# Introduction

The previous chapter was based on Python programming and Data Science. Data Science is a multidisciplinary field of study. The traditional field of statistics, computer science, and a subfield of artificial intelligence, known as Machine Learning is extended and utilized by Data Science. Data Science makes use of these models:

* Unstructured Data
* Structured Data.

**Structured data** involves any set of data that can be conveniently arranged into a table containing rows and columns. This type of data is normally stored in database management systems. However, **Unstructured data** cannot be normally stored in tabular form. A simple text document is an example of such a dataset.

The need to generate models at a large scale to make predictions for decision-making cuts across many industries. Data science, therefore, is being used in many industries, including healthcare, manufacturing industries, the finance sectors, e-commerce, education, and even government.

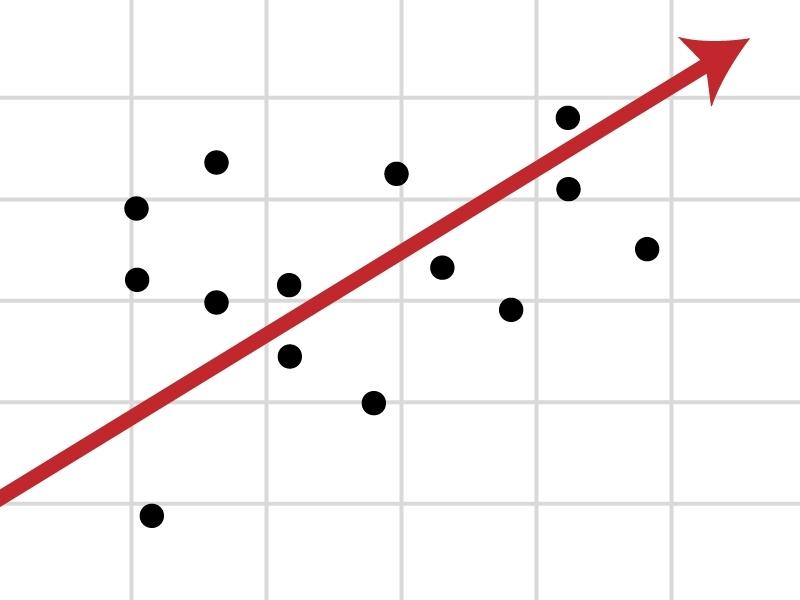
Regression is one of the key methods, that is used regularly in data science to model relationships between the variables. Target Variable, the value of which you are looking for, is a real number. Consider a case where a real estate business wants to understand and model the relationship between the prices of properties in a city and knowing the key attributes of the properties. Such a problem can be regarded as a Data Science problem and can be solved using regression.

In this example, the target variable of interest, the price of a property is a real number. Some factors of a property that can be used to predict its value are:

The area of land the property covers.

* The age of the property.
* Whether the property has a pool or not.
* The distance of the property from facilities such as hospitals, and schools.
* The number of bedrooms in a property.

To study above mentioned scenario briefly, **Regression analysis** can be used extensively. In this process, you have to create a function that maps the above-mentioned key attribute of a property to the target variable, which in this case, is the price of the property.



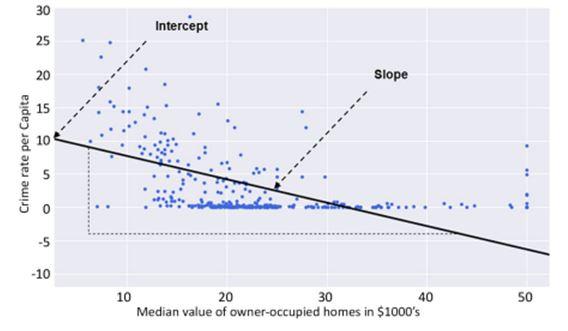
Regression analysis is part of a family of machine learning techniques called **Supervised machine learning**. It is called supervised because the machine learning algorithm that learns the model is provided a kind of question and answer dataset to learn from. In this scenario, “***question***” is the key attribute, and the “***answer***” is the property price for each property that is used in the study. Once the model has been learned by the algorithm, the model can be provided with a question for it to answer.

This chapter contains a brief introduction to Linear regression. Python provides a rich set of modules that can be utilized to conduct rigorous regression analyses of different types. In this chapter, some of these modules like pandas, statsmodels, seaborn, matplotlib, and skicit-learn will be used.

# What is Simple Linear Regression

Regression analysis involves finding a function, under a given set of assumptions, that illustrates the relationship between the dependent variable and the independent variable. When the number of independent variables is only one, and the relationship between the dependent and the independent variable is assumed to be a straight line, this type of regression analysis is called “**simple linear regression**”. The straight-line relationship is known as the line of best fit or the regression line.

In the following figure, it can be seen that the crime rate per capita and the median value of owner-occupied homes for the city of Boston, the largest city in the Commonwealth of Massachusetts. Regression analysis is used to gain an insight into what drives crime rates in the city. Such analysis can help policymakers with decision-making directed towards the reduction of the crime rate. In such scenarios, there is a dependent variable named “*crime rate*” with an independent variable, named Median value of owner-occupied homes. In other words, the variation in crime rates based on the different neighborhoods is to be explored.



In the above figure, **regression** **line** is shown in bold. Ignoring the poor quality of the fit of the regression line to the data in the figure, a decline in crime rate per capita can be seen as the median value of owner-occupied homes increases.

Almost one-third of the regression line has no data points scattered around it at all. Many data points are just clustered on the horizontal axis around the zero crime rate mark.A good regression line that fits the data well must lie amidst a cloud of data points. It can be seen that the relationship between the crime rate per capita and the median value of owner-occupied homes is not as linear as one may have thought initially.

One of the methods used to determine the regression line is called the method of Least Squares.

# The Method of Least Squares

The simple linear regression line involves unknown constants, β0, β1 that represent the intercept and slope of the regression line respectively. The slope indicates the steepness of a line and the intercept indicates the location where it intersects an axis. This relationship can be represented by the equation:

**Y ≈ β0+ β1x**

# Model Coefficients

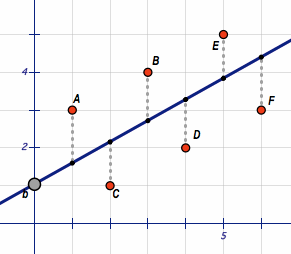
The above-mentioned unknown variables are known as model coefficients or parameters. Such a regression line is known as the population regression line, and as a probabilistic model, it fits the dataset approximately. The model is called probabilistic because it does not model all the variability in the dependent variable.

### Error

The difference between the actual dependent variable value and the predicted dependent variable value gives an error that is commonly known as the residual(εi) By repeating the calculation for every data point in the sample, the residual for every data point can be squared, to eliminate algebraic signs, and added together to obtain the Error sum of squares(ESS).

The least-squares method seeks to minimize the ESS.

# **Linear Squares Method Visual**



# Multiple Linear Regression

Multiple regression is a broader and extended class of regressions that involves the linear and nonlinear regressions with multiple explanatory variables. In other words, it ois a regression model that estimates the relationship between a dependent variable and two or more independent variables using a straigt line.

In order to understand it better, let us assume the three independent variables (X1, X2, X3) for the case where we want to fit a linear regression model. Multiple linear regression equations can be represented as:

**Y**≈ **β0+ β1X1+β2X2+ β3X3**

Each independent variable has its own coefficient (β1, β2, β3). These coefficients determine the influence of the change in their respective independent variable on a dependent variable.

# Estimating the Regression Coefficients

The regression coeffiecients in the above mentioned formula are estimated using the same least squares approach as discussed earlier in simple linear regression. In order to satisfy the least squares method, the chosen coefficients must minimize the sum of squared residuals.

We will make use of the Python programming language to evaluate these coeffiecient estimates practically.

### Logarithmic Transformations of Variables

Sometimes the relationship between the dependent and independent variables is not linear. This factor limits the use of linear regression. So, a logarithmic function can be used to transform the variable of interest according to the required nature of the relationship. Then the transformed variable tends to have a linear relationship with the other untransformed variables. It enables the use of linear regression to fit the data. This will be analyzed better in the exercises.

### Correlation Matrices

A linear relationship between two variables has already been analyzed using a straight-line graph. There is another method to visualize the linear relationship between the variables, known as a **correlation matrix**. A correlation matrix is like a cross-table of entries showing the correlation between pairs of variables. It illustrates how strongly the two variables are connected. But analyzing the raw figures in a table is not an easy task. So, a correlation matrix can be converted to a form of “**heatmap**” so that the correlation between variables can easily be observed using different colors. Such an example is shown in Exercise 2.0

# Regression Analysis using Python

All of the regression analyses done throughout this course are based on the “**Miami Housing Dataset**”.

We have loaded Python modules along with the dataset we need for analysis. After loading the necessary python modules required for analysis, we have loaded the Miami-housing .csv assigned the variable name “Miami\_data”. Removal of null values, checking duplicate values, listing and renaming of columns have been done. A correlation Matrix for the train\_dataset has also been made.

# Exercise 2.01

We have loaded Python modules and the dataset of "Miami Housing" for analysis. We have divided the DataFrame into training and test sets. Then we have plotted a correlation matrix for the train data set.

Open the following link to get started:

<https://bit.ly/3w1tsMJ>

Here is the practice exercise. Another "Housing" Dataset is to be used for practice. Outputs are there to visualize the results. You are required to complete this exercise:

<https://bit.ly/37kuoBr>

# The Correlation Coefficient

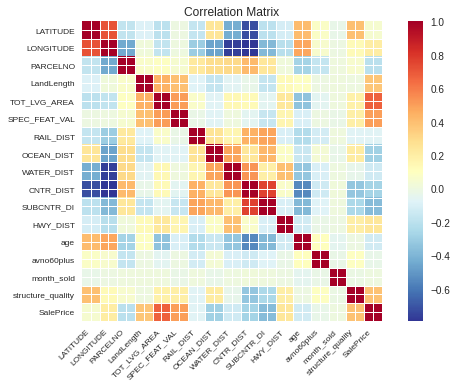
As mentioned before that a correlation matrix heatmap can be used to visualize the relationships between pairs of variables. The same relationships in numerical form can also be seen using the raw correlation coefficient numbers. These are values between -1 and +1. Pandas provide the “**corr**” function, which when called on Data Frame provides a matrix of the correlation of all numeric data types. This code to build a  correlation matrix is as follows:

corrMatrix = train\_data.corr(method = 'pearson')

Data scientists use the correlation coefficient as a statistic in order to measure the linear relationship between two variables, X and Y. In the case of the bivariate data, the correlation coefficient is commonly represented by “r”. The common method to measure the correlation between two variables is by using the Pearson correlation coefficient.

The value of “r” lies between -1 and +1. The value “+1” means that the relationship between the two variables X and Y means that both the variables increase or decrease in the same direction. While the value “-1” means that the increase in one variable “X” will decrease Y. The value “0” indicates “**No relationship**” between the two variables.

A correlation matrix of the training dataset:



# Exercise 2.02

We have used the technique to investigate any linear relationship that may exist between “**SalePrice**” and “**LandLength**” of houses to be sold in Miami. So scatter graphs fitted with a  regression line are made and the data is analyzed.

In this code, we have set the x and y labels, font size and name labels, x and y limits, and the "**tick**" parameters of matplotlib graph object (**ax**).

Open the following link to get started:

<https://bit.ly/3w2RVRL>

Here is another practice exercise. The same "Housing" Dataset is to be used for practice. Outputs are there to visualize the results. You are required to complete this exercise:

<https://bit.ly/3MP8Ywz>

# Exercise 2.03

In this exercise, we have transformed the variable by applying the Logarithm function on it to see if it provides a better fit of the regression line to the data.

We will also look at the usage of confidence intervals by including the 95% confidence interval of the regression coefficients on the plot.

Open the following link to get started:

<https://bit.ly/3pZ86eY>

Here is another practice exercise. The same "Housing" Dataset is to be used for practice. Outputs are there to visualize the results. You are required to complete this exercise:

<https://bit.ly/3pZSxE1>

# Exercise 2.04

### The StatsModels Formula API:

It can be seen in Exercise 2.02, a solid line represents the relationship between the Sale Price and Length of Land. Direct relation has been revealed. Now in order to obtain an equation that describes the line, Python provides a rich “**Application Programme Interface (API)**. The statsmodels formula API enables the data scientist to utilize the formula language to define the regression models.

We have examined a simple linear regression model where the “Sale Price” is the dependent variable” and the “Land Length” is the Independent Variable. We have utilized the statsmodels formula API to generate a linear regression model.

Open the following link to get started:

<https://bit.ly/3w2z3lO>

Here is another practice exercise. The same "Housing" Dataset is to be used for practice. Outputs are there to visualize the results. You are required to complete this exercise:

<https://bit.ly/3MIV1QR>

# Analyzing the Model

The .fit method provides many functions to explore its output. With these functions, the parameters of the model, the confidence intervals, and the p-values and t-values for the analysis can be retrieved from the results.

The .fit() method can be used in the following way:

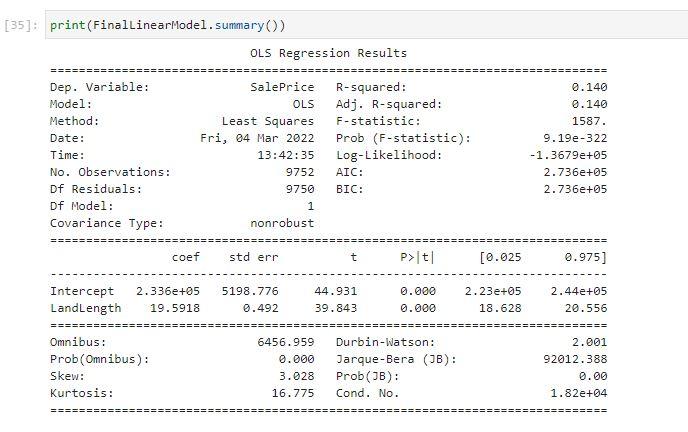
FinalLinearModel = Regression\_model.fit()

### Syntax

Its syntax includes a “**dot**” notation, after the variable name containing the results, with a relevant function name. **For example:**

linearModelResult.conf\_int()

This will output the confidence interval values. **summary()** function presents a table of all relevant results from the analysis. It will look like this visual:



### The Model Formula Languages

Python provides a competitive option for statistical analysis and modeling rivaling R and SAS. This involves the R-style formula language, by which statistical models can be easily defined. Statsmodels implements the R-style formula language by using the “**Patsy**” Python library to convert formulas and data to the matrices that are used in model fitting.

### Intercept Handling

The intercept of a model is defined by the “**string 1**” in Patsy formula strings. As the intercept is required most of the time, "string 1” is automatically included in every formula string definition. But if you want to delete the intercept from your model, then you can subtract one from the formula string that will define a model that passes through the origin.

### Exercise 2.05:

### Multiple Regression Analysis

Only one independent variable has been related to a dependent variable in this chapter. In order to model the variability to a higher level of accuracy, it is necessary to investigate all the independent variables that may contribute towards explaining the variability in the dependent variable. **Patsy** formula string will be used with the + operator to define a linear regression model that includes more than one independent variable in this exercise.

Open the following link to get started:

<https://bit.ly/3uc61hp>

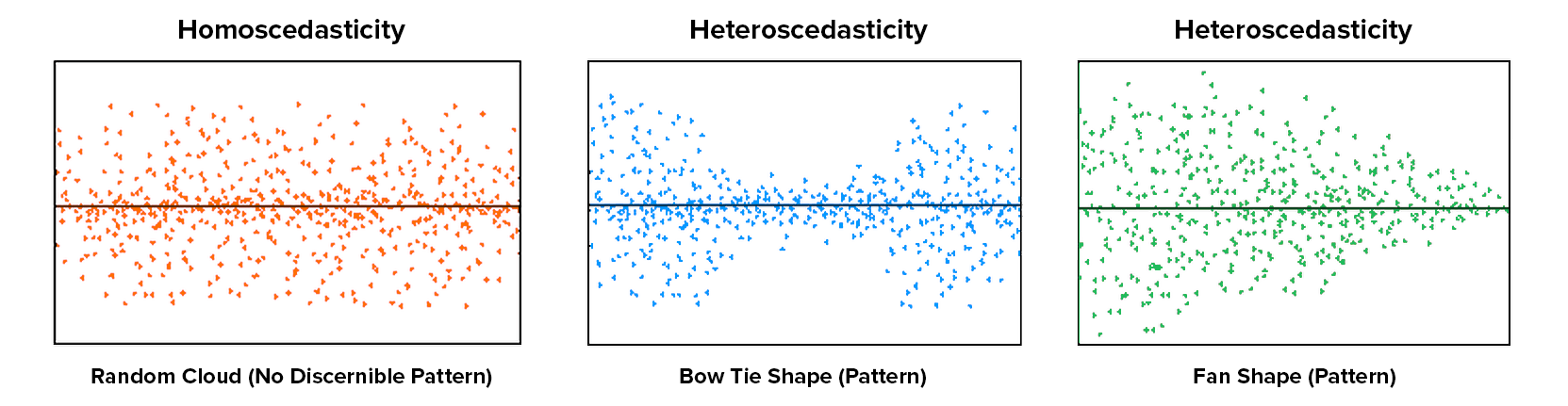
Here is another practice exercise. The same "Housing" Dataset is to be used for practice. Outputs are there to visualize the results. You are required to complete this exercise:

<https://bit.ly/3tSBBAp>

# Assumptions of Regression Analysis

Linear Regression analysis makes certain assumptions about the data it analyzes. It is necessary to check analysis to ensure the regression assumptions are not violated. Some of the main assumptions are as follows:

* The relationship between the two variables must be additive and linear.
* The residual terms must be normally distributed. This is so that the standard error of the estimate is calculated correctly.
* The residual term must have constant variance – ***homoskedasticity***. When this is not ensured, we have the heteroskedasticity problem. It refers to the variance of residual terms. It is assumed to be constant.



* The residual terms must not be correlated. When there is a correlation in the residual terms, we have the problem known as autocorrelation. The knowledge of one residual term must not give us information about what the next residual term will be.
* There should be no correlation among the independent variables. When the independent variables have correlation among themselves, we have to face a problem called multicollinearity.

# Results of Regression Analysis

Finding a model that explains the variability observed in a dependent variable is the primary objective of regression analysis. The statistic that measures how well a regression model explains this variability is called an **R-squared**. It is also known as the ***Coefficient of determination***. In addition to this, there are some other definitions:

**Total Sum of Squares(TSS) -**It gives us the measure of the total variance found in the dependent variable from its mean value.

**Regression Sum of Squares (RSS) -**It gives the measure of the amount of variability in the dependent variable that our model predicts. If the creation of a perfect model with no errors in prediction is assumed, then in such a scenario, TSS will be equal to RSS. A hypothetically perfect model will provide an explanation for all the variability we see in the dependent variable with respect to the mean value. But this is not so practical. Instead, we create models that are not perfect, so RSS is less than TSS. The missing amount by which RSS falls short of TSS is the amount of variability in the dependent variable that our regression model is not able to explain. This quantity is called **Error Sum of Squares (ESS)**.

**R-Squared** is the ratio of RSS to TSS. Thus it provides a percentage measure of how much variability a regression model is able to predict compared to the total variability in the dependent variable with respect to the mean. R-squared will become smaller when RSS grows smaller and vice versa.

# F-Test

The F-test validates the overall statistical significance of the strength of the relationship between a model and its dependent variables. If the p-value for the F-test is less than the chosen α-level, we reject the null hypothesis and conclude that the model is statistically significant overall. An F-value is generated upon fitting a regression model. This value can be used to determine whether the test is statistically significant. An increase in R2 increases the F-value. So, the larger the F-value, the better the chances of the overall statistical significance of the model. A good F-value is expected to be larger than one.

# T-Test

After the determination of a statistically significant model, the next step is to examine the significance of individual independent variables in the model. It is a statistical test that is used to compare the means of two groups. It is often used in hypothesis testing to identify whether a process has an effect on the population of interest.

The T-test is a parametric test of difference. It makes the same assumptions as other parametric tests.

# Regression using Scikit-Learn

### Linear Regression with Scikit-Learn

As we have seen in previous chapters, Linear Regression is a machine learning algorithm based on Supervised learning. Regression models are designed to model a target prediction value on the basis of independent variables. It was also seen how to implement the Linear Regression.

This section involves the demonstration to use the various Python Libraries to implement regression on a given data set. Sklearn is used for the same purpose. You have to follow these steps:

* Import the necessary libraries:

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

* Load and read the Dataset:

data = pd.read\_csv('<filename>.csv')

* Exploring the data Scatter
* Cleaning the Model
* Train the Model using this syntax:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3)

* Explore the results using:

y\_pred = regr.predict(X\_test)

plt.scatter(X\_test, y\_test, color ='b')

plt.plot(X\_test, y\_pred, color ='k')

plt.show()

### Logistic Regression Using Scikit-Learn

Logistic Regression is done to predict a category or classify objects or things into categories.

### Steps

After loading and cleaning the data, the following steps are to be followed for Training:

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y, test\_size=0.25,random\_state=0)

from sklearn.preprocessing import StandardScaler

sc\_X=StandardScaler()

X\_train=sc\_X.fit\_transform(X\_train)

X\_test=sc\_X.transform(X\_test)

* Feature Scaling is used to ensure that we get all the features on the same scale.

#Fitting logistic regession model

from sklearn.linear\_model import LogisticRegression

classifier=LogisticRegression(random\_state=0)

classifier.fit(X\_train,y\_train)

* Prediction of Results:

y\_pred=classifier.predict(X\_test)

* Exploring the Results:

from sklearn.metrics import confusion\_matrix

cm=confusion\_matrix(y\_test,y\_pred)

# Summary - Chapter 2

This chapter involves Regression related concepts. We learned that regression analysis is a supervised machine learning problem. The fundamentals of Linear regression analysis were learned. The usage of the Python module “***Pandas***” was also seen in this chapter. We have seen the correlation matrices along with fitting a line of best fit through bivariate data. The usage of the Statsmodels module was also seen.

It was also explored how to deal with multiple regression analysis. We discussed the R-squared statistic as a measure of the goodness of fit for regression models. It was also learned how to validate a regression model globally using the F-statistic. The statistical significance of individual model coefficients was made to be checked using a T-test. “Housing” based datasets have been used in this chapter that would allow a lot of hands-on learning.

Further, we will be moving to Chapter 3, which is based upon another important topic, “***Binary Classification***” that is mainly concerned with the categorization among supervised Data Science Problems.