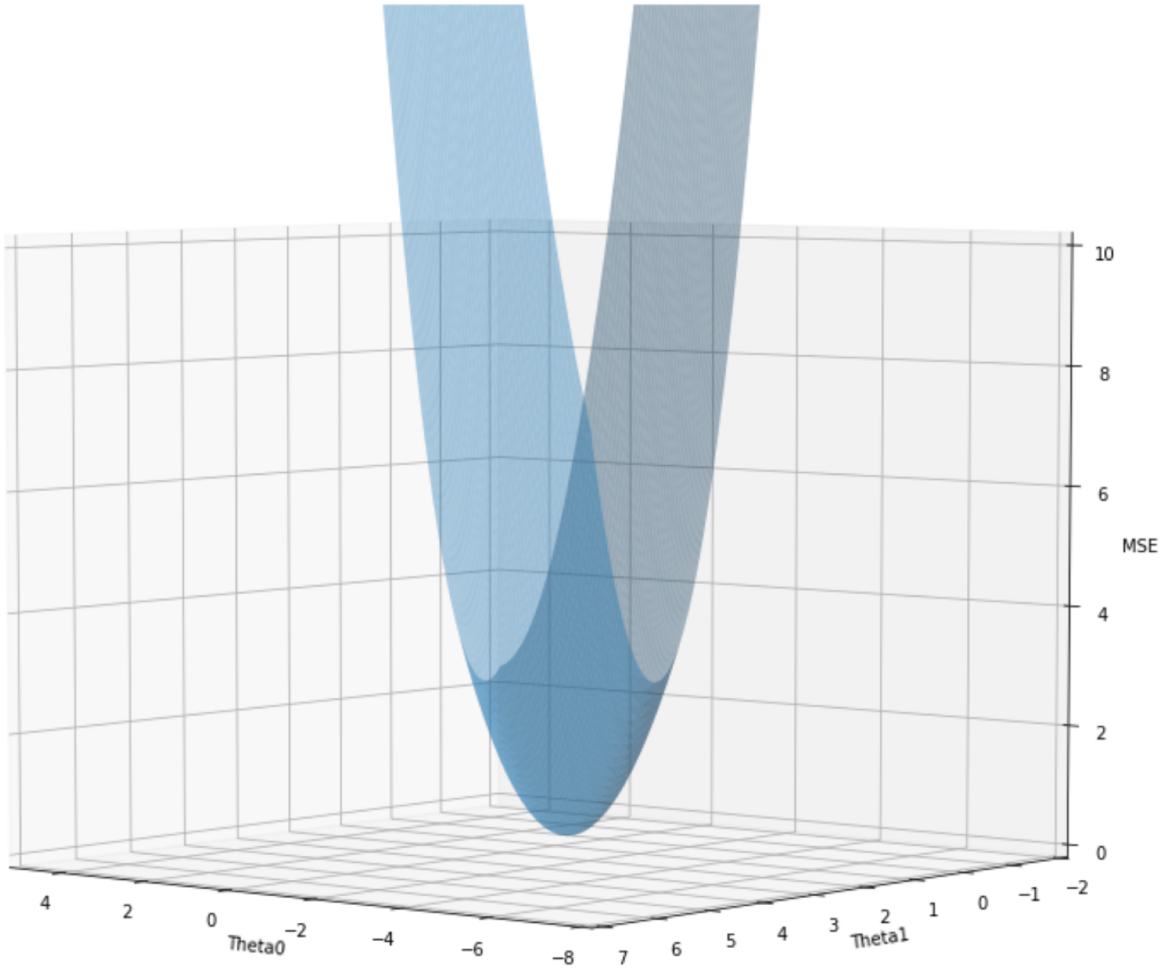
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

Minimum values are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$



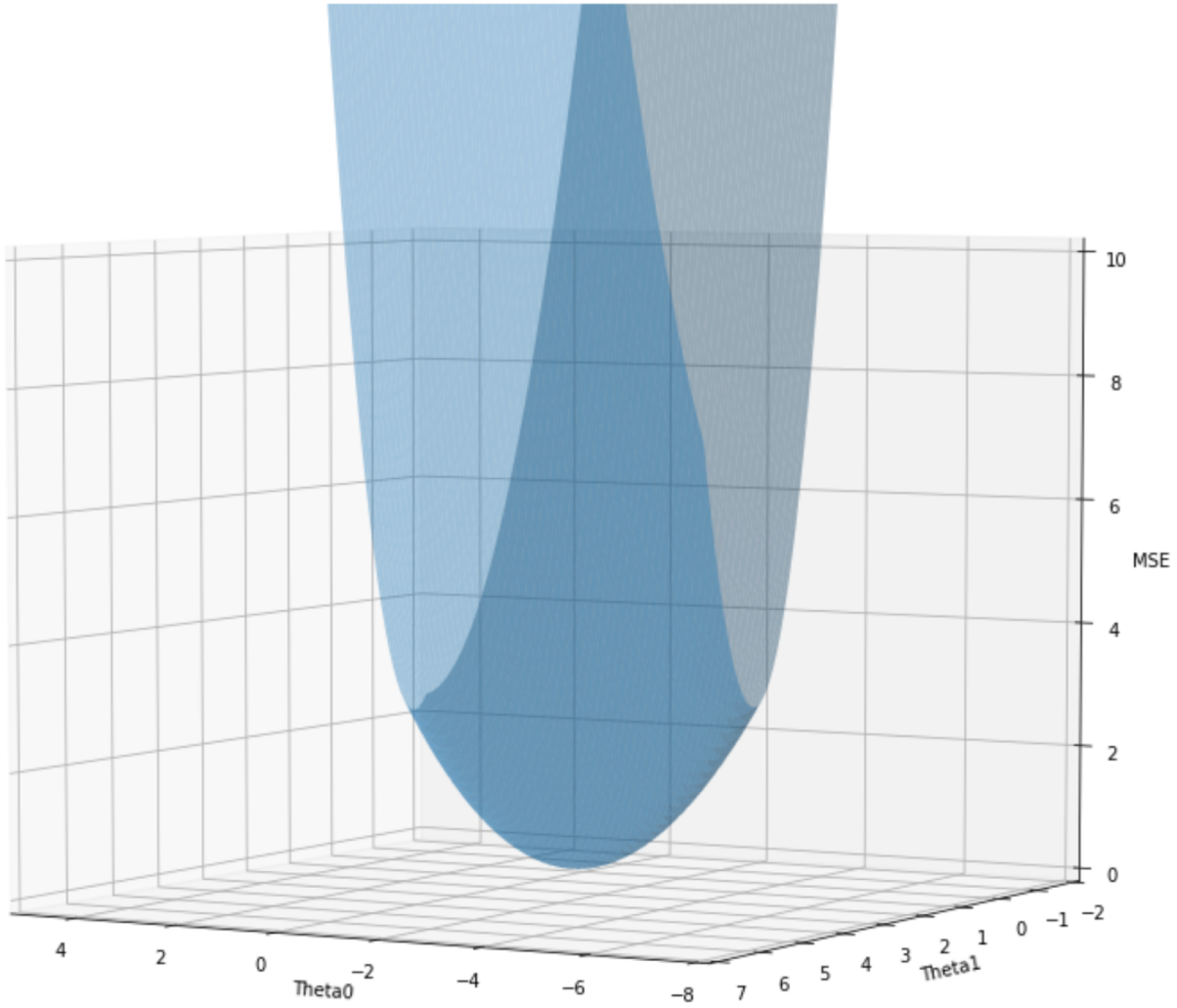
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

Minimum values are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$



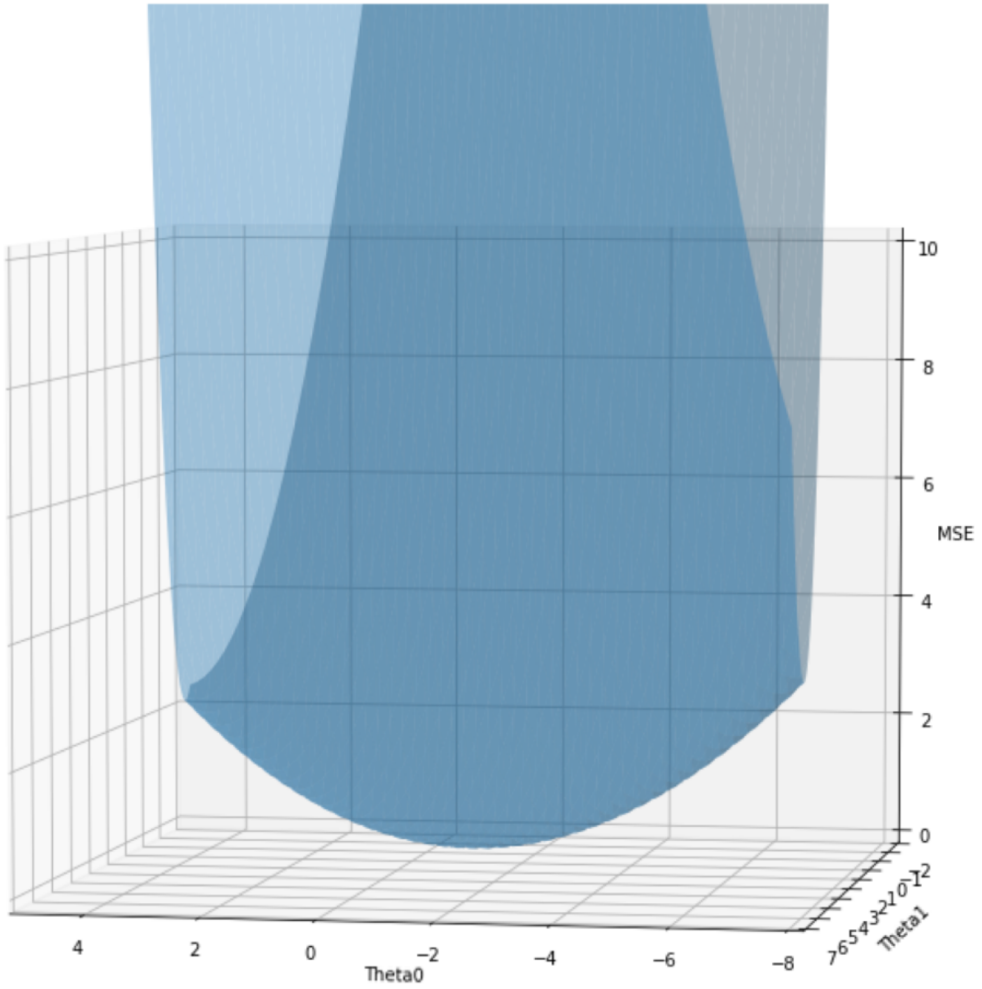
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

Minimum values are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$



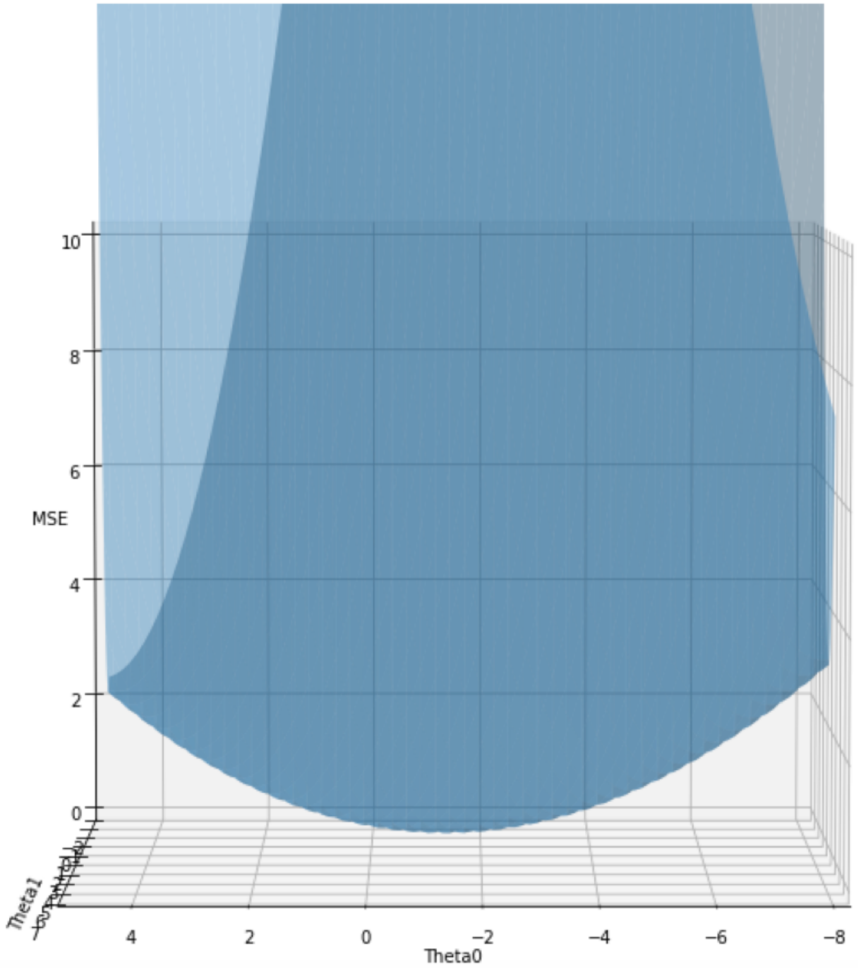
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Since there is a squared value involved, it is not too surprising that we are seeking the minimum of a 3D paraboloid. Let us plot the MSE as we vary the values of $\hat{\theta}_0$ and $\hat{\theta}_1$:

Minimum values are:

$\hat{\theta}_1 = 1.5455$

$\hat{\theta}_0 = -0.6364$

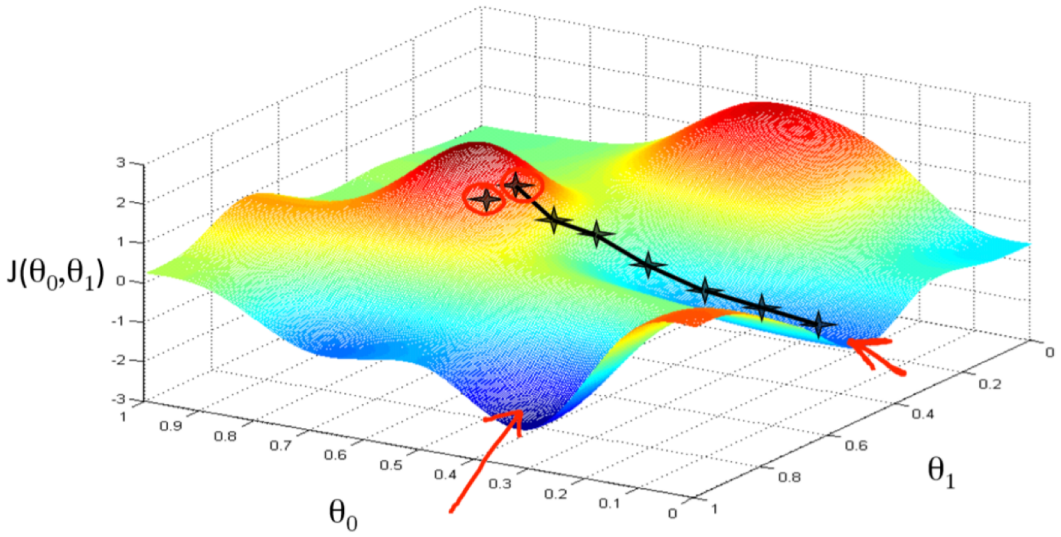


Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

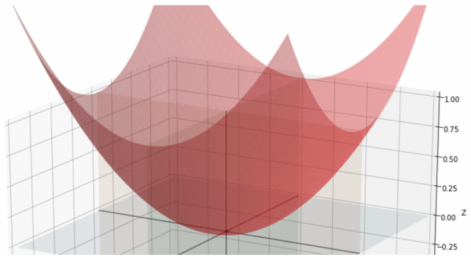
The Gradient Descent Algorithm: A **gradient** is a generalization of a derivative to functions of more than one variable:

> "Like the derivative, the gradient represents the slope of the tangent of the graph of the function. More precisely, the gradient points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the graph in that direction." - Wikipedia

In gradient descent, we pick a place to start, and move **down** the gradient until we find a minimum point:



When the search space is convex, such as a paraboloid, there will be a single minimum!



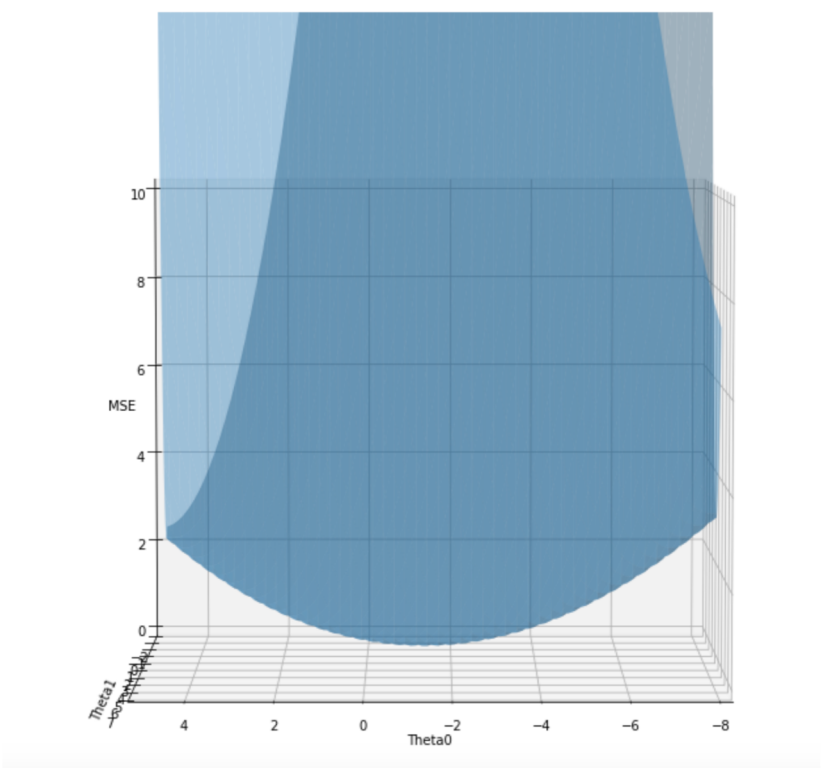
Another nice summary: https://hackernoon.com/gradient-descent-aynk-7cbe95a778da

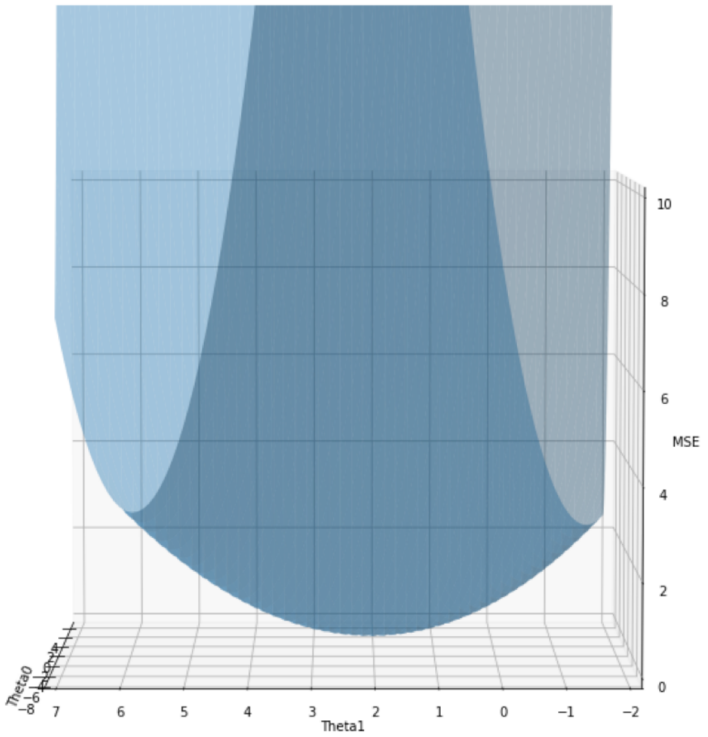
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Calculating the Gradient: We must calculate the partial derivatives of the cost function with respect to each of the parameters we are trying to minimize:

$$J(\hat{\theta}_0, \hat{\theta}_1) = \frac{1}{N} \sum_{i=1}^{N} (y_i - (\hat{\theta}_0 + \hat{\theta}_1 x_i))^2$$

The partial derivatives are just the slopes of the 2D graphs of each parameter considered in isolation, considering all other parameters are constants, as if slicing the search space along the axis of the parameter we are investgating:



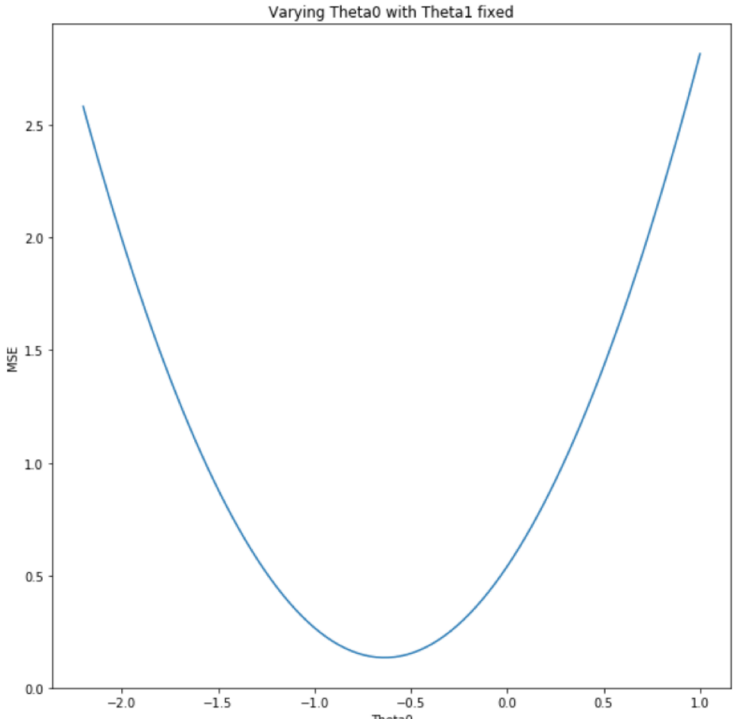


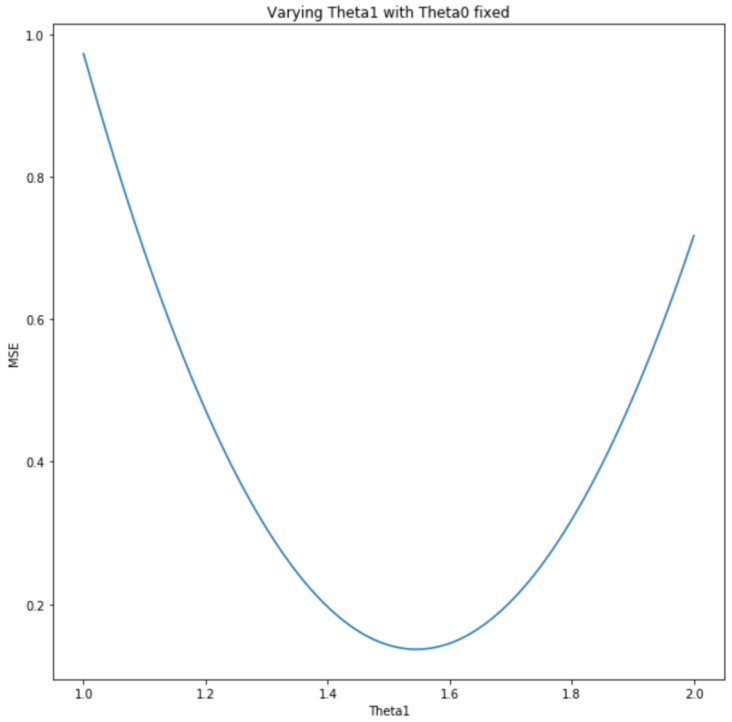
Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Calculating the Gradient: We must calculate the partial derivatives of the cost function with respect to each of the parameters we are trying to minimize:

$$J(\hat{\theta}_0, \hat{\theta}_1) = \frac{1}{N} \sum_{i=1}^{N} (y_i - (\hat{\theta}_0 + \hat{\theta}_1 x_i))^2$$

The partial derivatives are just the slopes of the 2D graphs of each parameter considered in isolation, considering all other parameters are constants, as if slicing the search space along the axis of the parameter we are investgating:





Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

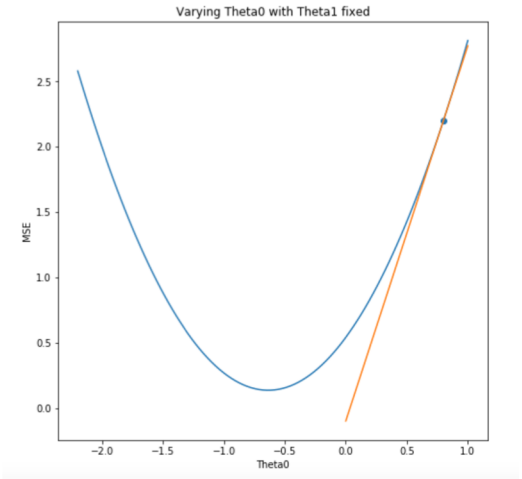
Partial Derivatives: To find a partial derivative of a function with multiple parameters, we find the rate at which each parameter varies – its derivative -- in isolation, considering each of the other parameters to effectively be a constant:

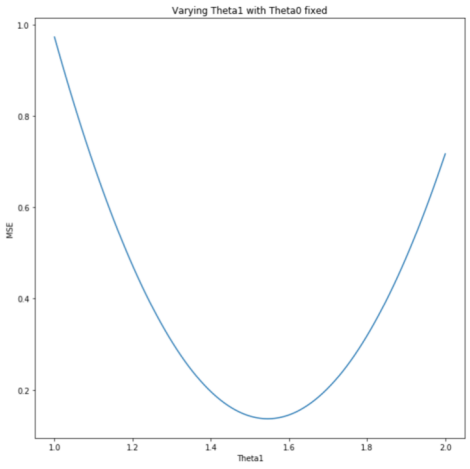
$$J(b,m) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (b + mx_i))^2$$

$$J'(b,m) = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} -2(y_i - (b + mx_i)) \\ \frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i - (b + mx_i)) \end{bmatrix}$$

We will use b and m instead of $\hat{\theta}_0$ and $\hat{\theta}_1$ and will represent points as column vectors:

$$(b,m) = \begin{bmatrix} b \\ m \end{bmatrix}$$





Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

To find the minimum value along one axis we will work with only one of the partial derivatives as a time, say the y-intercept:

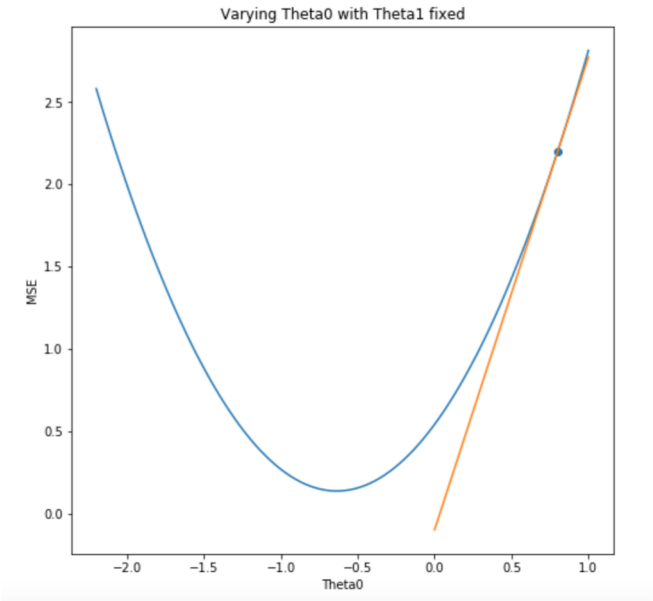
$$J'(b, m) = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} -2(y_i - (b + mx_i)) \\ \frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i - (b + mx_i)) \end{bmatrix}$$

Step One: Choose an initial point b_0.

Step Two: Choose a step size or learning rate λ and threshold of accuracy ε.

Step Three: Move that distance along the axis, in the decreasing direction (the negative of the slope), and repeat until the distance moved is less than ε.

Step Four: Output b_{n+1} as the minimum.



1. Choose b_0;
2. Choose λ;
3. Repeat $b_{n+1} = b_n - J'(b_n) \cdot \lambda$
   Until $|b_{n+1} - b_n| < \epsilon$
4. Output b_{n+1}.

Let's look at a notebook showing this...

Linear Regression Redux: Gradient Descent to find $\hat{\theta}_0$ and $\hat{\theta}_1$

Gradient Descent for Linear Regression:

To find a point in multiple dimensions, we simply do all dimensions in the same way at the same time. Here is the algorithm from the reading:

```
def update_weights(m, b, X, Y, learning_rate):
    m_deriv = 0
    b_deriv = 0
    N = len(X)
    for i in range(N):
        # Calculate partial derivatives
        # -2x(y - (mx + b))
        m_deriv += -2*X[i] * (Y[i] - (m*X[i] + b))

        # -2(y - (mx + b))
        b_deriv += -2*(Y[i] - (m*X[i] + b))

    # We subtract because the derivatives point in direction of steepest ascent
    m -= (m_deriv / float(N)) * learning_rate
    b -= (b_deriv / float(N)) * learning_rate

    return m, b
```