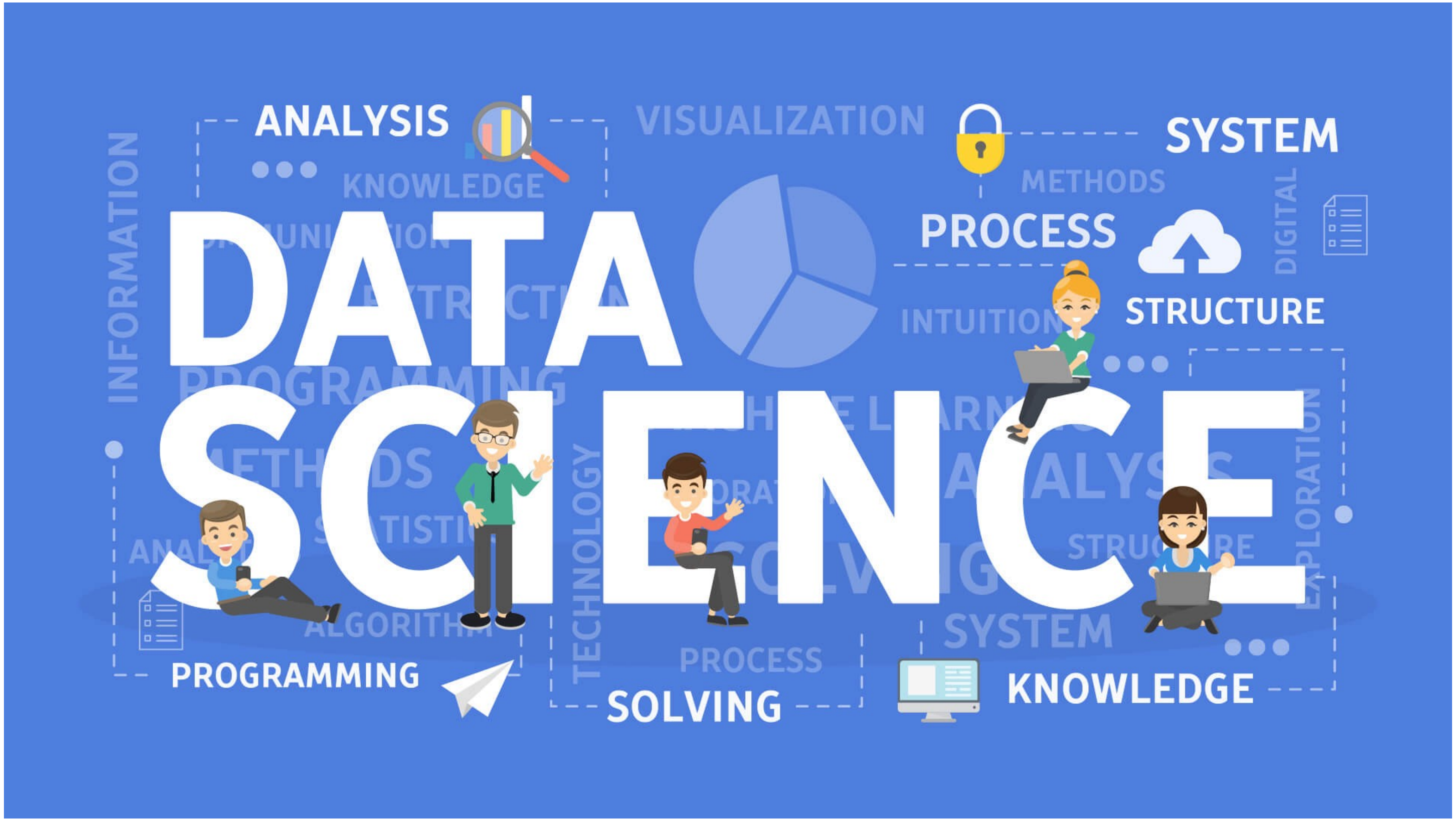
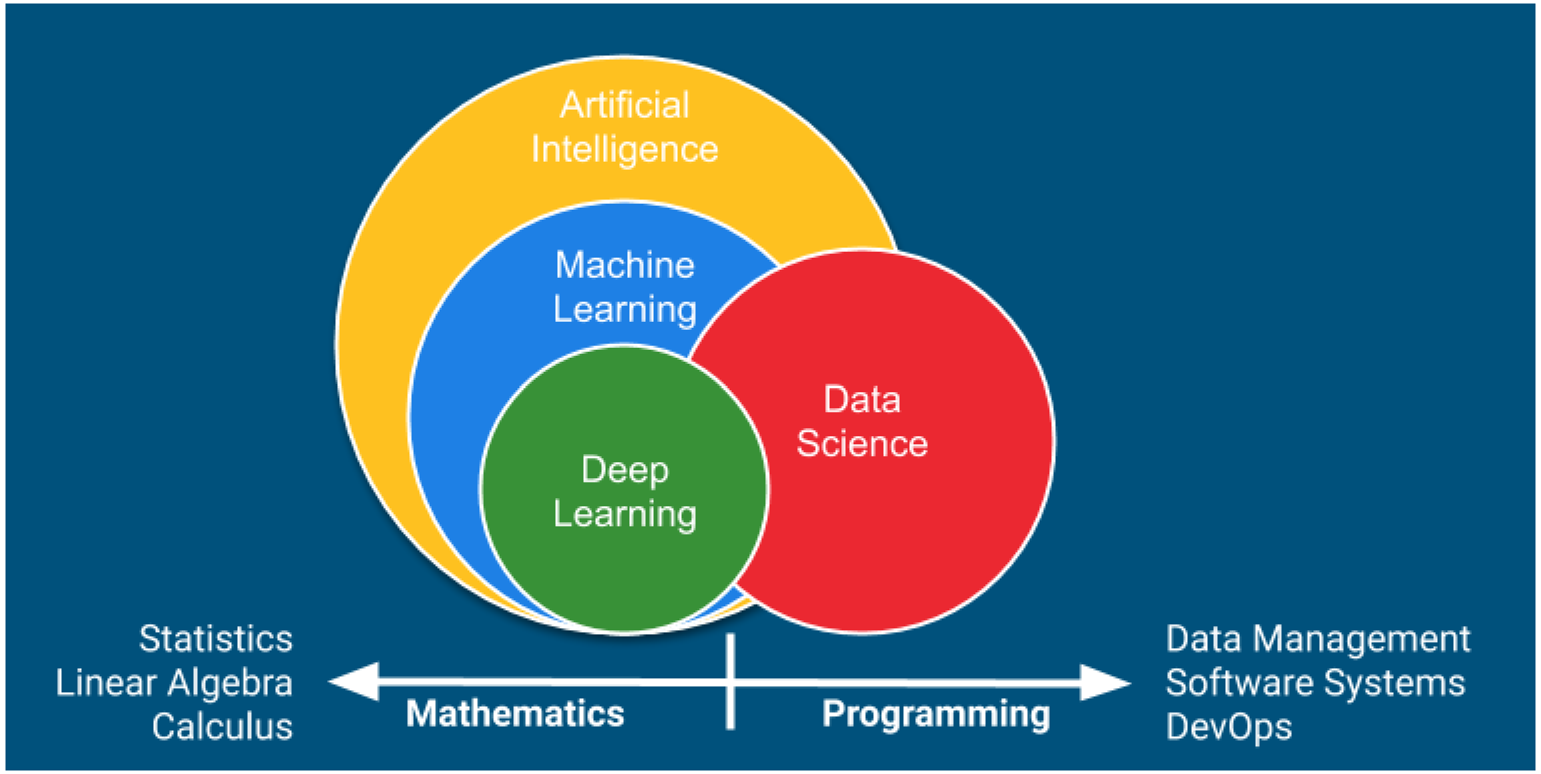


## What is Data Science?

Data Science is an amalgamation of Statistics, Computer Science and specific domain knowledge.

As more and more data gets generated across the world, we need to leverage it to make decisions and improve them.

A data scientist performs operations on the data provided to analyze and interpret it.



## Why Python For Data Science?

Most industry experts recommend starting your Data Science journey with Python

Across biggest companies and startups, Python is the most used language for Data Science and Machine Learning Projects

Stackoverflow survey for 2019 had Python outrank Java in the list of most loved languages

Python is a very versatile language since it has a wide array of functionalities already available.

The sheer range of functionalities might sound too exhaustive and complicated, you don’t need to be well-versed with them all.

Most data scientists have a few go-to libraries for their daily tasks like:

- for performing data cleaning and analysis - pandas
- for basic statistical tools – numpy, scipy

- for data visualization – matplotlib, seaborn
- Python has rapidly become the go-to language in the data science space and is among the first things recruiters search for in a data scientist's skill set.
- it consistently ranks top in global data science surveys and its widespread popularity will only keep on increasing in the coming years.
- Over the years, with strong community support, this language has obtained a dedicated library for data analysis and predictive modelling.

## Which companies use Python?

Many of the biggest and most popular companies use Python. Some of them are:

- Google, NASA, Amazon
- Social networking sites like Instagram, Reddit, Quora, etc
- Media streaming companies like Netflix and Spotify
- Rideshare companies like Uber and Lyft

**“Python has been an important part of Google since the beginning and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we are looking for more people with skills in this language.”**

- **Peter Norvig, Director of Research at Google Inc.**

## What do I need to start with Python for Data Science course?

- Google Colab
- A working laptop / desktop with 4 GB RAM
- A working Internet connection

This is all it takes for you to learn one of the popular language in Data Science.

## How much Python do I need to know to enter Data Science?

- Though Python has hundreds of libraries and many more functionalities, you don’t need to know all of them for learning Data Science.
- Rather than becoming an expert in the entire language, you would need to just be acquainted with the basic syntax of Python.
- We will also cover the most popular libraries used by Data Scientists and which you would be using too as a future Data Scientist!

## What if I don’t have Python installed on my system?

One of the best things about Python is the wide variety of platform that support writing it.

We will provide easy to follow instructions to work with Python using Google Colab, an extremely popular platform. No matter what Operating System you are using, we have you covered with guides for all of them.

## What are the most popular open-source libraries that Python supports?

- pandas, numpy, scipy, matlplotlib, seaborn are used for Data Science and Data Analysis
- scikit-learn, tensorflow, keras are used for basic and advanced machine learning
- libraries for deep learning like OpenCV(Computer Vision), NLTK(Natural Language Processing)

## Will I be able to apply what I have learnt here to machine learning and data science projects?

- course is designed to help you completely understand Python and start using it immediately for Data Science projects.
- If nothing else, we want to leave you with the process of data science and give you a higher level of understanding of what Data Science actually mean. An example of this is when one study says coffee is bad for you and the next month a study comes out saying coffee is good for you — and sometimes the studies are based on the same data! Determining what your results mean, beyond simple interpretations, is where the really hard parts of data science and statistics meet and are worthy of a book all their own. At the end of our data science journey, you will know more about the processes involved in answering some of these questions.

## Working with Big, Big Data

- Big data is a term used to refer to large and complex datasets that are too large for traditional data processing software (read databases, spread sheets, and traditional statistics packages like SPSS) to handle.
- The industry talks about big data using three different concepts, called the “Three V’s”: volume, variety, and velocity.

**Volume:-** Volume refers to how big the dataset is that we are considering. It can be really, really big — almost hard-to-believe big. For example, Facebook has more users than the population of China. There are over 250 billion images on Facebook and 2.5 trillion posts. That is a lot of data. A really big amount of data.

**variety:-**Note that photos are very different data types from temperature and humidity or location information. Sometimes they go together and sometimes they don’t. Photos (as we discovered in Book 4, “Artificial Intelligence and Python”) are very sophisticated data structures and are hard to interpret and hard to get machines to classify. Throw audio recordings in on that and you have a rather varied set of data types.

**Velocity:-**Velocity refers to how fast the data is changing and how fast it is being added to the data piles. Facebook users upload about 1 billion pictures a day, so in the next couple of years Facebook will have over 1 trillion images. Facebook is a high veloc- ity dataset. A low velocity dataset (not changing at all) may be the set of tempera- ture and humidity readings from your house in the last five years. Needless to say, high velocity datasets take different techniques than low velocity datasets.

## Cooking with Gas: The Five Step Process of Data Science

We generally can break down the process of doing science on data (especially big data) into five steps.

1. Capture the data
2. Process the data
    - Data Cleaning
    - Data Wrangling
    - Exploratory Data Analysis
3. Analyze the Model(statistical)
4. Communicate the Results
5. Maintain the data and results.

Advance steps

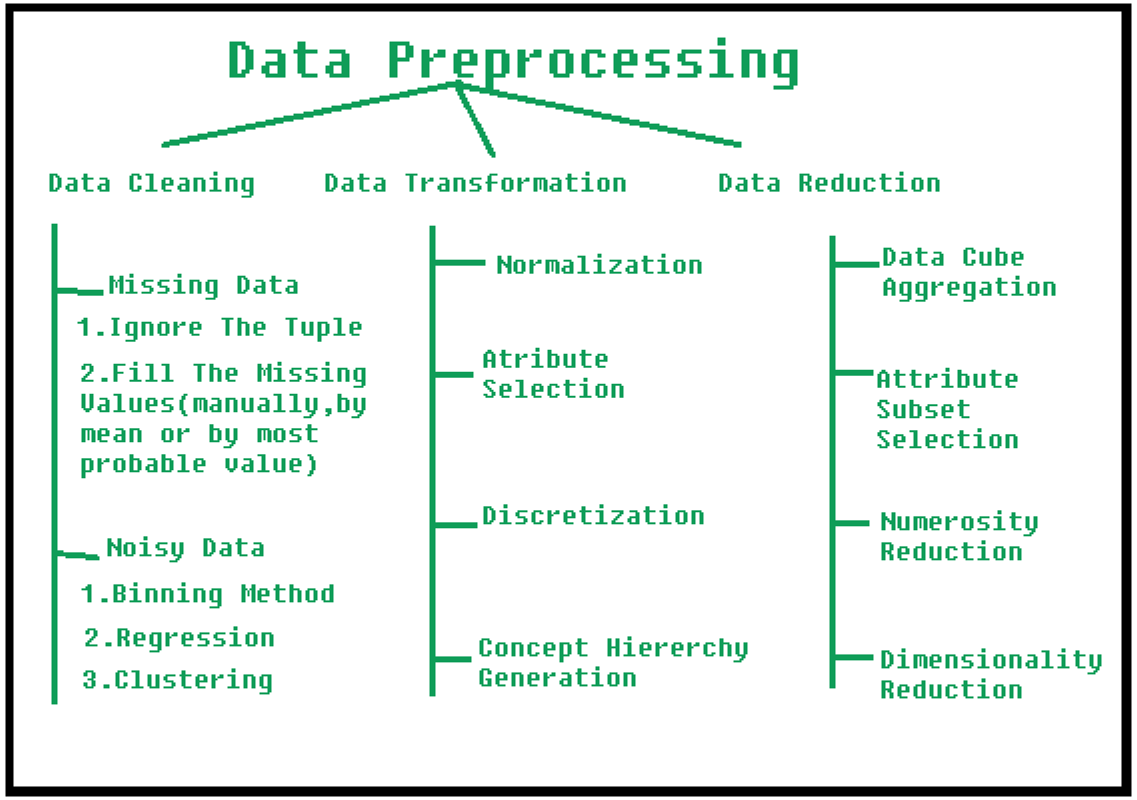
- Machine Learning
- Machine Learning Engineering

## Capturing the data

To have something to do analysis on, you have to capture some data. In any real- world situation, you probably have a number of potential sources of data. Inven- tory them and decide what to include. Knowing what to include requires you to have carefully defined what your business terms are and what your goals are for the upcoming analysis. Sometimes your goals can be vague in that sometimes, “you just want to see what you can get” out of the data.

If you can, integrate your data sources so it is easy to get to the information you need to find insights and build all those nifty reports you just can’t wait to show off to the management.

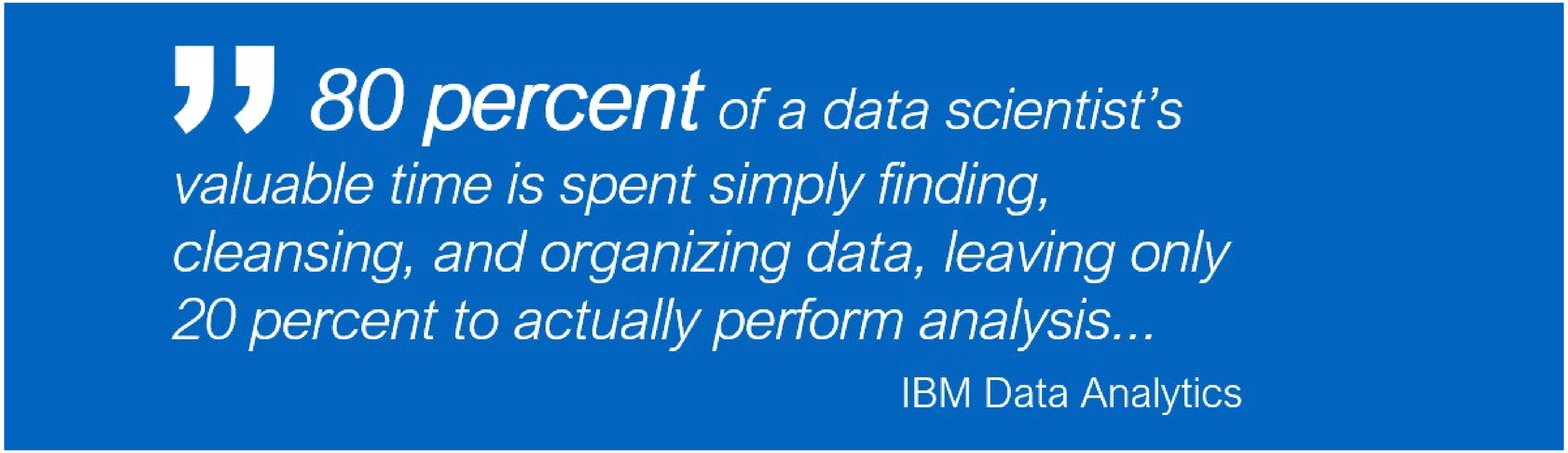
## Processing the data



In my humble opinion, this is the part of data science that should be easy, but it almost never is. I’ve seen data scientists spend months massaging their data so they can process and trust the data. You need to identify anomalies and outliers, eliminate duplicates, remove missing entries, and figure out what data is incon- sistent. And all this has to be done appropriately so as not to take out data that is important to your upcoming analysis work. It’s not easy to do in many cases. If you have house room temperatures that are 170 degrees C, it is easy to see that this data is wrong and inconsistent. (Well, unless your house is burning down.)

Cleaning and processing your data needs to be done carefully or else you will bias and maybe destroy the ability to do good inferences or get good answers down the line. In the real world, expect to spend a lot of time doing this step.

## Data Cleaning



> 80 percent of a data scientist’s valuable time is spent simply finding, cleansing, and organizing data, leaving only 20 percent to actually perform analysis...
>
> IBM Data Analytics

Data scientists spend a large amount of their time cleaning datasets and getting them down to a form with which they can work. In fact, a lot of data scientists argue that the initial steps of obtaining and cleaning data constitute 80% of the job.

Therefore, if you are just stepping into this field or planning to step into this field, it is important to be able to deal with messy data, whether that means missing values, inconsistent formatting, malformed records, or nonsensical outliers.

In this tutorial, we’ll leverage Python’s Pandas and NumPy libraries to clean data.

We’ll cover the following:

Dropping unnecessary columns in a DataFrame

Changing the index of a DataFrame

Using .str() methods to clean columns

Using the DataFrame.applymap() function to clean the entire dataset, element-wise

Renaming columns to a more recognizable set of labels

Skipping unnecessary rows in a CSV file

```
# import the pandas library
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f',
'h'],columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])

print (df)
```

```
        one       two     three
a  1.483700 -1.607389 -1.868171
b       NaN       NaN       NaN
c -0.508871  0.480661 -0.201371
d       NaN       NaN       NaN
e  0.585770  0.816944 -0.433578
f -0.255262 -0.641285 -2.080108
g       NaN       NaN       NaN
h  0.039093 -0.844791 -0.450252
```

## data wrangling

More often than not, you find yourself dealing with a lot of data, which is of no use to you in its raw form. The process of cleaning the data enough to input to the analytical algorithm is known as Data Wrangling. It is also referred to as Data Munging.

So, if you will ask any data analysts, data scientists or statisticians on which task they spend most of their time. The answer will be data cleaning or data wrangling and data munging, and not coding or running a model that uses the data.

```
import pandas as pd
left = pd.DataFrame({
         'id':[1,2,3,4,5]
         'id':[1,2,3,4,5],
         'Name': ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],
         'subject_id':['sub1','sub2','sub4','sub6','sub5']})
right = pd.DataFrame(
         {'id':[1,2,3,4,5],
         'Name': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],
         'subject_id':['sub2','sub4','sub3','sub6','sub5']})
pd.merge(left, right, how='inner', on=None, left_on=None, right_on=None,
left_index=False, right_index=False, sort=True)


print (left)
print (right)
```

```
   id    Name subject_id
0   1    Alex       sub1
1   2     Amy       sub2
2   3   Allen       sub4
3   4   Alice       sub6
4   5  Ayoung       sub5
   id   Name subject_id
0   1  Billy       sub2
1   2  Brian       sub4
2   3   Bran       sub3
3   4  Bryce       sub6
4   5  Betty       sub5
```

## Exploratory Data Analysis

Exploratory Data Analysis or (EDA) is understanding the data sets by summarizing their main characteristics often plotting them visually. This step is very important especially when we arrive at modeling the data in order to apply Machine learning. Plotting in EDA consists of Histograms, Box plot, Scatter plot and many more. It often takes much time to explore the data. Through the process of EDA, we can ask to define the problem statement or definition on our data set which is very important.

## ASSIGNMENT

## The Business Problem

Our client is a credit card company. They have brought us a dataset that includes some demographics and recent financial data (the past six months) for a sample of 30,000 of their account holders. This data is at the credit account level; in other words, there is one row for each account (you should always clarify what the definition of a row is, in a dataset). Rows are labeled by whether in the next month after the six month historical data period, an account owner has defaulted, or in other words, failed to make the minimum payment.

## Goal

Your goal is to develop a predictive model for whether an account will default next month, given demographics and historical data. Later in the book, we'll discuss the practical application of the model.