Linear regression is a **supervised learining** algorithm used when target / dependent variable **continues** real number. It establishes relationship between dependent variable y and one or more independent variable x using best fit line. It work on the principle of ordinary least square (OLS) / Mean square errror (MSE). In statistics ols is method to estimated unkown parameter of linear regression function, it's goal is to minimize sum of square difference between observed dependent variable in the given data set and those predicted by linear regression fuction.

Hypothesis representation

We will use $\mathbf{x_i}$ to denote the independent variable and $\mathbf{y_i}$ to denote dependent variable. A pair of $\mathbf{(x_i, y_i)}$ is called training example. The subscripe $\mathbf{i}$ in the notation is simply index into the training set. We have $\mathbf{m}$ training example then $\mathbf{i = 1, 2, 3, \ldots m}$.

The goal of supervised learning is to learn a *hypothesis function* $\mathbf{h}$, for a given training set that can used to estimate $\mathbf{y}$ based on $\mathbf{x}$. So hypothesis fuction represented as

$$\mathbf{h_\theta(x_i) = \theta_0 + \theta_1 x_i}$$

$\mathbf{\theta_0, \theta_1}$ are parameter of hypothesis. This is equation for **Simple / Univariate Linear regression**.

For **Multiple Linear regression** more than one independent variable exit then we will use $\mathbf{x_{ij}}$ to denote indepedent variable and $\mathbf{y_i}$ to denote dependent variable. We have $\mathbf{n}$ independent variable then $\mathbf{j = 1, 2, 3. \ldots . n}$. The hypothesis function represented as

$$\mathbf{h_\theta(x_i) = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \ldots . \theta_j x_{ij} \ldots \ldots \theta_n x_{mn}}$$

$\mathbf{\theta_0, \theta_1, \ldots . \theta_j \ldots . \theta_n}$ are parameter of hypothesis, $\mathbf{m}$ Number of training exaples, $\mathbf{n}$ Number of independent variable, $\mathbf{x_{ij}}$ is $\mathbf{i^{th}}$ training exaple of $\mathbf{j^{th}}$ feature.

In [1]:

```
import pandas  as pd #Data manipulation
# Import dataset
path = './'
df = pd.read_csv(path+'insurance.csv')
print('\nNumber of rows and columns in the data set: ',df.shape)
print('')

#Lets look into top few rows and columns in the dataset
df.head()
```

```
Number of rows and columns in the data set:  (1338, 7)
```

Out[1]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

When we look at the shape of dataset it has return as (1338,7).So there are $\mathbf{m = 1338}$ training exaple and $\mathbf{n = 7}$ independent variable. The target variable here is charges and remaining six variables such as age, sex, bmi, children, smoker, region are independent variable. There are multiple independent variable, so we need to fit Multiple linear regression. Then the hypothesis function looks like

$$\mathbf{h_\theta(x_i) = \theta_0 + \theta_1 age + \theta_2 sex + \theta_3 bmi + \theta_4 children + \theta_5 smoker + \theta_6 region}$$

This multiple linear regression equation for given dataset.
If $\mathbf{i = 1}$ then

$$\mathbf{h_\theta(x_1) = \theta_0 + \theta_1 19 + \theta_2 female + \theta_3 27.900 + \theta_4 1 + \theta_5 yes + \theta_6 southwest}$$

$$\mathbf{y_1 = 16884.92400}$$

If $\mathbf{i = 3}$ then

$$\mathbf{h_\theta(x_3) = \theta_0 + \theta_1 28 + \theta_2 male + \theta_3 33.000 + \theta_4 3 + \theta_5 no + \theta_6 northwest}$$

$$\mathbf{y_3 = 4449.46200}$$

Coefficients $\boldsymbol{\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6}$ are the parameters that will be the subject of the model computation, i.e. that set of values that minimize the prediction error.

Each coefficient $\boldsymbol{\theta_0}$ affects to each value in any row of the dataset. Coefficients are represented as a column vector, and each row can be represented as a row vector. let's see the example of the first row:

$$\mathbf{x_1} = (x_{11} \quad x_{12} \quad x_{13} \quad x_{14} \quad x_{15} \quad x_{16}) = (19 \quad female \quad 27.900 \quad 1 \quad no \quad northw$$

Hence the charge for the first row (see expression for $\mathbf{i = 1}$ above) can be expressed in matrix formulation as the dot product of the row vector (values) by the column vector (coefficients).

Matrix Formulation

Generalizing, we can write the row vector as $\mathbf{x_{ij}} = \left(\mathbf{x_{i1}} \ \mathbf{x_{i2}} \ \ldots \ \mathbf{x_{in}} \right)$, where **i** is the number of the row.

Now we combine all row vectors into single input matrix of size (m, n) and denoted it by $\mathbf{X}$ input matrix, which consist of all training exaples,

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & . & . & . & . & x_{1n} \\ x_{21} & x_{22} & . & . & . & . & x_{2n} \\ x_{31} & x_{32} & . & . & . & . & x_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & . & . & . & . & .x_{mn} \end{pmatrix}_{(m,n)}$$

We represent parameter of function and dependent variable in vactor form as

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ . \\ . \\ \theta_j \\ . \\ . \\ \theta_n \end{pmatrix}_{(n+1,1)} \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ y_i \\ . \\ . \\ y_m \end{pmatrix}_{(m,1)}$$

So we represent hypothesis function in vectorize form

$$\mathbf{y = h_\theta(x) = X\theta}$$

.

In above plot we fit regression line into the variables.

Cost function

A cost function measures how much error in the model is in terms of ability to estimate the relationship between x and y. We can measure the accuracy of our hypothesis function by using a cost function. This takes an average difference of observed dependent variable in the given the dataset and those predicted by the hypothesis function.

$$\mathbf{J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2}$$

$$\mathbf{J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2}$$

To implement the linear regression, take training example add an extra column that is x_0 feature, where $\mathbf{x_0 = 1}$. $\mathbf{x_o} = \begin{pmatrix} x_{i0} & x_{i1} & x_{i2} & . & . & . & x_{mi} \end{pmatrix}$,where $\mathbf{x_{i0} = 0}$ and input matrix will become as

$$\mathbf{X} = \begin{pmatrix} x_{10} & x_{11} & x_{12} & . & . & . & . & x_{1n} \\ x_{20} & x_{21} & x_{22} & . & . & . & . & x_{2n} \\ x_{30} & x_{31} & x_{32} & . & . & . & . & x_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m0} & x_{m1} & x_{m2} & . & . & . & . & .x_{mn} \end{pmatrix}_{(m,n+1)}$$

Each of the m input samples is similarly a column vector with n+1 rows x_0 being 1 for our convenience, that is $\mathbf{x_{10}, x_{20}, x_{30} . . . . x_{m0} = 1}$. Now we rewrite the ordinary least square cost function in matrix form as

$$\mathbf{J(\theta) = \frac{1}{m}(X\theta - y)^T(X\theta - y)}$$

Let's look at the matrix multiplication concept,the multiplication of two matrix happens only if number of column of firt matrix is equal to number of row of second matrix. Here input matrix $\mathbf{X}$ of size $\mathbf{(m, n+1)}$, parameter of function is of size $(n+1, 1)$ and dependent variable vector of size $\mathbf{(m, 1)}$. The product of matrix $\mathbf{X_{(m,n+1)}}\theta_{\mathbf{(n+1,1)}}$ will return a vector of size $\mathbf{(m, 1)}$, then product of $\mathbf{(X\theta - y)^T_{(1,m)})(X\theta - y)_{(m,1)}}$ will return size of unit vector.

Normal Equation

The normal equation is an analytical solution to the linear regression problem with a ordinary least square cost function. To minimize our cost function, take partial derivative of $\mathbf{J(\theta)}$ with respect to θ and equate to 0. The derivative of function is nothing but if a small change in input what would be the change in output of function.

$$\mathbf{min_{\theta_0,\theta_1..\theta_n} J(\theta_0, \theta_1..\theta_n)}$$

$$\mathbf{\frac{\partial J(\theta_j)}{\partial \theta_j} = 0}$$

where $\mathbf{j = 0, 1, 2, \ldots. n}$

Now we will apply partial derivative of our cost function,

$$\mathbf{\frac{\partial J(\theta_j)}{\partial \theta_j} = \frac{\partial}{\partial \theta}\frac{1}{m}(X\theta - y)^T(X\theta - y)}$$

I will throw $\mathbf{\frac{1}{m}}$ part away since we are going to compare a derivative to 0. And solve $\mathbf{J(\theta)}$
,

$$\mathbf{J(\theta) = (X\theta - y)^T(X\theta - y)}$$
$$\mathbf{= (X\theta)^T - y^T)(X\theta - y)}$$

$$\mathbf{= (\theta^T X^T - y^T)(X\theta - y)}$$
$$\mathbf{= \theta^T X^T X\theta - y^T X\theta - \theta^T X^T y + y^T y}$$
$$\mathbf{= \theta^T X^T X\theta - 2\theta^T X^T y + y^T y}$$

Here $\mathbf{y^T_{(1,m)} X_{(m,n+1)}\theta_{(n+1,1)} = \theta^T_{(1,n+1)} X^T_{(n+1,m)} y_{(m,1)}}$ because unit vector.

$$\mathbf{\frac{\partial J(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}(\theta^T X^T X\theta - 2\theta^T X^T y + y^T y)}$$

$$\mathbf{= X^T X\frac{\partial \theta^T \theta}{\partial \theta} - 2X^T y\frac{\partial \theta^T}{\partial \theta} + \frac{\partial y^T y}{\partial \theta}}$$

Partial derivative $\mathbf{\frac{\partial x^2}{\partial x} = 2x}$, $\mathbf{\frac{\partial kx^2}{\partial x} = kx}$, $\mathbf{\frac{\partial Constact}{\partial x} = 0}$

$$\frac{\partial \mathbf{J}(\theta)}{\partial \theta} = \mathbf{X^T X 2\theta - 2X^T y + 0}$$

$$\mathbf{0 = 2X^T X\theta - 2X^T y}$$

$$\mathbf{X^T X\theta = X^T}$$

$$\theta = \mathbf{(X^T X)^{-1} X^T y}$$

this the normal equation for linear regression

Model evaluation

We will predict value for target variable by using our model parameter for test data set. Then compare the predicted value with actual valu in test set. We compute **Mean Square Error** using formula

$$\mathbf{J}(\theta) = \mathbf{\frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2}$$

$\mathbf{R^2}$ is statistical measure of how close data are to the fitted regression line. $\mathbf{R^2}$ is always between 0 to 100%. 0% indicated that model explains none of the variability of the response data around it's mean. 100% indicated that model explains all the variablity of the response data around the mean.

$$\mathbf{R^2 = 1 - \frac{SSE}{SST}}$$

SSE = Sum of Square Error
SST = Sum of Square Total

$$\mathbf{SSE = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2}$$

$$\mathbf{SST = \sum_{i=1}^{m} (y_i - \bar{y}_i)^2}$$

Here $\mathbf{\hat{y}}$ is predicted value and $\mathbf{\bar{y}}$ is mean value of $\mathbf{y}$.

Model Validation

In order to validated model we need to check few assumption of linear regression model. The common assumption for *Linear Regression* model are following

1. Linear Relationship: In linear regression the relationship between the dependent and independent variable to be *linear*. This can be checked by scatter ploting Actual value Vs Predicted value
2. The residual error plot should be *normally* distributed.
3. The *mean* of *residual error* should be 0 or close to 0 as much as possible
4. The linear regression require all variables to be multivariate normal.

5. Linear regession assumes that there is little or no *Multicollinearity in the data*. *Multicollinearity occurs when the independent variables are too highly correlated with each other. The variance inflation factor* VIF* identifies correlation between independent variables and strength of that correlation. $\mathbf{VIF} = \frac{\mathbf{1}}{\mathbf{1-R^2}}$, If VIF >1 & VIF <5 moderate correlation, VIF < 5 critical level of multicollinearity.
6. Homoscedasticity: The data are homoscedastic meaning the residuals are equal across the regression line. We can look at residual Vs fitted value scatter plot. If heteroscedastic plot would exhibit a funnel shape pattern.