1. Which of the following is **NOT** a **linear** regression model:

**d. y = w\_0 \* w\_1 + log(w\_1) \* x**

1. Your estimated model for predicting house prices has a large positive weight on 'square feet living'. This implies that if we remove the feature 'square feet living' and refit the model, the new predictive performance will be **worse** than before.

**true / false**

1. *Complete the following:* Your estimated model for predicting house prices has a positive weight on 'square feet living'. You then add 'lot size' to the model and re-estimate the feature weights. The new weight on 'square feet living' [\_\_\_\_\_\_\_\_\_] be positive.

**will definitely / might**

1. If you double the value of a given feature (i.e. a specific column of the feature matrix), what happens to the least-squares estimated coefficients for every **other** feature? (assume you have no other feature that depends on the doubled feature i.e. no interaction terms).

**d. It is impossible to tell from the information provided / c. They stay the same**

1. Gradient descent/ascent is…

**b. An algorithm for minimizing/maximizing a function**

1. Gradient descent/ascent allows us to…

**b. Estimate model parameters from data**

1. Which of the following statements about step-size in gradient descent is/are **TRUE** (select all that apply)

It's important to choose the smallest step-size possible

The step-size doesn't matter

If the step-size is too large gradient descent may not converge

If the step-size is too small (but not zero) gradient descent may not converge

**If the step size is too small (but not zero) gradient descent may take a very long time to converge**

It's important to choose the smallest step-size possible

The step-size doesn't matter

**If the step-size is too large gradient descent may not converge**

If the step-size is too small (but not zero) gradient descent may not converge

**If the step size is too small (but not zero) gradient descent may take a very long time to converge**

1. Let's analyze how many computations are required to fit a multiple linear regression model *using the closed-form solution* based on a data set with 50 observations and 10 features. In the videos, we said that computing the inverse of the 10x10 matrix (H^T)H was on the order of D^3 operations. Let's focus on forming this matrix **prior** to inversion. How many multiplications are required to form the matrix (H^T)H?

**500 / 100 / 5000**

1. More generally, if you have D features and N observations what is the total complexity of computing ((H^T)H)^(-1)?

O(D^3)

O(ND^3)

O(ND^2 + D^3)

O(ND^2)

O(N^2D + D^3)

O(N^2D)

**a. O(D^3) / d. O(ND^2) / e. O(N^2D + D^3)** / **c. O(ND^2 + D^3)**