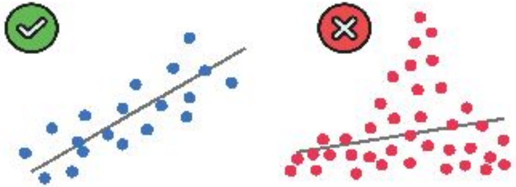
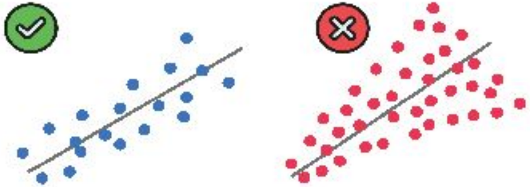
House Price= a + b×**Size** + c ×**No. Of room**

Assumptions of Linear Regression

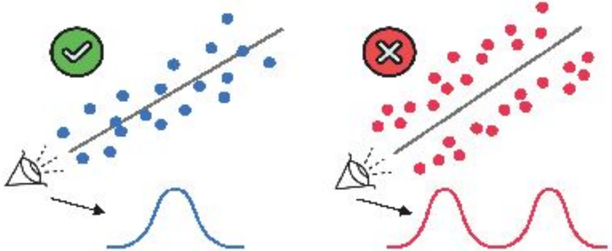
1. Linearity
(Linear relationship between Y and each X)



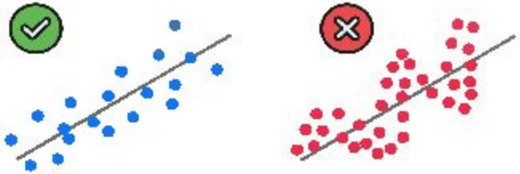
2. Homoscedasticity
(Equal variance)



3. Multivariate Normality
(Normality of error distribution)



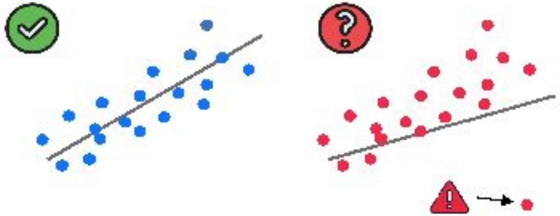
4. Independence
(of observations. Includes "no autocorrelation")



5. Lack of Multicollinearity
(Predictors are not correlated with each other)

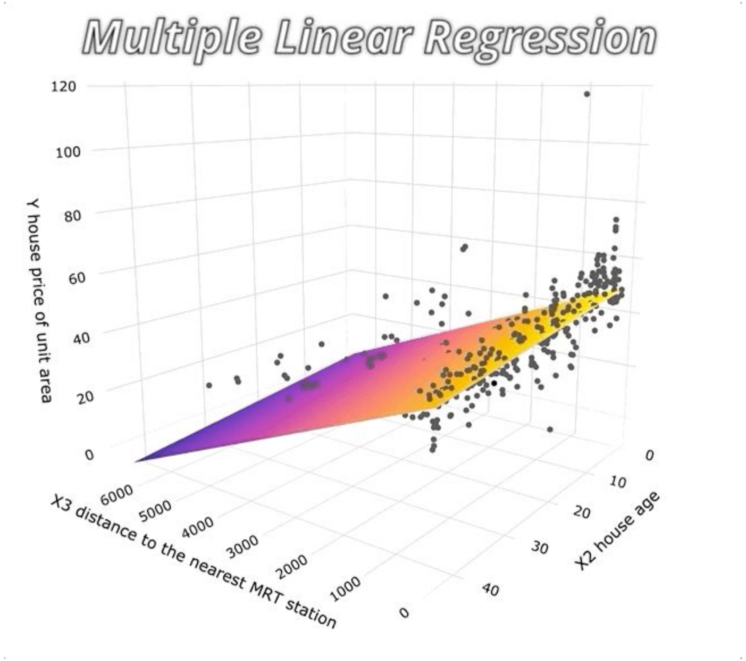
$X_1 \nsim X_2$ $\quad$ $X_1 \sim X_2$

6. The Outlier Check
(This is not an assumption, but an "extra")



Multiple Linear Regression:

- It's a form of linear regression that is used when there are two or more predictors.
- To predict the outcome from multiple input variables.
- The same as simple linear regression optimization is done using OLS or GD.
- E.g.:
  - In simple linear regression model predicts house price from number of rooms.
  - In multiple linear regression model predict price based on size, locations, number of rooms (higher dimension) etc.



Multiple Linear Regression (Equation)

Simple Linear Regression

$$y = b_0 + b_1{*}x_1$$

Multiple Linear Regression

Dependent variable (DV)

Independent variables (IVs)

$$y = b_0 + b_1{*}x_1 + b_2{*}x_2 + \dots + b_n{*}x_n$$

Constant Or Bias

Coefficients Or Weights

Cost Function: Sum of Squared Errors

- **Objective**: Minimize the cost function to find the best fit line.
- **Cost Function Equation**:
  $J(\beta) = \sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2$
  - m: Number of samples
  - Y_i: Actual value
  - $\hat{Y}_i$: Predicted value

Estimating Parameters Using Least Squares

- Normal Equation:
  $$\beta = (X^T X)^{-1} X^T Y$$
  - X: Input matrix (including bias term) >>> feature col (area , num_bedroom)
  - Y: Output vector >>> price col
  - β: Coefficient vector

Price = area*a1 + num_bedroom*a2 + b

>>> [a1 , a2 , b]

Example: Predicting House Prices

Area (sqm)
100
150
200
250

Solving for β

$$\beta = (X^T X)^{-1} X^T Y$$

Using computation:

$$\beta = \begin{pmatrix} 50000 \\ 1000 \\ 20000 \end{pmatrix}$$

[
b(b
a1 (
a2 (
]

Data Matrices

$$X = \begin{pmatrix} 1 & 100 & 2 \\ 1 & 150 & 3 \\ 1 & 200 & 4 \\ 1 & 250 & 4 \end{pmatrix}$$

$$\begin{pmatrix} \vdots \\ 350000 \end{pmatrix}$$

$$X^T Y = \begin{pmatrix} \vdots \\ 146000000 \\ 2525000 \end{pmatrix}$$

Final Model

Multiple Linear Regression Equation

$$\text{Price} = 50000 + 1000 \times \text{Area} + 20000 \times \text{Bedrooms}$$

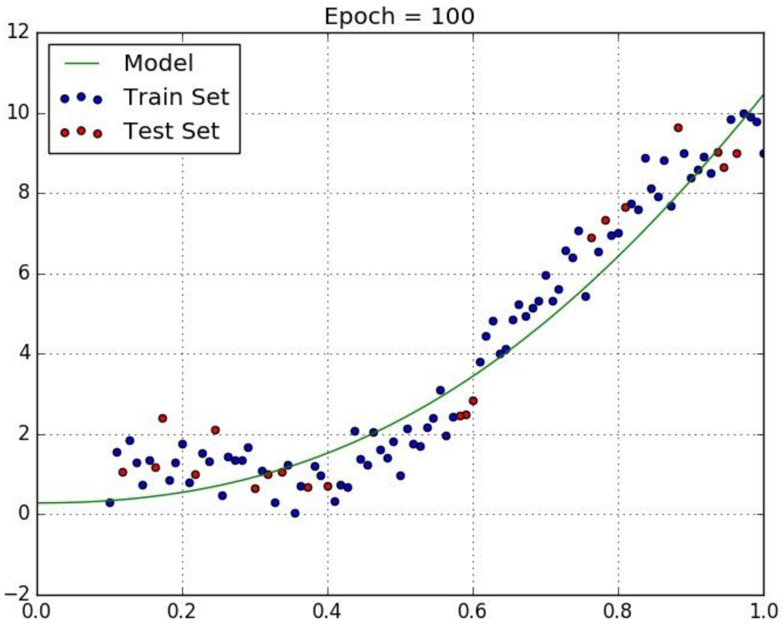
- **Intercept** (β_0): Base price = $50,000
- **Area** (β_1): Each additional square meter adds $1,000 to the price.
- **Bedrooms** (β_2): Each additional bedroom adds $20,000 to the price.

Polynomial Regression:

- Polynomial regression is a special case of linear regression.
- We fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables.

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n$$

- We call it linear because we don't look at it from the point of view of the x-variable. We look at coefficients.
- The larger the degree of polynomial (n) the more complex the model.



Note: $\theta 0$ is the bias, $\theta 1$, $\theta 2$, …, θn are the weights in the equation of the polynomial regression, and n is the degree of the polynomial.

Living Area (sq ft)
1200
1500
1800
2100

Living Area (sq ft)	Living Area^2 (sq ft^2)
1200	1,440,000
1500	2,250,000
1800	3,240,000
2100	4,410,000

Original Equation (Simple Linear Regression):

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Equation After Polynomial Expansion (Degree 2):

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

We found the following coefficients:

$\beta_0 = 100{,}000$

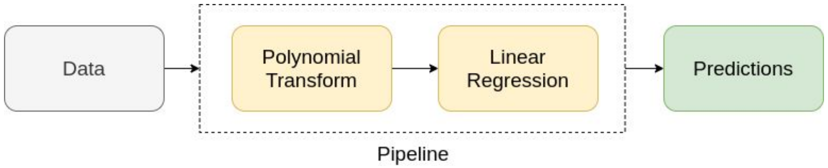
$\beta_1 = 150$

$\beta_2 = 0.02$

$$y = 100{,}000 + 150x + 0.02x^2$$

Polynomial Regression code:

- The implementation of polynomial regression is a two-step process.
  - First, we transform our data into a polynomial using the PolynomialFeatures API from sklearn.
  - Then use linear regression to fit the parameters.




```python
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

# make polynomial features object with degree of 2
poly = PolynomialFeatures(degree=2)
# Creating the new features data
poly_features = poly.fit_transform(x)
# Creating the polynomial regression model
poly_reg_model = LinearRegression()
# fit our data
poly_reg_model.fit(poly_features, y)
# predict labels
poly_reg_model.predict(poly_features)
# Calculate R2
poly_reg_model.score(poly_features, y)
# get model parameters
poly_reg_model.coef_
poly_reg_model.intercept_
```