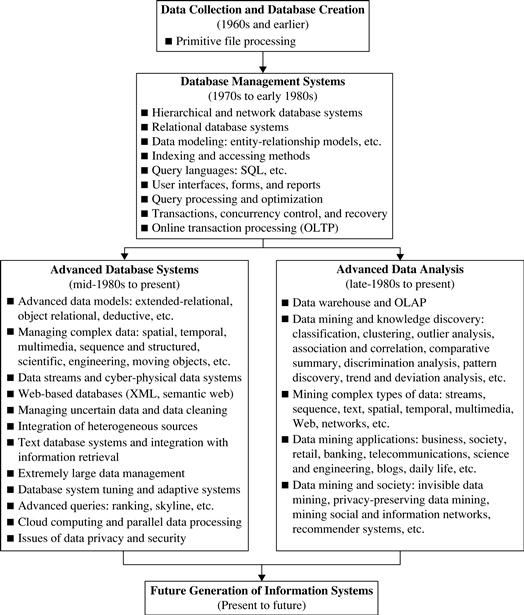
Data Mining (or KDD Knowledge Discovery from Database) concepts:

This explosively growing, widely available, and gigantic body of data makes our time truly the data age. Powerful and versatile tools are badly needed to automatically uncover valuable information from the tremendous amounts of data and to transform such data into organized knowledge.

**Data mining turns a large collection of data into knowledge**

Data mining can be viewed as a result of the natural evolution of information technology

KDD steps:

1. **Data cleaning** (to remove noise and inconsistent data)

2. **Data integration** (where multiple data sources may be combined)[3](https://www.safaribooksonline.com/library/view/data-mining-concepts/9780123814791/xhtml/ST0170.html#FN0020)

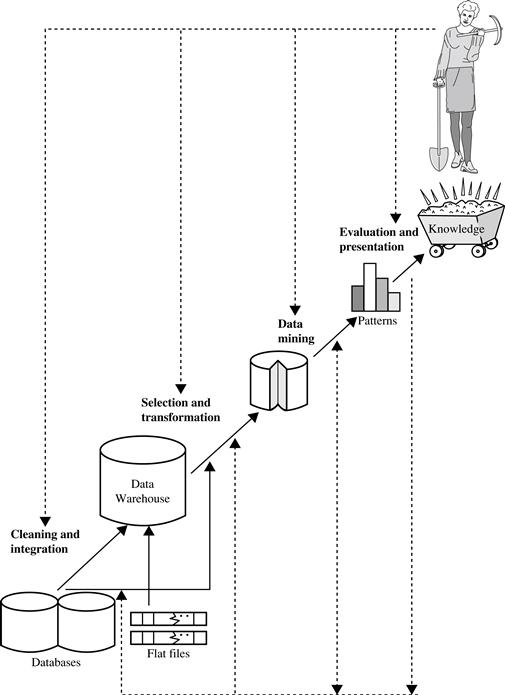
3. **Data selection** (where data relevant to the analysis task are retrieved from the database)

4. **Data transformation** (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations)[4](https://www.safaribooksonline.com/library/view/data-mining-concepts/9780123814791/xhtml/ST0170.html#FN0025)

5. **Data mining** (an essential process where intelligent methods are applied to extract data patterns)

6. **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on interestingness measures —see [Section 1.4.6](https://www.safaribooksonline.com/library/view/data-mining-concepts/9780123814791/xhtml/ST0055.html#S0085))

7. **Knowledge presentation** (where visualization and knowledge representation techniques are used to present mined knowledge to users)



What kind of data to be mined:

1. Database data DBMS
2. Data warehouse data
3. Transactional data

Relational DB data: RDBMS

**Python**

**What is better: SAS (Statistical Analysis System), R, Python?**

[**SAS**](https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/learning-path-business-analyst-sas/)**:**SAS has been the undisputed market leader in commercial analytics space. The software offers huge array of statistical functions, has good GUI (Enterprise Guide & Miner) for people to learn quickly and provides awesome technical support. However, it ends up being the most expensive option and is not always enriched with latest statistical functions.

[**R**](https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/learning-path-r-data-science/)**:**R is the Open source counterpart of SAS, which has traditionally been used in academics and research. Because of its open source nature, latest techniques get released quickly. There is a lot of documentation available over the internet and it is a very cost-effective option.

[**Python**](https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/learning-path-data-science-python/)**:** With origination as an open source scripting language, Python usage has grown over time. Today, it supports libraries (numpy, scipy and matplotlib) and functions for almost any statistical operation / model building you may want to do. Since introduction of pandas, it has become very strong in operations on structured data.

Compare these languages on following attributes: Score 1-low, 5-high

1. Availability / Cost: SAS-2, R-5, Pyhotn-5
2. Ease of learning: SAS (similar to SQL)-4.5, R (low level programming code)-2.5, Python (no GUI)-3.5
3. Data handling capabilities: all languages have parallel data handling capability. SAS-4, R-4, Python-4
4. Graphical capabilities: SAS (just function, graph difficult)-3, R-4.5, Python (call native lib or R lib)-4
5. Advancements in tool: adapting new changes. SAS-4, R-4.5, Python-4
6. Job scenario: SAS-4.5, R-3.5, Python-2.5
7. Customer service support and Community, SAS-4, R-3.5, Python-3

Conclusion

Clearly, there is no winner in this race yet. It will be pre-mature to place bets on what will prevail, given the dynamic nature of industry. Depending on your circumstances (career stage, financials etc.)

**Needless to say, Python still has few drawbacks too:**

It is an interpreted language rather than compiled language – hence might take up more CPU time. However, given the savings in programmer time (due to ease of learning), it might still be a good choice.

#### **Why Python 2.7?**

1. Awesome community support! This is something you’d need in your early days. Python 2 was released in late 2000 and has been in use for more than 15 years.
2. Plethora of third-party libraries! Though many libraries have provided 3.x support but still a large number of modules work only on 2.x versions. If you plan to use Python for specific applications like web-development with high reliance on external modules, you might be better off with 2.7.
3. Some of the features of 3.x versions have backward compatibility and can work with 2.7 version.

#### **Why Python 3.4?**

1. Cleaner and faster! Python developers have fixed some inherent glitches and minor drawbacks in order to set a stronger foundation for the future. These might not be very relevant initially, but will matter eventually.
2. It is the future! 2.7 is the last release for the 2.x family and eventually everyone has to shift to 3.x versions. Python 3 has released stable versions for past 5 years and will continue the same.

There is no clear winner

**Benefits of Python 2.7**

* Awesome online community. Easier to find answers when you get stuck at places.
* Tonnes of third party libraries

**Benefits of Python 3.5**

* Cleaner and faster
* It is the future!

### **Python Data Structures**

1. **List**: Lists are one of the most versatile data structure in Python. A list can simply be defined by writing a list of comma separated values in square brackets. Lists might contain items of different types, but usually the items all have the same type. Python lists are mutable and individual elements of a list can be changed.
2. **Strings** – Strings can simply be defined by use of single (‘), double (”) or triple (”’) inverted commas. Strings enclosed in tripe quotes (”’) can span over multiple lines and are used frequently in docstrings (Python’s way of documenting functions). \ is used as an escape character. Please note that Python strings are immutable, so you can’t change part of strings.

Raw string used to pass as is. Interpreter does not alter the raw string eg (add r):

Stmt = r’\n is a newline character by default. ’

1. **Tuples** – A tuple is represented by a number of values separated by commas. Tuples are immutable and the output is surrounded by parentheses so that nested tuples are processed correctly. Additionally, even though tuples are immutable, they can hold mutable data if needed.

Since Tuples are immutable and can not change, they are faster in processing as compared to lists. Hence, if your list is unlikely to change, you should use tuples, instead of lists.

Tuple\_ex=1, 2, 3, 4, 5

1. **Dictionary –**Dictionary is an unordered set of *key: value* pairs, with the requirement that the keys are unique (within one dictionary). A pair of braces creates an empty dictionary: {}.

### **Python Iteration and Conditional Constructs**

1. **For loop:**

for i in [Python Iterable]:

expression(i)

fact=1

for i in range(1,N+1):

fact \*= i

if [condition]:

\_\_execution if true\_\_

else:

\_\_execution if false\_\_

**Following are a list of libraries, you will need for any scientific computations and data analysis:**

* **NumPy** stands for Numerical Python. The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms,  advanced random number capabilities and tools for integration with other low level languages like Fortran, C and C++
* **SciPy** stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.
* **Matplotlib** for plotting vast variety of graphs, starting from histograms to line plots to heat plots.. You can use Pylab feature in ipython notebook (ipython notebook –pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.
* **Pandas** for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python’s usage in data scientist community.
* **Scikit Learn** for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
* **Statsmodels** for statistical modeling. Statsmodels is a Python module that allows users to explore data, estimate statistical models, and perform statistical tests. An extensive list of descriptive statistics, statistical tests, plotting functions, and result statistics are available for different types of data and each estimator.
* **Seaborn** for statistical data visualization. Seaborn is a library for making attractive and informative statistical graphics in Python. It is based on matplotlib. Seaborn aims to make visualization a central part of exploring and understanding data.
* **Bokeh** for creating interactive plots, dashboards and data applications on modern web-browsers. It empowers the user to generate elegant and concise graphics in the style of D3.js. Moreover, it has the capability of high-performance interactivity over very large or streaming datasets.
* **Blaze** for extending the capability of Numpy and Pandas to distributed and streaming datasets. It can be used to access data from a multitude of sources including Bcolz, MongoDB, SQLAlchemy, Apache Spark, PyTables, etc. Together with Bokeh, Blaze can act as a very powerful tool for creating effective visualizations and dashboards on huge chunks of data.
* **Scrapy** for web crawling. It is a very useful framework for getting specific patterns of data. It has the capability to start at a website home url and then dig through web-pages within the website to gather information.
* **SymPy** for symbolic computation. It has wide-ranging capabilities from basic symbolic arithmetic to calculus, algebra, discrete mathematics and quantum physics. Another useful feature is the capability of formatting the result of the computations as LaTeX code.
* **Requests** for accessing the web. It works similar to the the standard python library urllib2 but is much easier to code. You will find subtle differences with urllib2 but for beginners, Requests might be more convenient.

**Additional libraries, you might need:**

* **os** for Operating system and file operations
* **networkx** and **igraph** for graph based data manipulations
* **regular expressions** for finding patterns in text data
* **BeautifulSoup** for scrapping web. It is inferior to Scrapy as it will extract information from just a single webpage in a run.

**We will take you through the 3 key phases:**

1. **Data Exploration** – finding out more about the data we have
2. **Data Munging** – cleaning the data and playing with it to make it better suit statistical modeling
3. **Predictive Modeling** – running the actual algorithms and having fun

**Pandas:**

2 key data structure in Pandas library:

1. Series: 1D array
2. DataFrames: 2D array

**Analytics:**

Systematic computational analysis of data or statistics

**Business Analytics types:**

1. applied business analytics
2. theoretical business analytics

**Applied Business Analytics** – This is the work, where the emphasis is to solve a problem at hand. What matters is that you have a strategy / algorithm, which is better than what is happening currently. You deal with practical problems in this stream – messy data, missing values, bad data capture etc. More than 95% of times, you would typically use the algorithm and outputs straight from the tools. As long as you are aware of the assumptions in your model, can check whether they are holding good and interpret the output of the algorithms correctly, you are good for applied business analytics.

* 1. For example, if you know the assumptions in Linear regression and can interpret what is R-square and adjusted R-square, you would be good to apply Linear Regression.

**Theoretical Business Analytics** – This is the research area in business analytics. When you have a problem, where current set of algorithms are already optimized and applying standard techniques would not provide you any further uplift. This is when, you need to get into statistical details of various algorithms and then improve them.

One of the first things a business analyst needs to do is understand various distributions of parameters and population.

There are 3 variety of measures, required to understand a distribution:

* Measure of Central tendency
* Measure of dispersion
* Measure to describe shape of curve

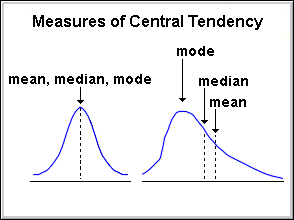
#### **Measures of Central tendency:**

Measures of central tendencies are measures, which help you describe a population, through a single metric. For example, if you were to compare Saving habits of people across various nations, you will compare average Savings rate in each of these nations.

Following are the measures of central tendency:

* Mean – or the average
* Median – the value, which divides the population in two half
* Mode – the most frequent value in a population

The following image illustrates how mean, median and mode would be placed in a couple of scenarios:

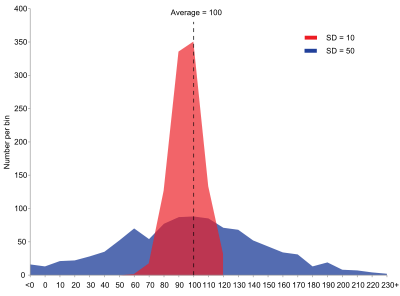
[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/central_tendency.gif)

Among the three measures, mean is typically affected the most by Outliers (unusually high or low values), followed by the median and mode.

#### **Measures of Dispersion:**

Measures of dispersion reveal how is the population distributed around the measures of central tendency.

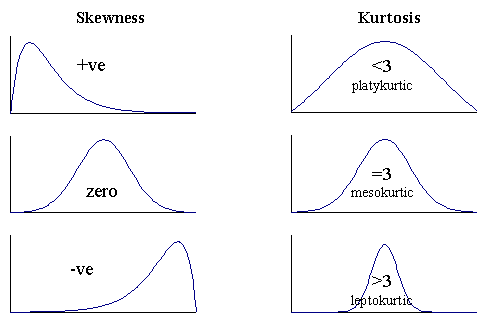
* **Range** – Difference in the maximum and minimum value in the population
* **Quartiles** – Values, which divide the populaiton in 4 equal subsets (typically referred to as first quartile, second quartile and third quartile)
* **Inter-quartile range**– The difference in third quartile (Q3) and first quartile (Q1). By definition of quartiles, 50% of the population lies in the inter-quartile range.
* **Variance:**The average of the squared differences from the Mean.
* **Standard Deviation:** is square root of Variance



Difference in distribution of 2 populations with same mean, median and mode. Source: Wikipedia

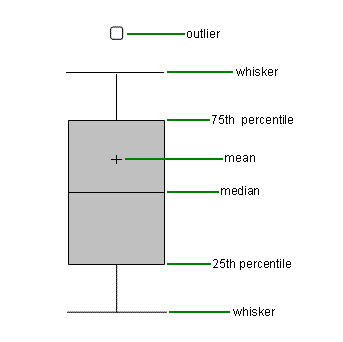
#### **Measures to describe shape of distribution:**

* **Skewness** – Skewness is a measure of the asymmetry. Negatively skewed curve has a long left tail and vice versa.
* **Kurtosis** – Kurtosis is a measure of the “peaked ness”. Distributions with higher peaks have positive kurtosis and vice-versa

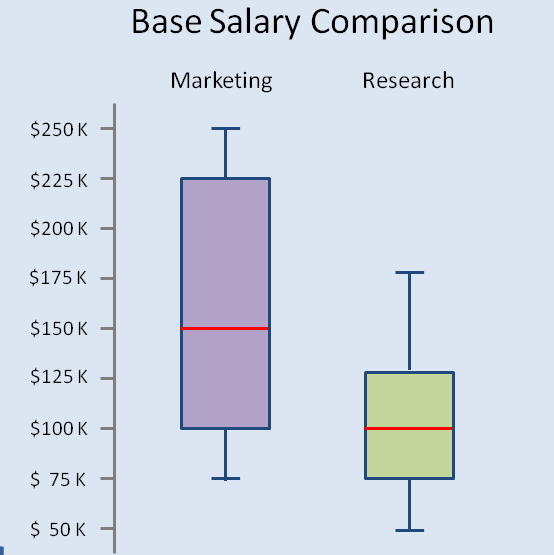
[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/skewness-and-kurtosis.gif)

**A few practical tips to understand distributions better:**

* **Use of box plots:** Box plots are one of the easiest and most intuitive way to understand distributions. They show mean, median, quartiles and Outliers on single plot.

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/box-plot.gif)

* You can use box plots next to each other for various categories / segments of population to understand overlap / differences in the population. Following is an example of one such comparison (with illustrative data):

[](https://www.analyticsvidhya.com/blog/wp-content/uploads/2014/07/box_plot_comparison.png)In this post, we looked use of statistics to plot and understand distributions of populations – first steps for any business analyst to do in a project. In the articles to follow in this series, we will look at use of confidence intervals, hypothesis testing, probabilities and measures to judge various predictive models. If you would want me to cover more topics, please let me know through comments below.

**Data Science:** 3 things

* 1. **Hacking skills**: able to write code get data from anywhere. Gather right data in right format, run it in right hyper model.
  2. **Statistics and Math knowledge**: how to model problem gather right data and apply technique. Evaluate and validate different **predictive** modeling strategies.
  3. **Substantive expertise:** understand problem on which u want to apply DS techniques. Domain knowledge, business problem.

**ML = Hacking skills + Math & Statistics knowledge**

**Traditional research = Math & Statistics + Substantive expertise.**

**Danger Zone = Hacking skills + Substantive expertise**

**Data Science = Hacking Skills + ML + Math & stats + Traditional research + Substantive expertise + Danger zone**

**Opportunities:**

* 1. **Better Infrastructure and Tool:** Focus on Abstraction/ Automation and **Scalability.**

2 options:

* + 1. Build projects on top of open source like dataRobot or
    2. Start building open source like libraries scikit tlearn, pandas. Main wish lists are: Shared memory integration, better Hadoop integration, native support for big data, distributive in-memory data frame, easy to use yet powerful deep learning.
  1. **Specific Use cases for applying Machine Learning:** find a problem u are passionate about that can be solved using ML. Example: Medical health, drug discovery for better drugs, for health care try to predict patient health proactively and let your patient know. Also used in Internet of Things (IoT) machine data, education design learning platform, ask them question and build next type of question based on their ability to learn, Computer vision or deep learning, Robotics.
  2. **Data Science Services:** for those companies want to provide DS services. Have higher bunch of PDH students.

GIT

1. What is the difference between Git and GitHub?

Git is a version control system; think of it as a series of [snapshots](http://git-scm.com/figures/18333fig0105-tn.png) (commits) of your code. You see a path of these snapshots, in which order they where created. You can make branches to experiment and come back to snapshots you took.

GitHub, is a web-page on which you can publish your Git repositories and collaborate with other people.

1. Is Git saving every repository locally (in the user's machine) and in GitHub?

No, it's only local. You can decide to push (publish) some branches on GitHub.

1. Can you use Git without GitHub? If yes, what would be the benefit for using GitHub?

Yes, Git runs local if you don't use GitHub. An alternative to using GitHub could be running Git on files hosted on Dropbox, but GitHub is a more streamlined service as it was made especially for Git.

1. How does Git compare to a backup system such as Time Machine?

It's a different thing, Git lets you track changes and your development process. If you use Git with GitHub, it becomes effectively a backup. However usually you would not push all the time to GitHub, at which point you do not have a full backup if things go wrong. I use git in a folder that is synchronized with Dropbox.

1. Is this a manual process, in other words if you don't commit you won't have a new version of the changes made?

Yes, committing and pushing are both manual.

1. If are not collaborating and you are already using a backup system why would you use Git?
   * If you encounter an error between commits you can use the command git diff to see the differences between the current code and the last working commit, helping you to locate your error.
   * You can also just go back to the last working commit.
   * If you want to try a change, but are not sure that it will work. You create a branch to test you code change. If it works fine, you merge it to the main branch. If it does not you just throw the branch away and go back to the main branch.
   * You did some debugging. Before you commit you always look at the changes from the last commit. You see your debug print statement that you forgot to delete.

## What is GitHub?

## GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.