**Set up a linear regression**

A univariate linear regression identifies the relationship between a single feature and the target tensor. In this exercise, we will use a property's lot size and price. Just as we discussed in the video, we will take the natural logarithms of both tensors, which are available as price\_log and size\_log.

In this exercise, you will define the model and the loss function. You will then evaluate the loss function for two different values of intercept and slope. Remember that the predicted values are given by intercept + features\*slope. Additionally, note that keras.losses.mse() is available for you. Furthermore, slope and intercept have been defined as variables.

**Instructions**

**100 XP**

* Define a function that returns the predicted values for a linear regression using intercept, features, and slope, and without using add()or multiply().
* Complete the loss\_function() by adding the model's variables, intercept and slope, as arguments.
* Compute the mean squared error using targets and predictions.

# Define a linear regression model

def linear\_regression(intercept, slope, features = size\_log):

return (intercept + features\*slope)

# Set loss\_function() to take the variables as arguments

def loss\_function(intercept, slope, features = size\_log, targets = price\_log):

# Set the predicted values

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

# Return the mean squared error loss

return keras.losses.mse(targets, predictions)

# Compute the loss for different slope and intercept values

print(loss\_function(0.1, 0.1).numpy())

print(loss\_function(0.1, 0.5).numpy())