Regression Analysis:

Definition

* **Regression** is a technique to determine the **relationship** between two or more **variables**.
* You can apply regression to scenarios that require **prediction** or **causal inference**.
* You can use regression to understand the **extent** to which the **area of a house** affects the **housing prices**.
* **Regress** means **predicting** **one variable from another**.

##### What can Regression Show ?

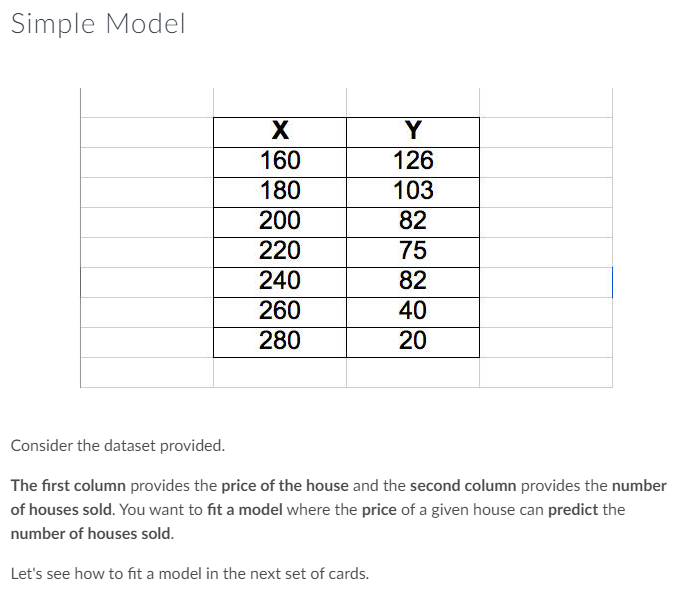
* **Regression** can show how **one variable** varies with respect to **another** variable.
* For example, the **price** of a **wine bottle** can vary **depending** on the **average growing season temperature**.

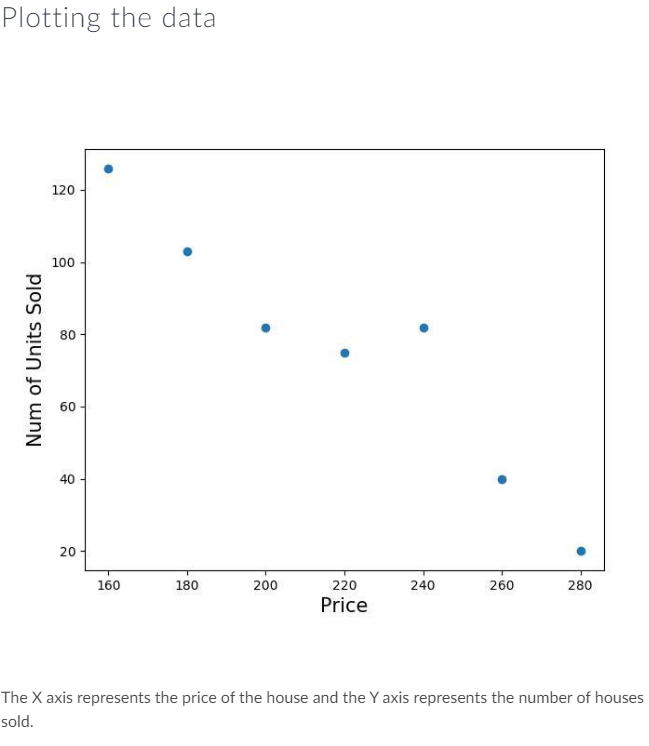
##### What Regression cannot show ?

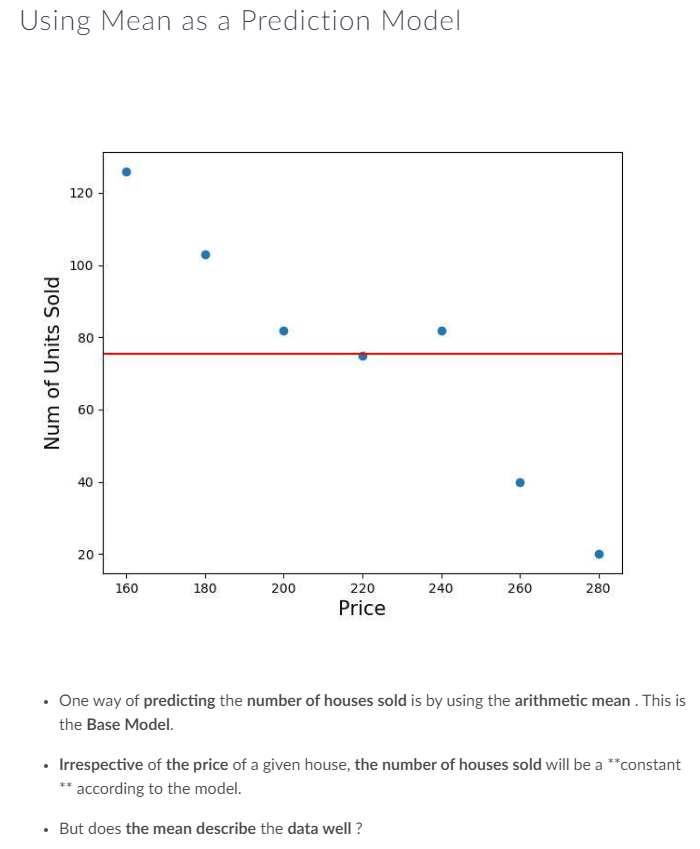
* **Regression cannot show** any **causal relationship** between **two variables**.
* For example, if the **area of the house is an independent variable** and the **price of the house is a dependent variable**, you cannot conclude that **houses with larger areas** will **increase the price of the house**.

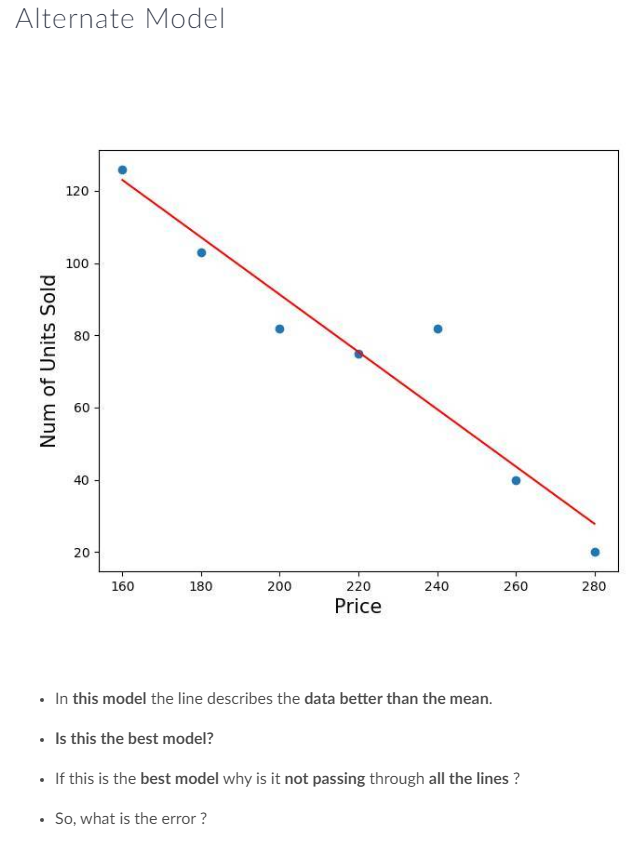
Topic Summary

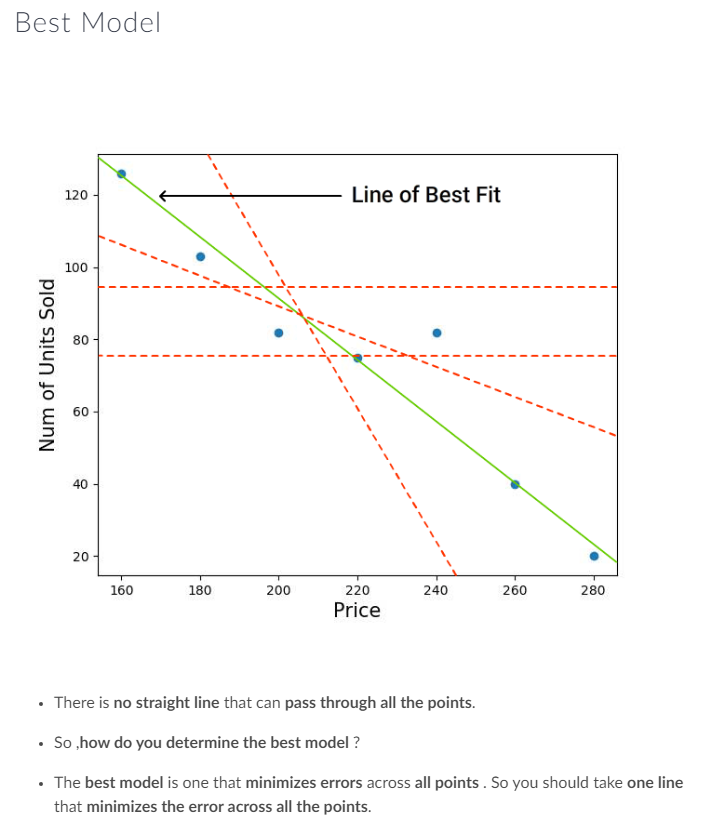
* **Correlation** is a **measure** that describes the **strength of relationship** between two variables.
* **Regression** explains in more detail about this **strength**



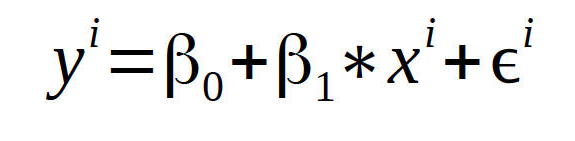




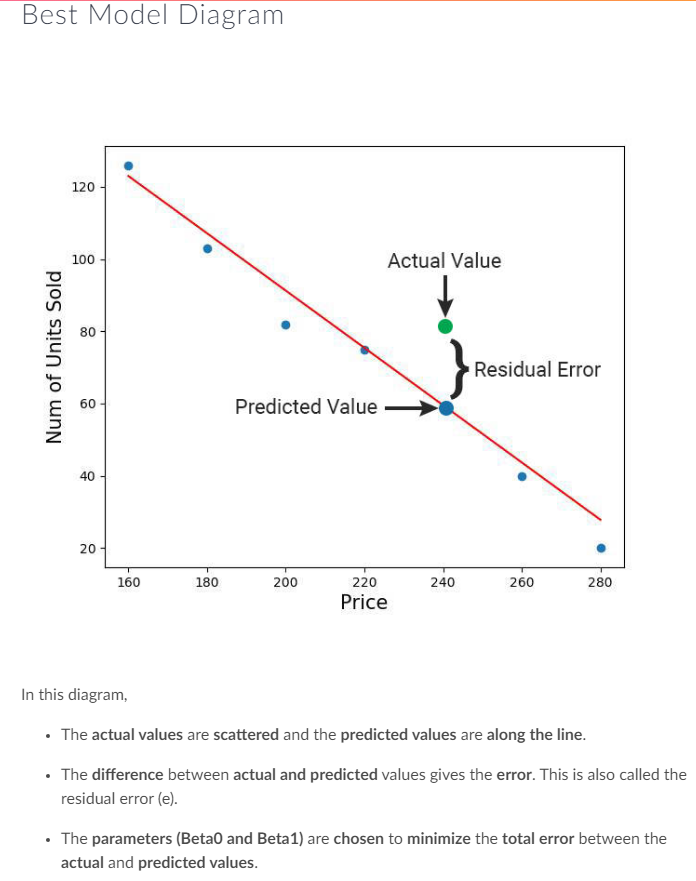




Model Representation

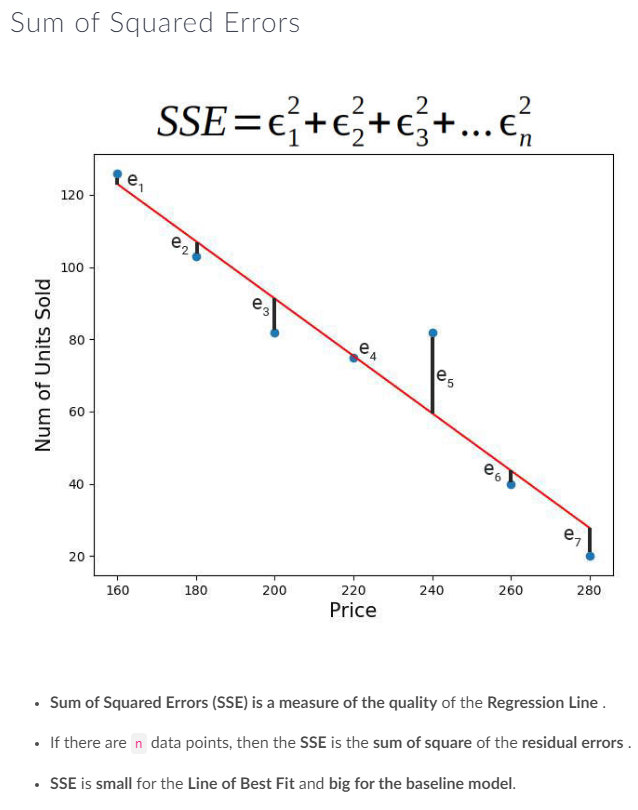


* y - Dependent variable
* x - Independent variable
* e - Error measure
* B0 and B1 Parameters that best fit the model



##### Measure of Quality

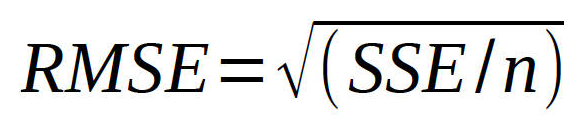
* You have seen how to **fit a model** that **best describes the data**. However, you can **never** get a **perfect fit**.
* How will you **measure** the **error/deviation** in a **model** that is **fit to the data** ?



##### Best Fit Line

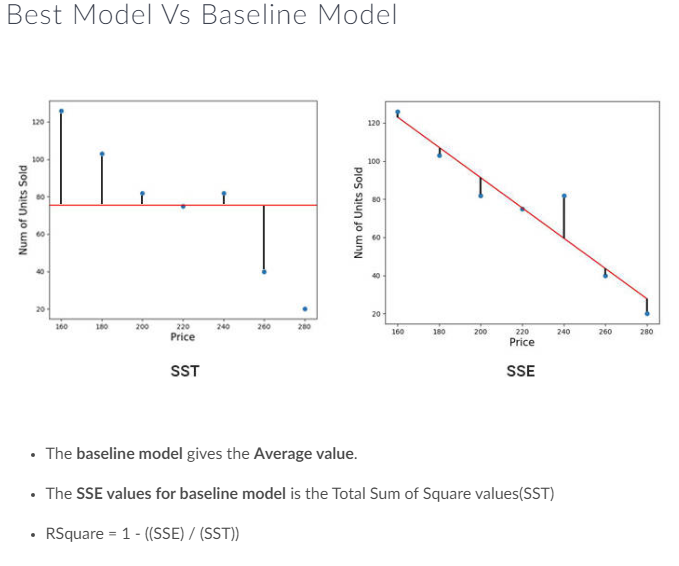
* The line with the **minimum SSE** is the **Regression Line**. **SSE** is sometimes **difficult to interpret because,**
* It depends on the **number of values (n)**
* The **units** are **hard to comprehend**
* So, is there a better way to gauge the quality of the **Regression Model ?**

##### RMSE



At times, the **SSE** is **difficult to interpret** and the **units** are **difficult to comprehend**. So, the **alternative measure of quality** is the **Root Mean Square Error (RMSE)**.

RMSE shrinks the magnitude of error by taking the square root of SSE divided by the number of observations (n).



##### R Square(R Sq) Properties

* **SSE and SST** values should be **greater than zero**.
* **R Sq** lies between **0 and 1**.
* **R Sq** is a **unit less** quantity.
* **R Sq = 0** means the **model** is just **as good as** the **base line** and there is **no improvement** from the **baseline model**.
* **R Sq = 1** means it is a **perfect model**. Ideally, you should strive towards getting the **R Sq close to 1** . But some models with **R Sq = 0** are also accepted depending on the scenario.

##### Model Interpretation

This is the equation for line of best fit

* y = 249.85714 - 0.7928571x
* For a unit change in X there is a .793 decrease in Y
* For a unit increase in price of the house, .793 lesser houses are sold .
* B0 is 249.85714
* B1 is -0.7928571

Practice:

Playground Setup

* To setup the playground and try the code, install the following: -Go the terminal and enter the following command \*\*pip install --user statsmodels \*\*.
* If your installation is successful you should get the following message:

Installing collected packages: patsy, statsmodels

Successfully installed patsy-0.4.1 statsmodels-0.8.0

##### Python Package for Linear Regression

* You will be learn the statsmodels package in Python.
* statsmodels is a complement to scipy package and provides the following :
  + **Descriptive Statistics**
  + **Estimation and Inference** for **statistical Models**

##### Data Frame Creation

Let us now setup the initial data for our regression analysis. You will need to load Price and No of house units sold into the data frame.

import pandas as pd

price = [160,180,200,220,240,260,280]

sale = [126,103,82,75,82,40,20]

priceDF = pd.DataFrame(price, columns=list('x'))

saleDF = pd.DataFrame(sale, columns=list('y'))

houseDf = pd.concat((priceDF, saleDF),axis=1)

print(houseDf)

print(priceDF)

##### Statsmodels Usage

Let us now see how to fit the data and get the regression outputs in Python.

Statsmodel can take input similar to R (Pass the variables with the dataframe) or take input as arrays.

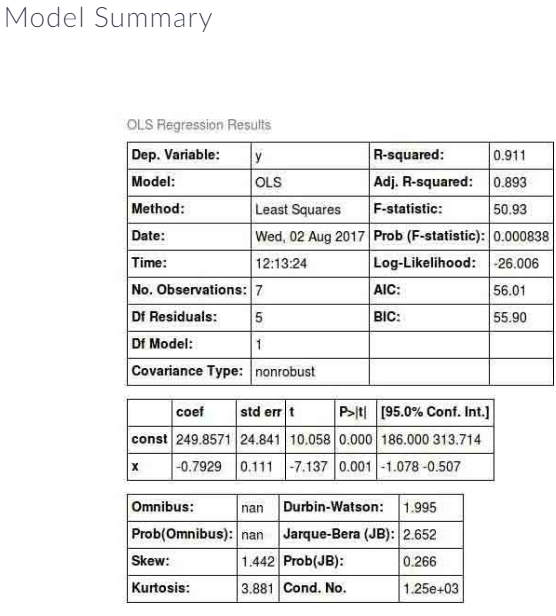
####Input as dataframe:

import statsmodels.api as sm

import statsmodels.formula.api as smf

smfModel = smf.ols('y~x',data=houseDf).fit()

print(smfModel.summary())



##### Understanding the Output

* **Dep. Variable**: The Dependent Variable
* **Model**: Algorithm used. Here, it is Ordinary Least Squares
* **Method**: Parameter Fitting method. Here, it is Least Squares
* **No. Observations:** Number of rows used for model fitting.
* **DF Residuals:** The degrees of freedom of the residuals (Difference between the number of observations and parameters).
* **DF Model:** The degrees of freedom of the model (The number of parameters estimated in the model excluding the constant term) .
* **R-squared:** Measure that says how well the model has performed with respect to the baseline model.

##### Data Prep

Now that you understand how to perform regression analysis using statsmodel, it's time execute the data set created using the following code:

from sklearn.datasets import fetch\_california\_housing

from sklearn.datasets import load\_boston

import pandas as pd

boston = load\_boston()

california = fetch\_california\_housing()

dataset = pd.DataFrame(boston.data, columns=boston.feature\_names)

dataset['target'] = boston.target

print(dataset.head())