

Below, use the slider control to select the value of w that minimizes cost. It can take a few seconds for the plot to update.

In [4]: `plt_intuition(x_train,y_train)`

w 150

Figure 1

Minimize Cost: Current Cost = 3125



The plot contains a few points that are worth mentioning.

- cost is minimized when $w = 200$, which matches results from the previous lab
- Because the difference between the target and prediction is squared in the cost equation, the cost increases rapidly when w is either too large or too small.
- Using the w and b selected by minimizing cost results in a line which is a perfect fit to the data.

Cost Function Visualization- 3D

You can see how cost varies with respect to *both* w and b by plotting in 3D or using a contour plot.

It is worth noting that some of the plotting in this course can become quite involved. The plotting routines are provided and while it can be instructive to read through the code to become familiar with the methods, it is not needed to complete the course successfully. The routines are in `lab_utils_uni.py` in the local directory.

Larger Data Set

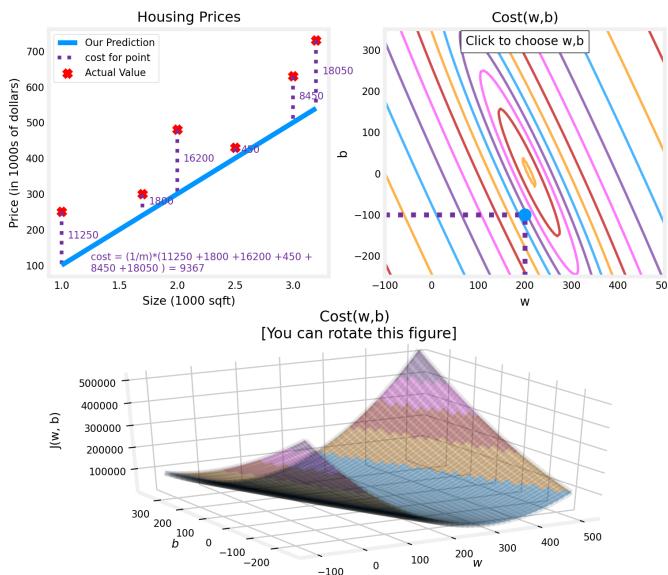
It is instructive to view a scenario with a few more data points. This data set includes data points that do not fall on the same line. What does that mean for the cost equation? Can we find w , and b that will give us a cost of 0?

In [5]: `x_train = np.array([1.0, 1.7, 2.0, 2.5, 3.0, 3.2])
y_train = np.array([250, 300, 480, 430, 630, 730,])`

In the contour plot, click on a point to select w and b to achieve the lowest cost. Use the contours to guide your selections. Note, it can take a few seconds to update the graph.

In [6]: `plt.close('all')
fig, ax, dyn_items = plt_stationary(x_train, y_train)
updater = pit_update_onclick(fig, ax, x_train, y_train, dyn_items)`

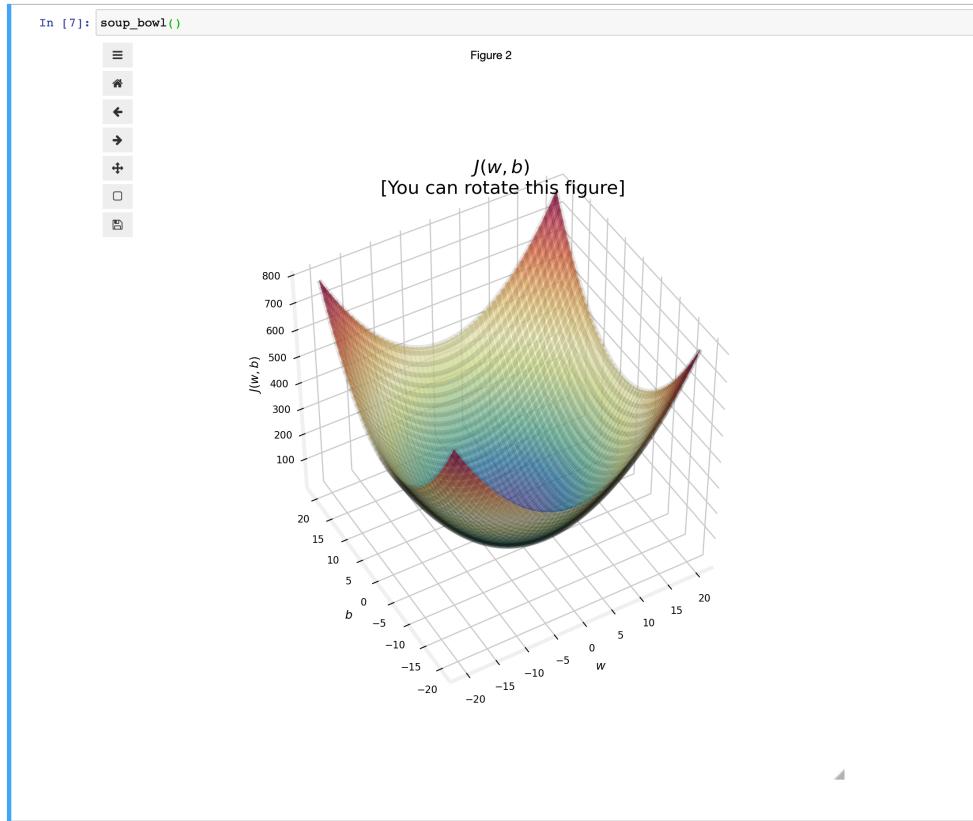
Figure 1



Above, note the dashed lines in the left plot. These represent the portion of the cost contributed by each example in your training set. In this case, values of approximately $w = 209$ and $b = 2.4$ provide low cost. Note that, because our training examples are not on a line, the minimum cost is not zero.

Convex Cost surface

The fact that the cost function squares the loss ensures that the 'error surface' is convex like a soup bowl. It will always have a minimum that can be reached by following the gradient in all dimensions. In the previous plot, because the w and b dimensions scale differently, this is not easy to recognize. The following plot, where w and b are symmetric, was shown in lecture:



Congratulations!

You have learned the following:

- The cost equation provides a measure of how well your predictions match your training data.
- Minimizing the cost can provide optimal values of w , b .

In []: