

Machine Learning

Mohamad Ghassany

2018-01-09

Contents

Welcome	5
Course Overview	5
Introduction	7
What is Machine Learning ?	7
Supervised Learning	7
Unsupervised Learning	13
 I Supervised Learning	 17
1 Linear Regression	19
1.1 Notation	19
1.2 Model Representation	20
1.3 Why Estimate f ?	22
Prediction	22
Inference	23
1.4 Simple Linear Regression Model	23
1.5 Estimating the Coefficients	23
1.6 Assessing the Accuracy of the Coefficient Estimates	24
Hypothesis testing	25
1.7 ANOVA and model fit	25
1.7.1 ANOVA	25
1.7.2 The R^2 Statistic	28

Welcome

Welcome to this course. It is only a little introduction to Machine Learning.

The aim of Machine Learning is to build computer systems that can adapt to their environments and learn from experience. Learning techniques and methods from this field are successfully applied to a variety of learning tasks in a broad range of areas, including, for example, spam recognition, text classification, gene discovery, financial forecasting. The course will give an overview of many concepts, techniques, and algorithms in machine learning, beginning with topics such as linear regression and classification and ending up with topics such as kmeans and Expectation Maximization. The course will give the student the basic ideas and intuition behind these methods, as well as a more formal statistical and computational understanding. Students will have an opportunity to experiment with machine learning techniques in R and apply them to a selected problem.

Course Overview

Introduction

What is Machine Learning ?

What is Machine Learning?

Two definitions of Machine Learning are offered. Arthur Samuel described it as: “the field of study that gives computers the ability to learn without being explicitly programmed.” This is an older, informal definition.

Tom Mitchell provides a more modern definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .”

Machine Learning is also called Statistical Learning.

Example: playing checkers.

E = the experience of playing many games of checkers

T = the task of playing checkers.

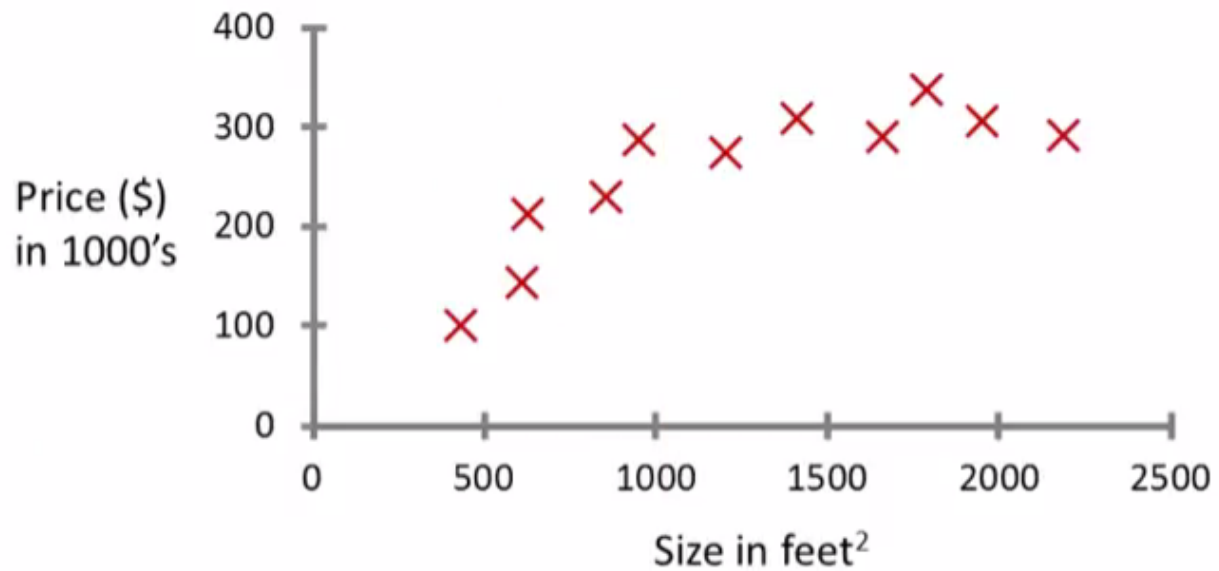
P = the probability that the program will win the next game.

In general, any machine learning problem can be assigned to one of two broad classifications:

Supervised learning and Unsupervised learning.

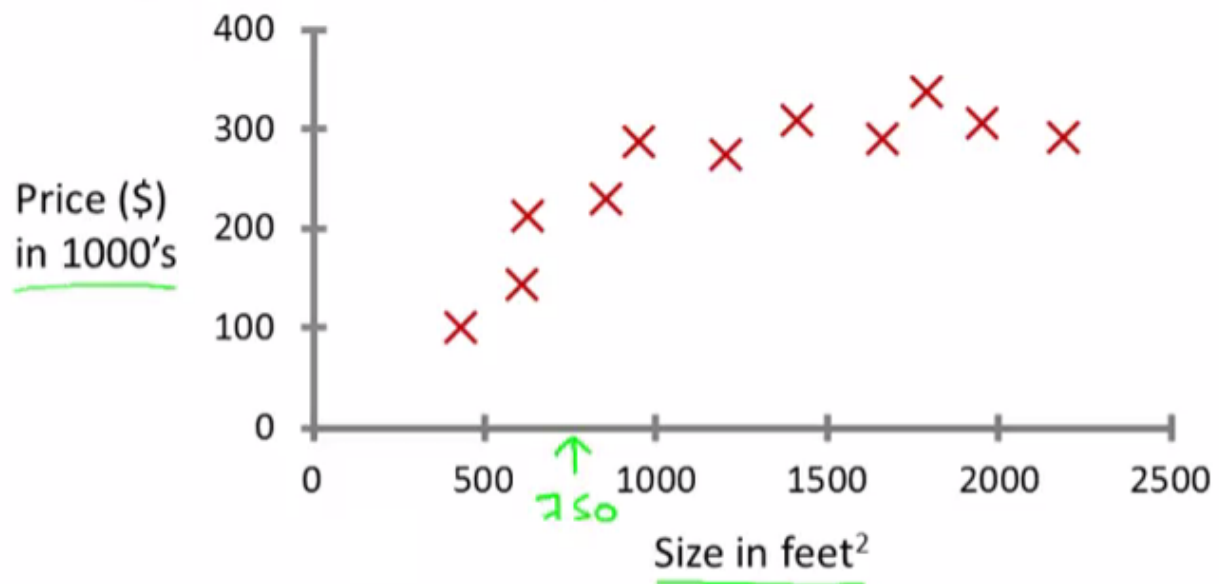
Supervised Learning

Supervised Learning is probably the most common type of machine learning problem. Let's start with an example of what is it. Let's say we want to predict housing prices. We plot a data set and it looks like this.



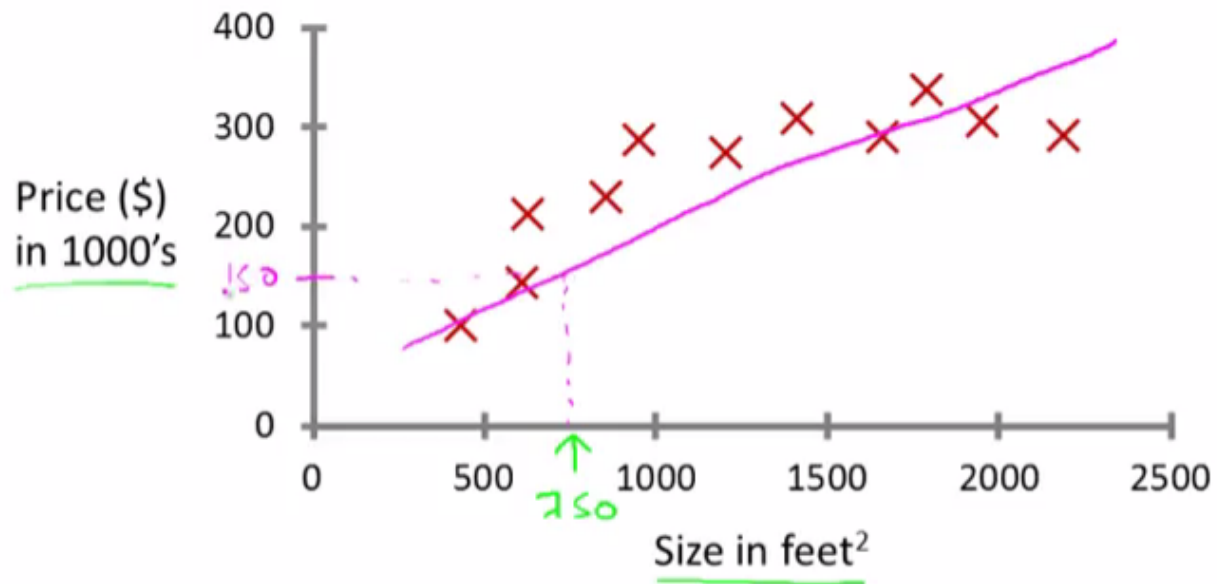
Here on the horizontal axis, the size of different houses in square feet, and on the vertical axis, the price of different houses in thousands of dollars.

So. Given this data, let's say we own a house that is, say 750 square feet and hoping to sell the house and we want to know how much we can get for the house.

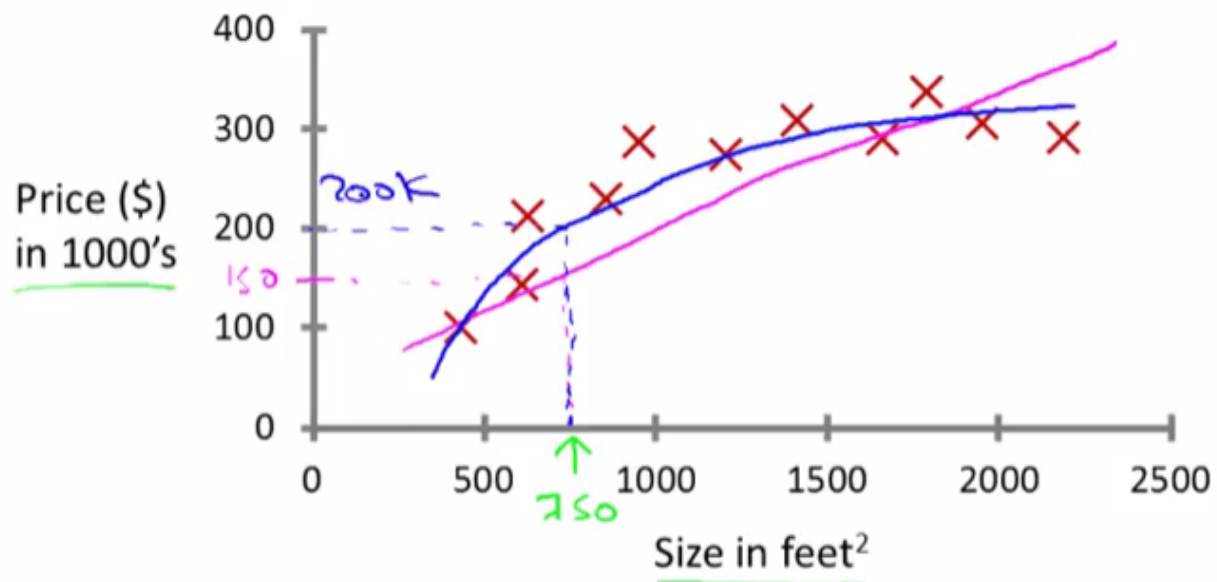


So how can the learning algorithm help?

One thing a learning algorithm might be able to do is put a straight line through the data or to “fit” a straight line to the data and, based on that, it looks like maybe the house can be sold for maybe about \$150,000.



But maybe this isn't the only learning algorithm we can use. There might be a better one. For example, instead of sending a straight line to the data, we might decide that it's better to fit a *quadratic function* or a *second-order polynomial* to this data.



If we do that, and make a prediction here, then it looks like, well, maybe we can sell the house for closer to \$200,000.

This is an example of a supervised learning algorithm.

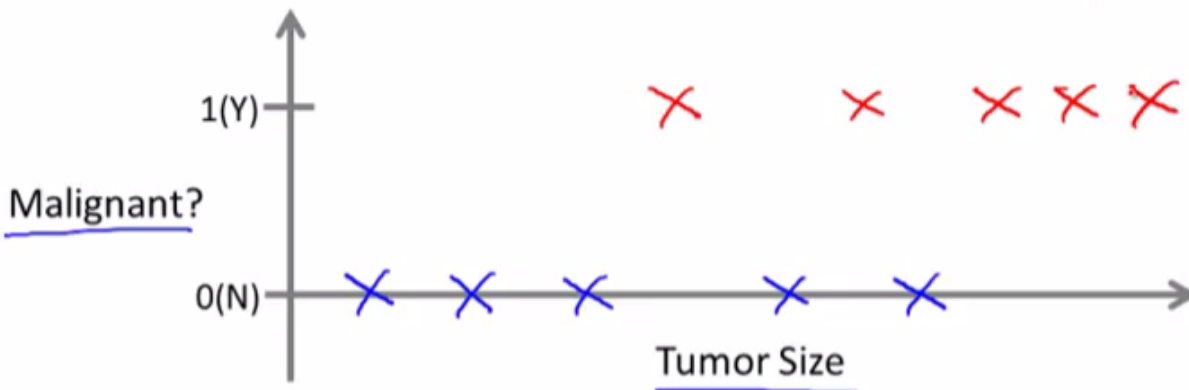
The term supervised learning refers to the fact that we gave the algorithm a data set in which the “**right answers**” were given.

The example above is also called a regression problem. A regression problem is when we try to predict a **continuous** value output. Namely the price in the example.

Here's another supervised learning example. Let's say we want to look at medical records and try to predict

of a breast cancer as malignant or benign. If someone discovers a breast tumor, a lump in their breast, a malignant tumor is a tumor that is harmful and dangerous and a benign tumor is a tumor that is harmless. Let's see a collected data set and suppose in the data set we have the size of the tumor on the horizontal axis and on the vertical axis we plot one or zero, yes or no, whether or not these are examples of tumors we've seen before are malignant (which is one) or zero if not malignant or benign.

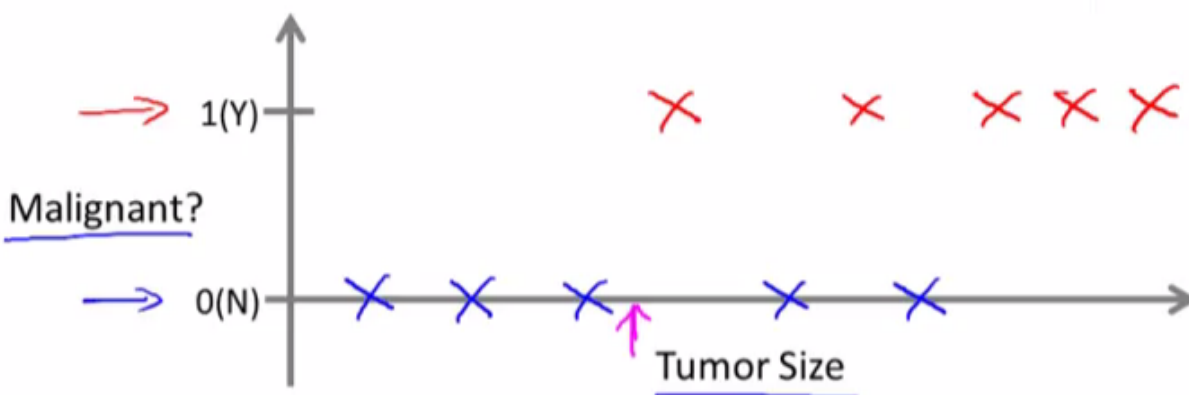
Breast cancer (malignant, benign)



In this data set we have five examples of benign tumors, and five examples of malignant tumors.

Let's say a person who tragically has a breast tumor, and let's say her breast tumor size is known (rose arrow in the following figure).

Breast cancer (malignant, benign)



The machine learning question is, can you estimate what is the probability that a tumor is malignant versus benign? To introduce a bit more terminology this is an example of a **classification** problem.

The term classification refers to the fact that here we're trying to predict a **discrete** value output: zero or one, malignant or benign. And it turns out that in classification problems sometimes you can have more than two values for the two possible values for the output.

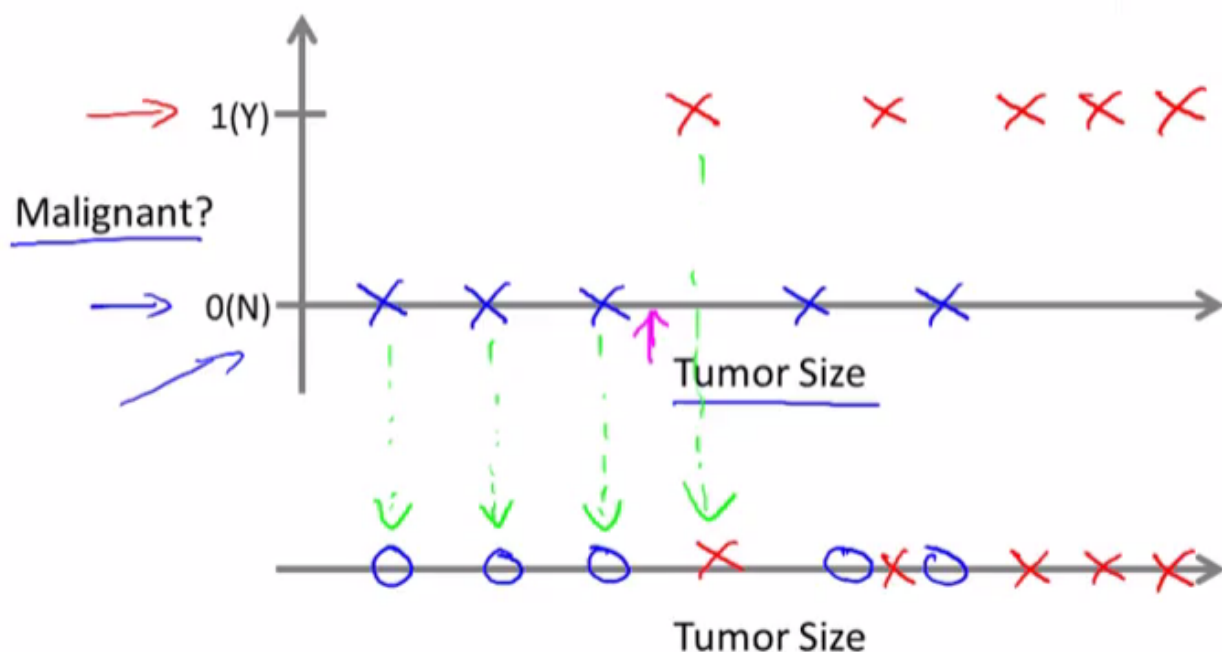
In classification problems there is another way to plot this data. Let's use a slightly different set of symbols to plot this data. So if tumor size is going to be the attribute that we are going to use to predict malignancy

or benignness, we can also draw the data like this.



All we did was we took the data set on top and just mapped it down using different symbols. So instead of drawing crosses, we are now going to draw 0's for the benign tumors.

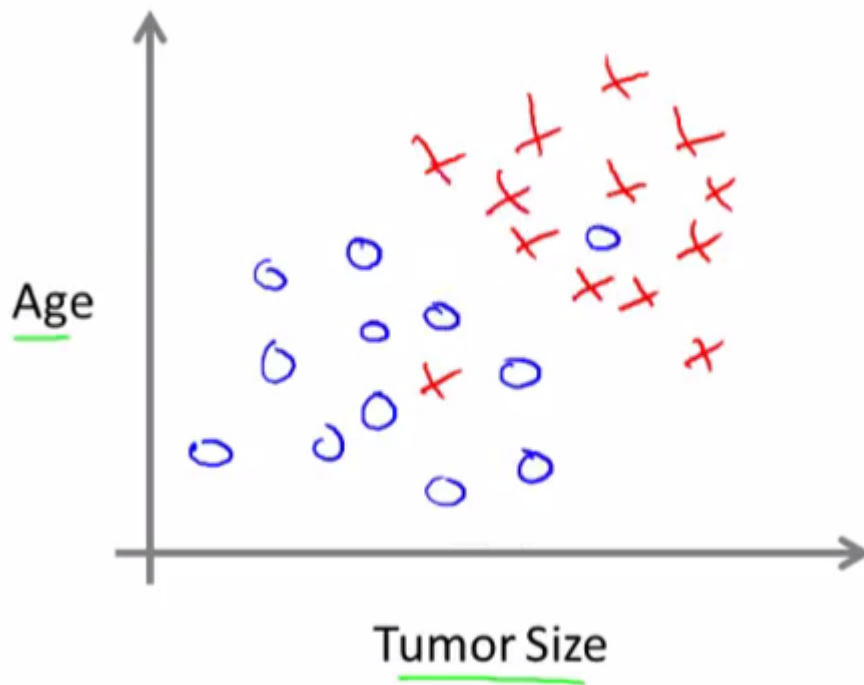
Breast cancer (malignant, benign)



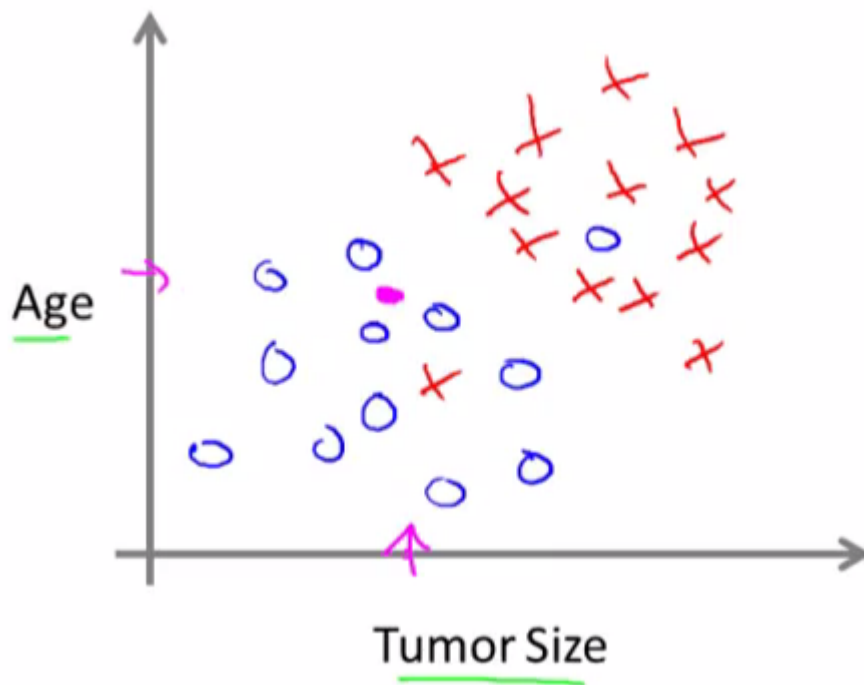
Now, in this example we use only one **feature** or one attribute, mainly, the *tumor size* in order to predict whether the tumor is malignant or benign.

In other machine learning problems we may have more than one feature.

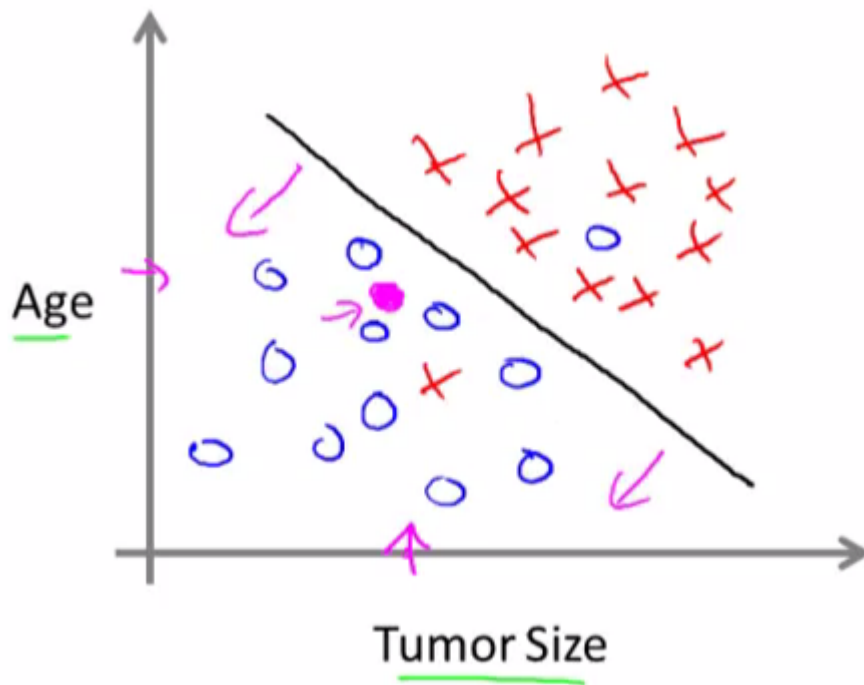
Here's an example. Let's say that instead of just knowing the tumor size, we know both the age of the patients and the tumor size. In that case maybe the data set will look like this.



So, let's say a person who tragically has a tumor. And maybe, their tumor size and age falls around there (rose point):



So given a data set like this, what the learning algorithm might do is throw a straight line through the data to try to separate out the malignant tumors from the benign ones. And with this, hopefully we can decide that the person's tumor falls on this benign side and is therefore more likely to be benign than malignant.



In this example we had **two features**, namely, the age of the patient and the size of the tumor. In other machine learning problems we will often have more features.

Most interesting learning algorithms is a learning algorithm that can deal with, not just two or three or five features, but an **infinite number of features**. So how do you deal with an infinite number of features. How do you even store an infinite number of things on the computer when your computer is gonna run out of memory.

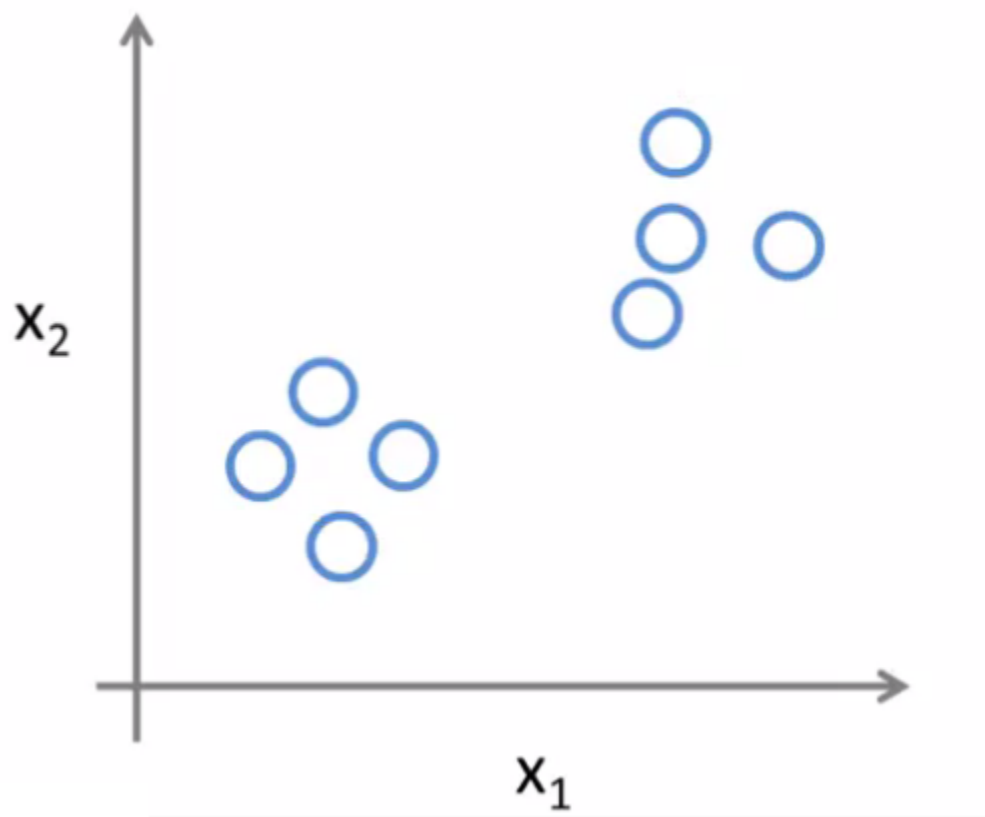
Unsupervised Learning

The second major type of machine learning problem is called Unsupervised Learning.

The difference between Unsupervised Learning and Supervised Learning is that in Supervised Learning we are told explicitly what is the so-called right answers (data are labeled).

In Unsupervised Learning, we're given data that doesn't have any labels or that all has the same label or really no labels. Like in this example:

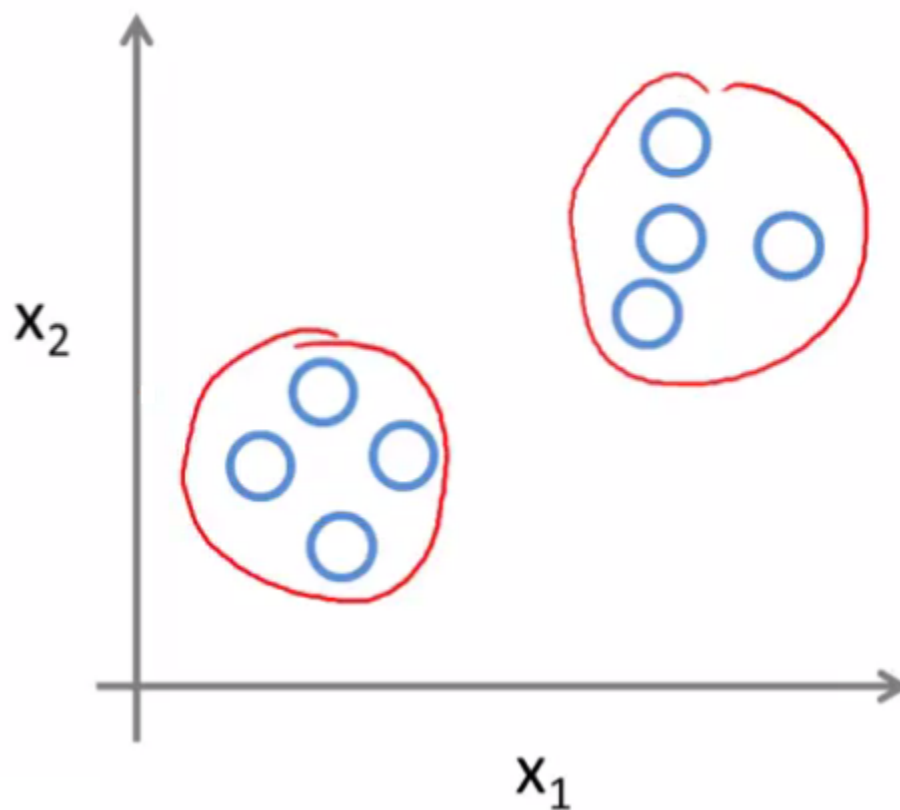
Unsupervised Learning



So we're given the data set and we're not told what to do with it and we're not told what each data point is. Instead we're just told, here is a data set. Can you find some structure in the data?

Given this data set, an Unsupervised Learning algorithm might decide that the data lives in two different clusters.

Unsupervised Learning



This is called a **clustering** algorithm.

Here are two examples where Unsupervised Learning or clustering is used.

Social network analysis:



So given knowledge about which friends you email the most or given your Facebook friends or your Google+ circles, can we automatically identify which are cohesive groups of friends, also which are groups of people that all know each other?

Market segmentation:



Market segmentation

Many companies have huge databases of customer information. So, can you look at this customer data set and automatically discover market segments and automatically group your customers into different market segments so that you can automatically and more efficiently sell or market your different market segments together?

This is Unsupervised Learning because we have all this customer data, but we don't know in advance what are the market segments and for the customers in our data set, we don't know in advance who is in market segment one, who is in market segment two, and so on. But we have to let the algorithm discover all this just from the data.

Part I

Supervised Learning

Chapter 1

Linear Regression

1.1 Notation

In general, we will let x_{ij} represent the value of the j th variable for the i th observation, where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, p$. We will use i to index the samples or observations (from 1 to n) and j will be used to index the variables (or features) (from 1 to p). We let \mathbf{X} denote a $n \times p$ matrix whose (i, j) th element is x_{ij} . That is,

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1p} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{np} \end{pmatrix}$$

Note that it is useful to visualize \mathbf{X} as a spreadsheet of numbers with n rows and p columns. We will write the rows of \mathbf{X} as x_1, x_2, \dots, x_n . Here x_i is a vector of length p , containing the p variable measurements for the i th observation. That is,

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}$$

(Vectors are by default represented as columns.)

We will write the columns of \mathbf{X} as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p$. Each is a vector of length n . That is,

$$\mathbf{x}_j = \begin{pmatrix} \mathbf{x}_{1j} \\ \mathbf{x}_{2j} \\ \vdots \\ \mathbf{x}_{nj} \end{pmatrix}$$

Using this notation, the matrix \mathbf{X} can be written as

$$\mathbf{X} = (\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_p)$$

or

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}$$

The T notation denotes the transpose of a matrix or vector.

We use y_i to denote the i th observation of the variable on which we wish to make predictions. We write the set of all n observations in vector form as

$$\mathbf{y} = \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{pmatrix}$$

Then the observed data consists of $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where each x_i is a vector of length p . (If $p = 1$, then x_i is simply a scalar).

1.2 Model Representation

Let's consider the example about predicting housing prices. We're going to use this data set as an example,

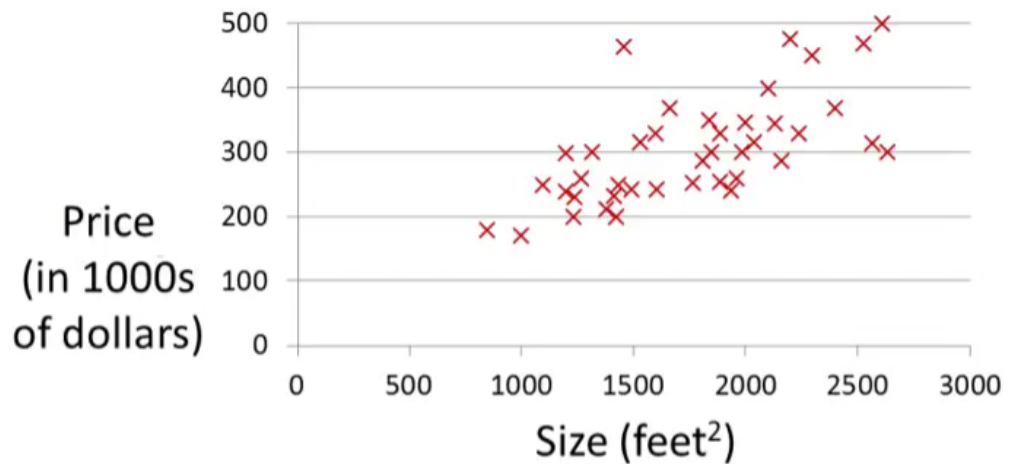


Figure 1.1:

Suppose that there is a person trying to sell a house of size 1250 square feet and he wants to know how much he might be able to sell the house for. One thing we could do is fit a model. Maybe fit a straight line to this data. Looks something like this,

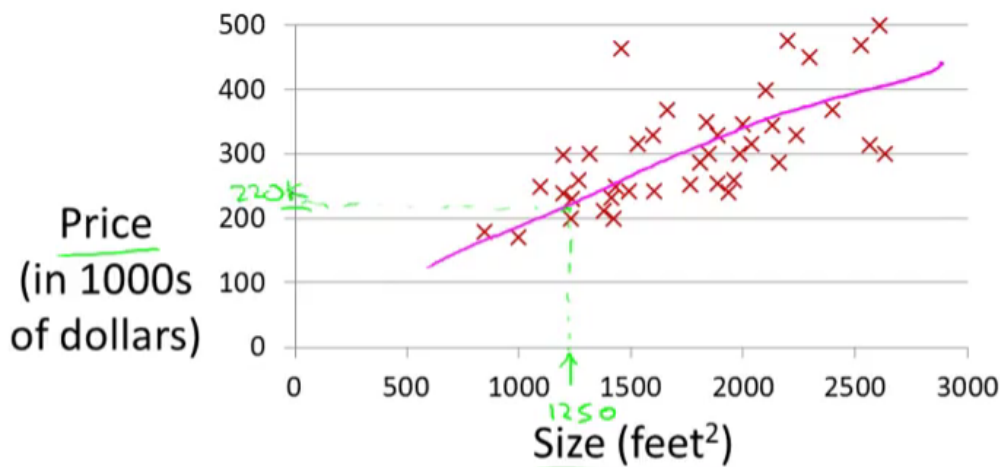


Figure 1.2:

and based on that, maybe he can sell the house for around \$220,000. Recall that this is an example of a supervised learning algorithm. And it's supervised learning because we're given the "right answer" for each of our examples. More precisely, this is an example of a regression problem where the term regression refers to the fact that we are predicting a real-valued output namely the price.

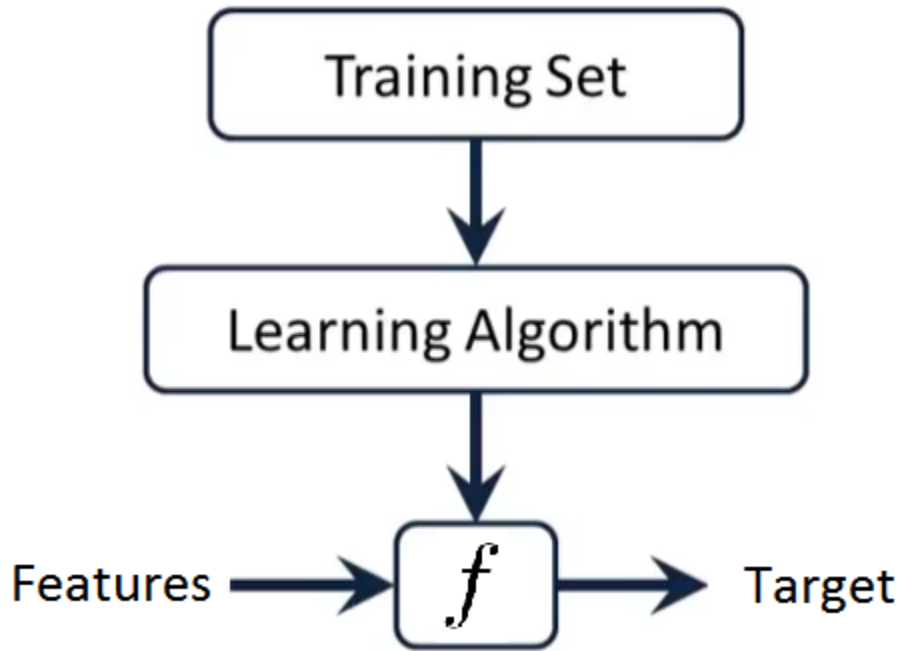
More formally, in supervised learning, we have a data set and this data set is called a **training set**. So for housing prices example, we have a training set of different housing prices and our job is to learn from this data how to predict prices of the houses.

Let's define some notation from this data set:

- The size of the house is the input variable.
- The house price is the output variable.
- The input variables are typically denoted using the variable symbol X ,
- The inputs go by different names, such as *predictors*, *independent variables*, *features*, *predictor* or sometimes just *variables*.
- The output variable is often called the *response*, *dependent variable* or *target*, and is typically denoted using the symbol Y .
- (x_i, y_i) is the i th training example.
- The set of $\{(x_i, y_i)\}$ is the training set.
- n is the number of training examples.

So here's how this supervised learning algorithm works. Suppose that we observe a quantitative response Y and p different predictors, X_1, X_2, \dots, X_p . We assume that there is some relationship between Y and $X = (X_1, X_2, \dots, X_p)$, which can be written in the very general form

$$Y = f(X) + \epsilon$$



Here f is some fixed but unknown function of X_1, X_2, \dots, X_p , and ϵ is a random error term, which is independent of X and has mean zero. The f function is also called *hypothesis* in Machine Learning. In general, the function f may involve more than one input variable. In essence, Supervised Learning refers to a set of approaches for estimating f .

1.3 Why Estimate f ?

There are two main reasons that we may wish to estimate f : *prediction* and *inference*.

Prediction

In many situations, a set of inputs X are readily available, but the output Y cannot be easily obtained. In this setting, since the error term averages to zero, we can predict Y using

$$\hat{Y} = \hat{f}(X)$$

where \hat{f} represents our estimate for f , and \hat{Y} represents the resulting prediction for Y . Like in the example above about predicting housing prices.

We can measure the accuracy of \hat{Y} by using a **cost function**. In the regression models, the most commonly-used measure is the *mean squared error* (MSE), given by

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$$

Inference

We are often interested in understanding the way that Y is affected as X_1, X_2, \dots, X_p change. In this situation we wish to estimate f , but our goal is not necessarily to make predictions for Y . We instead want to understand the relationship between X and Y , or more specifically, to understand how Y changes as a function of X_1, X_2, \dots, X_p . In this case, one may be interested in answering the following questions:

- Which predictors are associated with the response?
- What is the relationship between the response and each predictor?
- Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

1.4 Simple Linear Regression Model

Simple linear regression is a very straightforward approach for predicting a quantitative response Y on the basis of a single predictor variable X . It assumes that there is approximately a linear relationship between X and Y . Mathematically, we can write this linear relationship as

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

$$Y \approx \beta_0 + \beta_1 X$$

where β_0 and β_1 are two unknown constants that represent the *intercept* and *slope*, also known as **coefficients** or **parameters**, and ϵ is the error term.

Given some estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model coefficients, we predict future inputs x using

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

where \hat{y} indicates a prediction of Y on the basis of $X = x$. The *hat* symbol, $\hat{\cdot}$, denotes an estimated value.

1.5 Estimating the Coefficients

Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for Y based on the i th value of X . Then $e_i = y_i - \hat{y}_i$ represents the i th **residual**.

We define the **residual sum of squares (RSS)** as

$$\begin{aligned} RSS &= e_1^2 + e_2^2 + \dots + e_n^2 \\ &= \sum_{i=1}^n e_i^2 \end{aligned}$$

or equivalently as

$$\begin{aligned} RSS &= (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \dots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2 \\ &= \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \end{aligned}$$

The *least squares* approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the RSS. The minimizing values can be shown to be

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{s_{xy}}{s_x^2}$$

and

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where:

- $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ is the *sample mean*.
- $s_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$ is the *sample variance*. The sample standard deviation is $s_x = \sqrt{s_x^2}$.
- $s_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$ is the *sample covariance*. It measures the degree of linear association between x_1, \dots, x_n and y_1, \dots, y_n . Once scaled by $s_x s_y$, it gives the *sample correlation coefficient*, $r_{xy} = \frac{s_{xy}}{s_x s_y}$.



Click here to see the influence of the distance employed in the sum of squares. Try to minimize the sum of squares for the different datasets. The choices of intercept and slope that minimize the sum of squared distances for a kind of distance are not the optimal for a different kind of distance.

1.6 Assessing the Accuracy of the Coefficient Estimates

The standard error of an estimator reflects how it varies under repeated sampling. We have

$$\begin{aligned} \text{SE}(\hat{\beta}_1)^2 &= \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \\ \text{SE}(\hat{\beta}_0)^2 &= \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right] \end{aligned}$$

where $\sigma^2 = \text{Var}(\epsilon)$

In general, σ^2 is known, but can be estimated from the data. The estimate of σ is known as the *residual standard error*, and is given by

$$\text{RSE} = \sqrt{\frac{\text{RSS}}{(n-2)}}$$

These standard errors can be used to compute *confidence intervals*. A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter. It has the form

$$\hat{\beta}_1 \pm 2 \cdot \text{SE}(\hat{\beta}_1)$$

That is, there is approximately a 95% chance that the interval

$$\left[\hat{\beta}_1 - 2 \cdot \text{SE}(\hat{\beta}_1), \hat{\beta}_1 + 2 \cdot \text{SE}(\hat{\beta}_1) \right]$$

will contain the true value of β_1 . Similarly, a confidence interval for β_0 approximately takes the form

$$\hat{\beta}_0 \pm 2 \cdot \text{SE}(\hat{\beta}_0)$$

Hypothesis testing

Standard errors can also be used to perform *hypothesis tests* on the coefficients. The most common hypothesis test involves testing the *null hypothesis* of

$$H_0 : \text{There is no relationship between } X \text{ and } Y$$

versus the *alternative hypothesis*

$$H_1 : \text{There is some relationship between } X \text{ and } Y$$

Mathematically, this corresponds to testing

$$H_0 : \beta_1 = 0$$

versus

$$H_1 : \beta_1 \neq 0$$

since if $\beta_1 = 0$ then the simple linear regression model reduces to $Y = \beta_0 + \epsilon$, and X is not associated with Y .

To test the null hypothesis H_0 , we compute a ***t-statistic***, given by

$$t = \frac{\hat{\beta}_1 - 0}{\text{SE}(\hat{\beta}_1)}$$

This will have a *t*-distribution (*Student*) with $n - 2$ degrees of freedom, assuming $\beta_1 = 0$.

Using statistical software, it is easy to compute the probability of observing any value equal to $|t|$ or larger. We call this probability the ***p-value***.

If *p*-value is small enough (typically under 0.01 (1% error) or 0.05 (5% error)) we reject the null hypothesis, that is we declare a relationship to exist between X and Y .

1.7 ANOVA and model fit

1.7.1 ANOVA

In this section we will see how the variance of Y is decomposed into two parts, each one corresponding to the regression and to the error, respectively. This decomposition is called the *ANalysis Of VAriance* (ANOVA).

Before explaining ANOVA, it is important to recall an interesting result: *the mean of the fitted values $\hat{Y}_1, \dots, \hat{Y}_n$ is the mean of Y_1, \dots, Y_n* . This is easily seen if we plug-in the expression of $\hat{\beta}_0$:

$$\frac{1}{n} \sum_{i=1}^n \hat{Y}_i = \frac{1}{n} \sum_{i=1}^n (\hat{\beta}_0 + \hat{\beta}_1 X_i) = \hat{\beta}_0 + \hat{\beta}_1 \bar{X} = (\bar{Y} - \hat{\beta}_1 \bar{X}) + \hat{\beta}_1 \bar{X} = \bar{Y}.$$

The ANOVA decomposition considers the following measures of variation related with the response:

- $SST = \sum_{i=1}^n (Y_i - \bar{Y})^2$, the **total sum of squares**. This is the *total variation* of Y_1, \dots, Y_n , since $SST = ns_y^2$, where s_y^2 is the sample variance of Y_1, \dots, Y_n .
- $SSR = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$, the **regression sum of squares**¹. This is the variation explained by the regression line, that is, *the variation from \bar{Y} that is explained by the estimated conditional mean $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$* . $SSR = ns_{\hat{y}}^2$, where $s_{\hat{y}}^2$ is the sample variance of $\hat{Y}_1, \dots, \hat{Y}_n$.
- $SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$, the **sum of squared errors**². Is the variation around the conditional mean. Recall that $SSE = \sum_{i=1}^n \hat{\varepsilon}_i^2 = (n-2)\hat{\sigma}^2$, where $\hat{\sigma}^2$ is the sample variance of $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$.

The ANOVA decomposition is

$$\underbrace{SST}_{\text{Variation of } Y'_i s} = \underbrace{SSR}_{\text{Variation of } \hat{Y}'_i s} + \underbrace{SSE}_{\text{Variation of } \hat{\varepsilon}'_i s}$$

The graphical interpretation of this equation is shown in the following figures.

¹Recall that SSR is different from RSS (Residual Sum of Squares)

²Recall that SSE and RSS (for $(\hat{\beta}_0, \hat{\beta}_1)$) are just different names for referring to the same quantity: $SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^n (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2 = RSS(\hat{\beta}_0, \hat{\beta}_1)$.



Figure 1.3: Visualization of the ANOVA decomposition. SST measures the variation of Y_1, \dots, Y_n with respect to \bar{Y} . SSR measures the variation with respect to the conditional means, $\hat{\beta}_0 + \hat{\beta}_1 X_i$. SSE collects the variation of the residuals.



Click here to see the ANOVA decomposition and its dependence on σ^2 and $\hat{\sigma}^2$.

The ANOVA table summarizes the decomposition of the variance. Here is given in the layout employed by R.

	Degrees of freedom	Sum Squares	Mean Squares	F -value	p -value
Predictor	1	SSR	$\frac{SSR}{1}$	$\frac{SSR/1}{SSE/(n-2)}$	p
Residuals	$n - 2$	SSE	$\frac{SSE}{n-2}$		

The `anova` function in R takes a model as an input and returns the ANOVA table.

The “ F -value” of the ANOVA table represents the value of the F -statistic $\frac{SSR/1}{SSE/(n-2)}$. This statistic is employed to test

$$H_0 : \beta_1 = 0 \quad \text{vs.} \quad H_1 : \beta_1 \neq 0,$$

that is, the hypothesis of no linear dependence of Y on X . The result of this test is completely equivalent to the t -test for β_1 that we saw previously in the Hypothesis testing (this is something *specific for simple linear regression* – the F -test will not be equivalent to the t -test for β_1 in the Multiple Linear Regression).

It happens that

$$F = \frac{SSR/1}{SSE/(n-2)} \stackrel{H_0}{\sim} F_{1,n-2},$$

where $F_{1,n-2}$ is the *Snedecor's F distribution*³ with 1 and $n - 2$ degrees of freedom.

If H_0 is true, then F is expected to be *small* since SSR will be close to zero. The p -value of this test is the same as the p -value of the t -test for $H_0 : \beta_1 = 0$.

1.7.2 The R^2 Statistic

To calculate R^2 , we use the formula

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

where $TSS = \sum (y_i - \bar{y})^2$ is the *total sum of squared*.

R^2 measures the *proportion of variability in Y that can be explained using X* . An R^2 statistic that is close to 1 indicates that a large proportion of the variability in the response has been explained by the regression. A number near 0 indicates that the regression did not explain much of the variability in the response; this might occur because the linear model is wrong, or the inherent error σ^2 is high, or both.

It can be shown that in this simple linear regression setting that $R^2 = r^2$, where r is the correlation between X and Y :

³The $F_{n,m}$ distribution arises as the quotient of two independent random variables χ_n^2 and χ_m^2 , $\frac{\chi_n^2/n}{\chi_m^2/m}$.

$$r = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$



R^2 does not measure the correctness of a linear model but its **usefulness** (for prediction, for *explaining the variance* of Y), assuming the model is correct.

Trusting blindly the R^2 can lead to catastrophic conclusions, since the model may not be correct.

So remember:



A large R^2 means *nothing* if the **assumptions of the model do not hold**. R^2 is the proportion of variance of Y explained by X , but, of course, *only when the linear model is correct*.