Polynomial Regression

A special case of : Basis Functions.

As discussed in very great detail Simple Linear Regression maps the relationship between Y and X as linear :

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

In a non-linear settings the linear regression can be extended by **Polynomial function** :

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 \cdots + \beta_p x_i^p$$

or

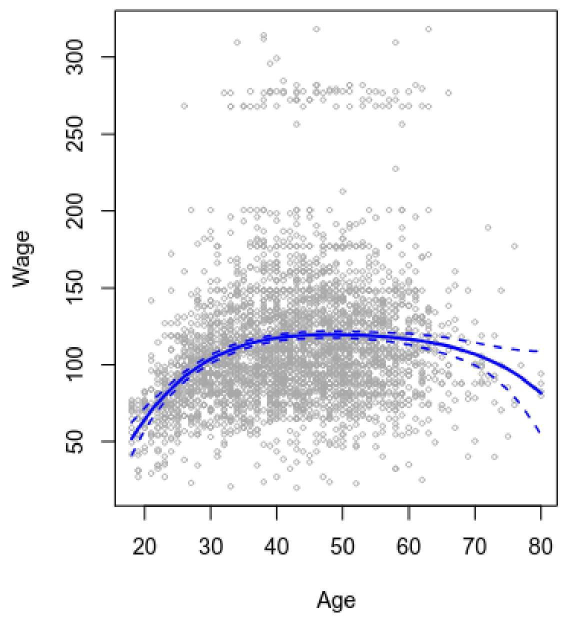
$$Y = X\beta + \epsilon$$

With :

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^p \end{bmatrix}$$

- p is the number of features or dimensions in out data matrix $\mathbf{X}$

Even tho it's used in non-linear settings **Polynomial Regression** is still linear regression since the coefficients β_j are still of a degree one, So the Ordinary Least Squares still works



- Before using polynomial regression it's important to **standardize** the features since all the **polynomials** are on different scales
- When the relationships are expected to be **polynomial**
- quick exploratory analysis and getting to know the dataset more
- **Polynomial** can be applied to any linear model we discuss so far Logistic Regression, Multiple Linear Regression,

Polynomial Regression Limits

Polynomial Regression isn't used in modern machine learning due to :

- Poor interpretation
- **High Degrees** without scaled features will blow up the gradient calculations
- Overfitting with High Degrees
- Numerically unstable

- Bad for extrapolation the **tails** tend to wiggle around the end of the data and beyond it
  Alternative is to use **kearnelized** models such as **SVR** with **polynomial kernel** or Generalized Additive Models.

Note : **Polynomial Regression** follows the same model selection methods discuss in previous chapters(Cross-Validation, Choosing the Optimal Model)