

# Lecture 1: Introduction to regression and prediction

Rafael A. Irizarry and Hector Corrada Bravo

January, 2010

## Introduction

A common situation in applied sciences is that one has an independent variable or outcome  $Y$  and one or more dependent variables or covariates  $X_1, \dots, X_p$ . One usually observes these variables for multiple “subjects”.

Note: We use upper case to denote a random variable. To denote actual numbers we use lower case. One way to think about it:  $Y$  has not happened yet, and when it does, we see  $Y = y$ .

One may be interested in various things: What effects do the covariates have on the outcome? How well can we describe these effects? Can we predict the outcome using the covariates?, etc. . .

## Linear Regression

Linear regression is the most common approach for describing the relation between predictors (or covariates) and outcome. Here we will see how regression relates to prediction.

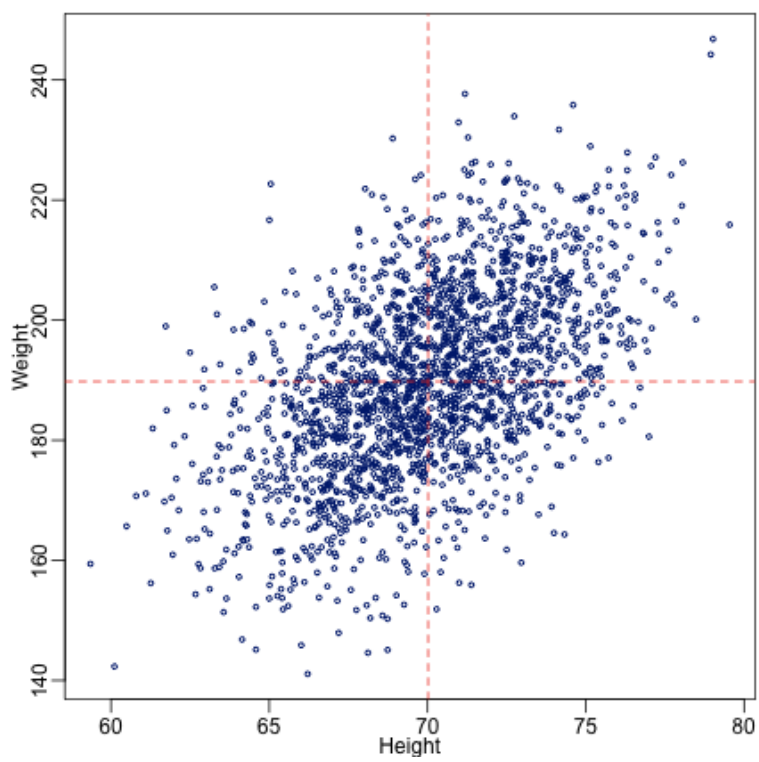
Let’s start with a simple example. Let’s say we have a random sample of US males and we record their heights ( $X$ ) and weights ( $Y$ ).

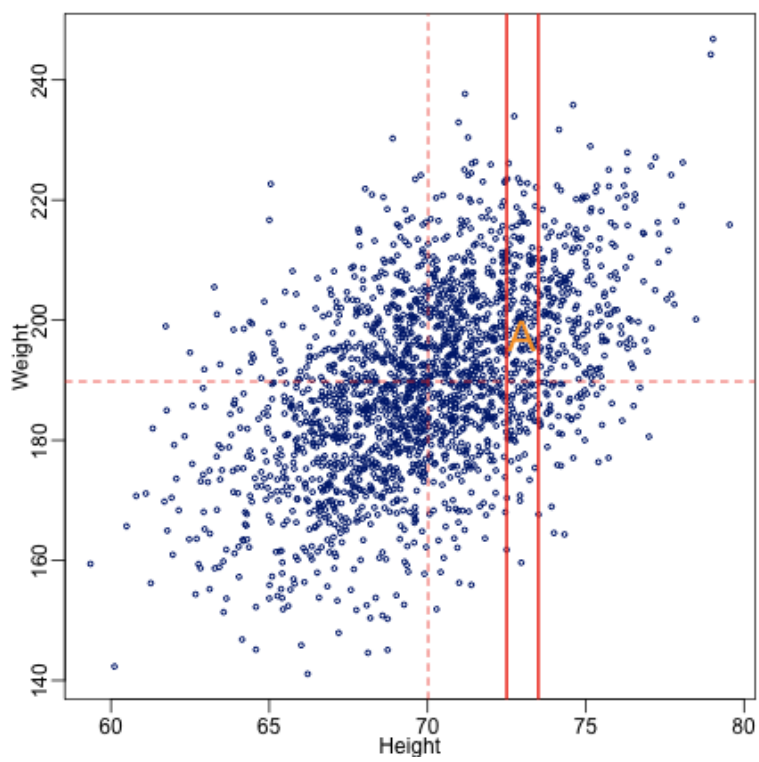
Say we pick a random subject. How would you predict their weight?

What if I told you their height? Would your strategy for predicting change?

We can show mathematically that for a particular definition of “best”, described below, the average is the best predictor of a value picked from that population. However, if we have information about a related variable, then the conditional average is best.

One can think of the conditional average as the average weights for all men of a particular height.





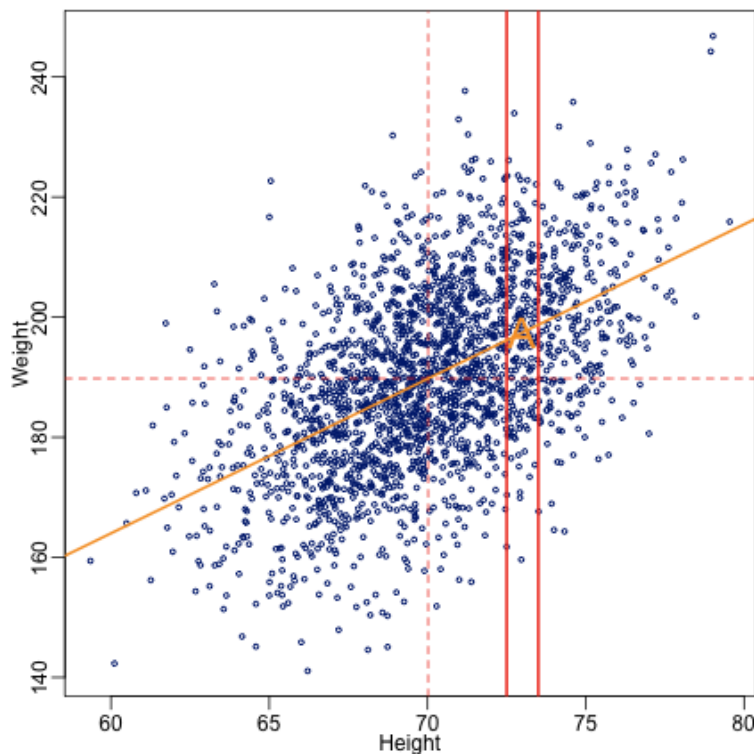
In the case of weight and height, the data actually look bivariate normal (football shaped) and one can show that the best predictor (the conditional average) of weight given height is

$$E[Y|X = x] = \mu_Y + \rho \frac{\sigma_Y}{\sigma_X} (x - \mu_X)$$

with  $\mu_X = E[X]$  (average height),  $\mu_Y = E[Y]$  (average weight), and where  $\rho$  is the correlation coefficient of height and weight.

If we obtain a random sample of the data, then each of the above parameters is substituted by the sample estimates and we get a familiar expression:

$$\hat{Y}(x) = \bar{Y} + r \frac{SD_Y}{SD_X} (x - \bar{X}).$$



Technical note: Because in practice it is useful to describe distributions of populations with continuous distributions we will start using the word *expectation* or the phrase *expected value* instead of average. We use the notation  $E[\cdot]$ . If you think of integrals as sums, then you can think of expectations as averages.

Notice that equation (1.1) can be written in this, more familiar, notation:

$$E[Y|X = x] = \beta_0 + \beta_1 x.$$

Because the conditional distribution of  $Y$  given  $X$  is normal, then we can write the even more familiar version:

$$Y = \beta_0 + \beta_1 X + \epsilon,$$

with  $\epsilon$  a mean 0, normally distributed random variable that is independent of  $X$ . This notation is popular in many fields because  $\beta_1$  has a nice interpretation and its typical (least squares) estimate has nice properties.

When more than one predictor exists, it is quite common to extend this linear regression model to the multiple linear regression model:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

with  $\epsilon$  as unbiased (0 mean) error independent of the  $X_j$  as before.

A drawback of these models is that they are quite restrictive. Linearity and additivity are two very strong assumptions. This may have practical consequences. For example, by assuming linearity one may never notice that a covariate has an effect that increases and then decreases. We will see various examples of this in class.

Linear regression is popular mainly because of the interpretability of the parameters. However, the interpretation only makes sense if the model is an appropriate approximation of the natural data generating process. It is likely that the linear regression model from a randomly selected publication will do a terrible job at predicting results in data where the model was not trained on. Prediction is not really given much importance in many scientific fields, e.g. Epidemiology and Social Sciences. In other fields, e.g. Surveillance, Finance and web-commerce is everything. Notice that in the fields where prediction is important, linear regression is not as popular.

## Prediction

Methods for prediction can be divided into two general groups: continuous and discrete outcomes.

When the data is discrete we will refer to it as *classification*. Other terms are *discriminant analysis*, *pattern recognition*.

When the data is continuous we will refer to it as *regression*. Other terms are *smoothing* and *curve estimation*.

These seem very different but they have some in common. In this class, we will talk about the commonalities, but in general, we will discuss these two cases separately.

The main common characteristic in both cases is that we observe predictors  $X_1, \dots, X_p$  and we want to predict the outcome  $Y$ .

Note: I will use  $X$  to denote the vector of all predictors. So,  $X_i$  are the predictors for the  $i$ -th subject and can include age, gender, ethnicity, etc.

Note: Given a prediction method we will use  $f(x)$  to denote the prediction we get if the predictors are  $X = x$ .

Q: What are examples of prediction problems?

So, what does it mean to predict well? Let's look at the continuous data case first.

If I have a prediction  $f(X)$  based on predictors  $X$ , how do I define a "good prediction" mathematically. A common way of defining closeness is with Euclidean distance:

$$L\{Y, f(X)\} = \{Y - f(X)\}^2.$$

We sometime call this the *loss function*.

Notice that because both  $Y$  and  $f(X)$  are random variables, so is (2.2). Minimizing a random variable is meaningless because it is not a number. A common thing to do is minimize over the average loss, or the **expected prediction error**:

$$E_X E_{Y|X}[\{Y - f(X)\}^2 | X].$$

For a given  $x$ , the expected loss is minimized by the conditional expectation:

$$f(x) = E[Y | X = x],$$

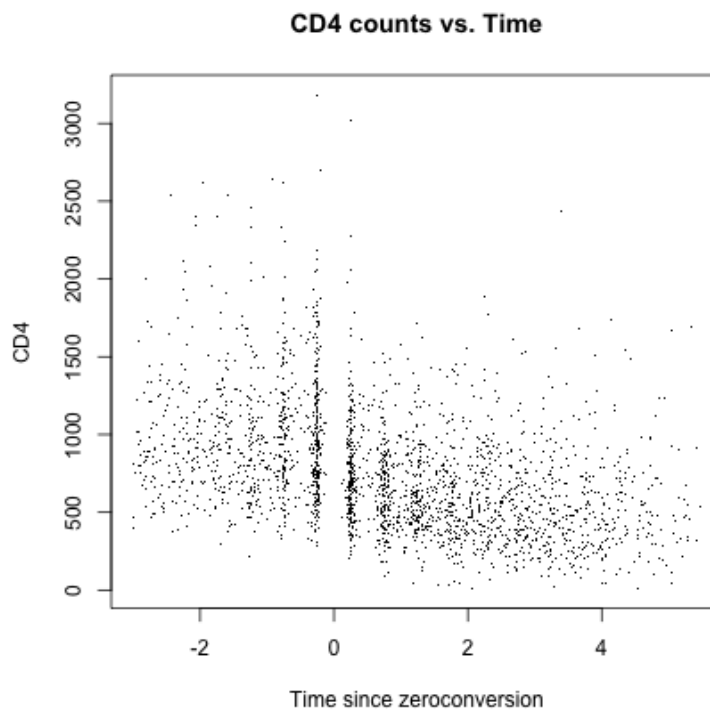
so it all comes down to getting a good estimate of  $E[Y | X = x]$ . We usually call  $f(x)$  the *regression function*.

Note: For discrete problems we usually want a plausible prediction. Note  $f(x)$  is typically a continuous number and not a class. We can take an extra step and define a prediction rule. For example, for binary outcomes, we can say: if  $f(x) > 0.5$ , I predict a 1, otherwise, predict 0. However, it is useful to change the loss function. More on this later.

Notice that if the regression model holds, then

$$f(X) = E[Y|X_1 = x_1, \dots, X_p = x_p] = \beta_0 + \sum_{j=1}^p \beta_j x_j.$$

For Gaussian models, the solution is the same for least squares and MLE. However, many times, it is hard to believe that the linear regression model holds. A simple example comes from AIDS research.



Technical note: It should be noted that for some designed experiments it does not make sense to assume the  $X$  are random variables. In this case, we usually assume we have “design points”  $x_{1i}, \dots, x_{pi}, i = 1, \dots, n$  and non-IID observations  $Y_1, \dots, Y_n$  for each design point. In most cases, the theory for both cases is very similar if not the same. These are called *random design model* and *fixed design model* respectively.

## Other settings

A major focus of this class is prediction, or *supervised learning*. However, we will also see a few other learning settings. For instance, suppose we only observe vectors of random variables,  $X_1, \dots, X_p$  but no outcome  $Y$ ? In this case we still want to find some informative structure (e.g. *clustering*). This setting is called *unsupervised learning*. We can include probability density estimation under this setting.

We will also spend a little time looking at settings where along with outcomes  $Y$  we don't observe vectors of covariates, but rather other, harder to represent structures (e.g. strings, graphs). For example, suppose you want to classify proteins from their amino acid sequence.

Finally, we might take a look at settings where we do not receive training data (pairs of covariate vectors  $X$  and outcomes  $Y$ ) at once, or in a *batch*. One such setting is called *online learning*, where training data is obtained one point at a time, and we want to minimize **expected prediction error** over the entire sequence of training data. Think of google building a model of your web-searching profile.