**Preparing to batch train**

Before we can train a linear model in batches, we must first define variables, a loss function, and an optimization operation. In this exercise, we will prepare to train a model that will predict price\_batch, a batch of house prices, using size\_batch, a batch of lot sizes in square feet. In contrast to the previous lesson, we will do this by loading batches of data using pandas, converting it to numpy arrays, and then using it to minimize the loss function in steps.

Variable(), keras(), and float32 have been imported for you. Note that you should not set default argument values for either the model or loss function, since we will generate the data in batches during the training process.

**Instructions**

**100 XP**

* Define intercept as having an initial value of 10.0 and a data type of 32-bit float.
* Define the model to return the predicted values using intercept, features, and slope.
* Define a function called loss\_function() that takes intercept, slope, targets, and features as arguments. Do not set default argument values.
* Define the mean squared error loss function using targets and predictions.

# Define the intercept and slope

intercept = Variable(10.0, float32)

slope = Variable(0.5, float32)

# Define the model

def linear\_regression(intercept, slope, features):

# Define the predicted values

return intercept + features\*slope

# Define the loss function

def loss\_function(intercept, slope, targets, features):

# Define the predicted values

predictions = linear\_regression(intercept, slope, features)

# Define the MSE loss

return keras.losses.mse(targets, predictions)

Excellent work! Notice that we did not use default argument values for the input data, features and targets. This is because the input data has not been defined in advance. Instead, with batch training, we will load it during the training process.