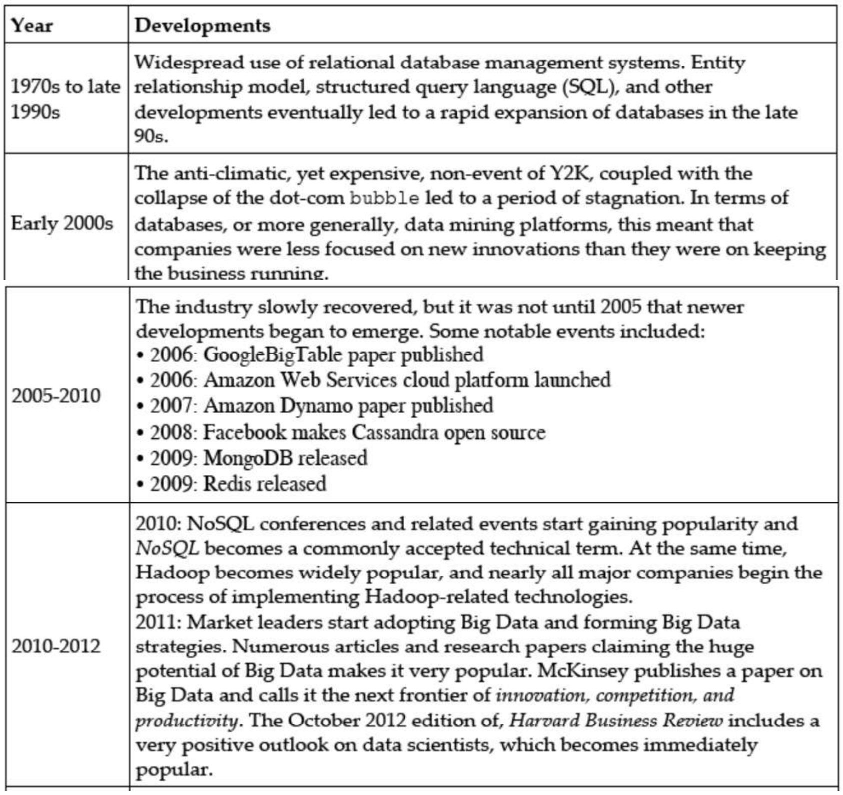
Unit V

21CSE221T – Big Data Tools and Techniques

Enterprise data science overview

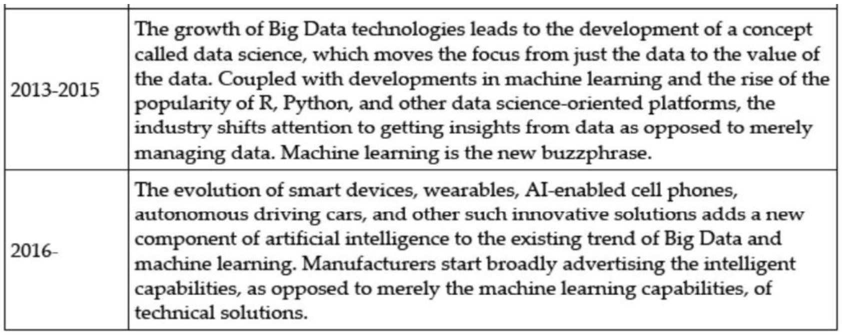
- Two categories
  - **Highly technical researchers** who used complex computing languages and/or hardware for their professional tasks
  - **Analysts** who could use tools such as Excel and BI platforms in order to perform both simple and complex data analysis

Big Data Evolution in Enterprise



Year	Developments
1970s to late 1990s	Widespread use of relational database management systems. Entity relationship model, structured query language (SQL), and other developments eventually led to a rapid expansion of databases in the late 90s.
Early 2000s	The anti-climatic, yet expensive, non-event of Y2K, coupled with the collapse of the dot-com `bubble` led to a period of stagnation. In terms of databases, or more generally, data mining platforms, this meant that companies were less focused on new innovations than they were on keeping the business running.
2005-2010	The industry slowly recovered, but it was not until 2005 that newer developments began to emerge. Some notable events included: • 2006: GoogleBigTable paper published • 2006: Amazon Web Services cloud platform launched • 2007: Amazon Dynamo paper published • 2008: Facebook makes Cassandra open source • 2009: MongoDB released • 2009: Redis released
2010-2012	2010: NoSQL conferences and related events start gaining popularity and *NoSQL* becomes a commonly accepted technical term. At the same time, Hadoop becomes widely popular, and nearly all major companies begin the process of implementing Hadoop-related technologies. 2011: Market leaders start adopting Big Data and forming Big Data strategies. Numerous articles and research papers claiming the huge potential of Big Data makes it very popular. McKinsey publishes a paper on Big Data and calls it the next frontier of *innovation, competition, and productivity*. The October 2012 edition of, *Harvard Business Review* includes a very positive outlook on data scientists, which becomes immediately popular.

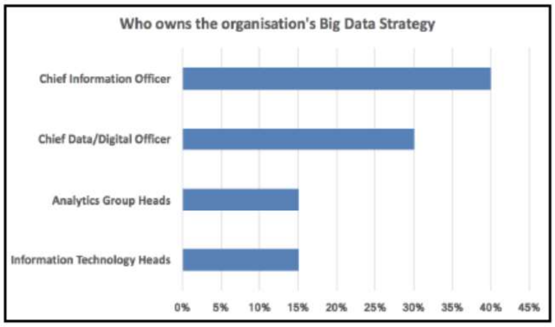
Big Data Evolution in Enterprise

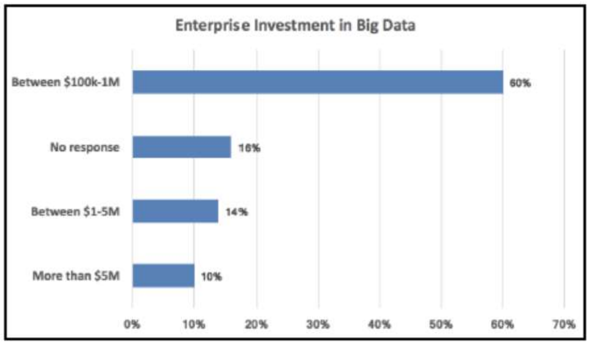


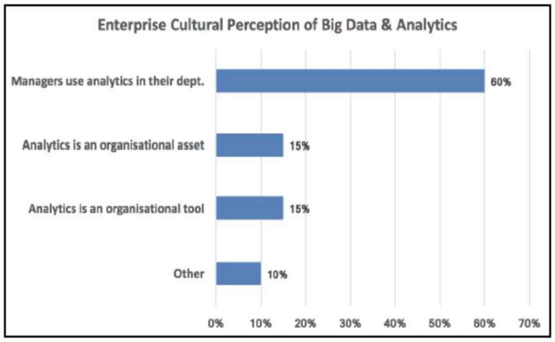
2013-2015	The growth of Big Data technologies leads to the development of a concept called data science, which moves the focus from just the data to the value of the data. Coupled with developments in machine learning and the rise of the popularity of R, Python, and other data science-oriented platforms, the industry shifts attention to getting insights from data as opposed to merely managing data. Machine learning is the new buzzphrase.
2016-	The evolution of smart devices, wearables, AI-enabled cell phones, autonomous driving cars, and other such innovative solutions adds a new component of artificial intelligence to the existing trend of Big Data and machine learning. Manufacturers start broadly advertising the intelligent capabilities, as opposed to merely the machine learning capabilities, of technical solutions.

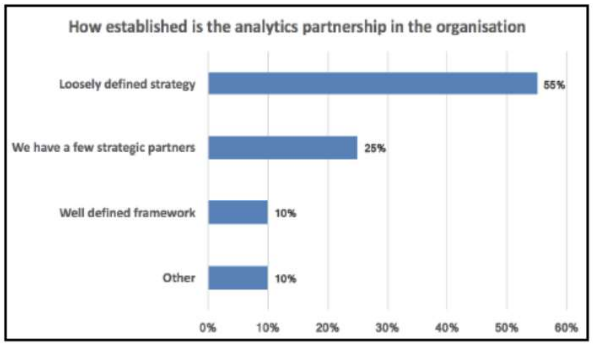
- The responsibility for implementing a Big Data Initiative, is generally delegated to the IT or Analytics Department of a company or Chief Information Officer or the Chief Data/Digital Officer

Enterprise Big Data Strategy

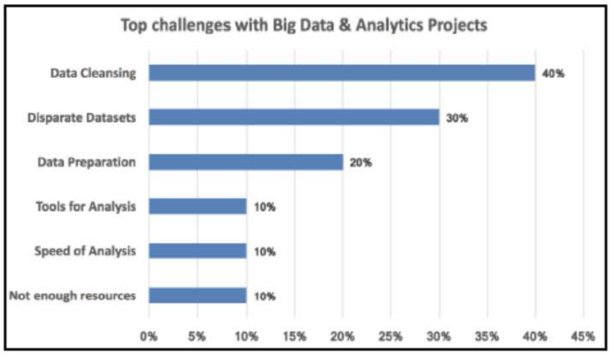

Who owns the organisation's Big Data Strategy

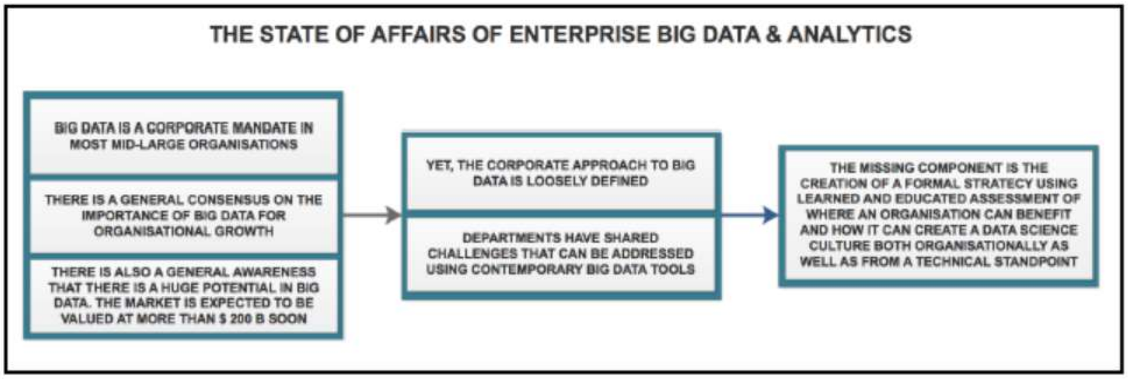

Enterprise Investment in Big Data


Enterprise Cultural Perception of Big Data & Analytics


How established is the analytics partnership in the organisation

Enterprise Big Data Strategy





THE STATE OF AFFAIRS OF ENTERPRISE BIG DATA & ANALYTICS

BIG DATA IS A CORPORATE MANDATE IN MOST MID-LARGE ORGANISATIONS

THERE IS A GENERAL CONSENSUS ON THE IMPORTANCE OF BIG DATA FOR ORGANISATIONAL GROWTH

THERE IS ALSO A GENERAL AWARENESS THAT THERE IS A HUGE POTENTIAL IN BIG DATA. THE MARKET IS EXPECTED TO BE VALUED AT MORE THAN $ 200 B SOON

YET, THE CORPORATE APPROACH TO BIG DATA IS LOOSELY DEFINED

DEPARTMENTS HAVE SHARED CHALLENGES THAT CAN BE ADDRESSED USING CONTEMPORARY BIG DATA TOOLS

THE MISSING COMPONENT IS THE CREATION OF A FORMAL STRATEGY USING LEARNED AND EDUCATED ASSESSMENT OF WHERE AN ORGANISATION CAN BENEFIT AND HOW IT CAN CREATE A DATA SCIENCE CULTURE BOTH ORGANISATIONALLY AS WELL AS FROM A TECHNICAL STANDPOINT

Enterprise Big Data Strategy

- Conclusions
  - Investment in Big Data within the enterprise, with most organizations in the $100k to $1M range. Nearly *all respondents had made* ***at least some investment*** *in Big Data*
  - Analytics was being **used by managers** in their departments, but there wasn't a uniform level of engagement across all departments
  - Cross-functional collaboration of analytics initiatives, was **loosely defined**
  - **Data cleansing** was the top most challenge in enterprise big data

A roadmap to enterprise analytics success

- The **general challenges associated with Big Data analytics are as follows:**
  - Nearly every company is investing in Big Data, machine learning, and AI
  - Often, the company has a corporate mandate
  - Finding the right use cases can be challenging
  - Even after you *find them, the outcome may be uncertain*
  - Even after you *achieve them, whether or not the optimal targets have been* identified can be elusive (for example, when using HDFS for storing only data)

A roadmap to enterprise analytics success

- G**eneral guidelines for data science and analytics initiatives:**
  - **Conduct meetings and one-on-one reviews with business partners in the** organization to review their workflows and get feedback on where analytics and/or data mining would provide the most value
  - **Identify specific aspects of business operations that are important and related** to the firm's *revenue stream; the use case would have a measurable impact once* completed
  - The use cases do not have to be *complex; they can be simple tasks, such as ML or* Data Mining
  - Intuitive, easily understood, you can explain it to friends and family

A roadmap to enterprise analytics success

- G**eneral guidelines for data science and analytics initiatives:**
    - Ideally the use case takes effort to accomplish today using conventional means. The solution should not only benefit a **range of users, but also have executive visibility**
    - Identify **Low Difficulty - High Value (Short) vs High Difficulty - High Value** (**Long) use cases**
    - Educate business sponsors, share ideas, show **enthusiasm (like a long job** interview)
    - Score **early wins for Low Difficulty - High Value, create Minimum Viable Solutions**, and get management to buy in before further enhancing the use solutions developed. (takes time)

A roadmap to enterprise analytics success

- Few examples of *low difficulty but potentially high value projects could be*
  - **Automating manual tasks** conducted on a frequent basis by a business group; for instance, reports that are created in Excel may be easily automated using a combination of open source tools and databases.
  - **Converting manual stock analytics to automated versions** using programming scripts. This could involve tasks such as creating regular tables, pivot tables, and charts that are created in Excel but can be converted into automated processes.
  - **Creating web interfaces** using R Shiny for business applications and implementing predictive analytics functionalities.
  - **Moving certain parts of the IT infrastructure to a cloud platform**. This may seem counter-intuitive, especially if the organization is not used to working in cloud environments. However, the ease and simplicity of managing cloud deployments can mean an overall reduction in the total cost of ownership and operational overhead.

Data science solutions in the enterprise

- Broad categories
  - Enterprise data warehouse and data mining
    - Traditional data warehouse systems
    - Enterprise and open source NoSQL Databases
    - Cloud databases
    - GPU databases
  - Enterprise data science: machine learning, artificial intelligence
    - The R programming language ,Python
    - OpenCV, Caffe, and others
    - Spark
    - Deep learning
    - H2O and Driverless AI
    - Datarobot
    - Command-line tools
    - Apache MADlib
    - Machine learning as a service
  - Enterprise infrastructure solutions
    - Cloud computing , Virtualization
    - Containers – Docker, Kubernetes, and Mesos
    - On-premises hardware
    - Enterprise Big Data

Enterprise data warehouse and data mining - Traditional data warehouse systems

- Oracle Exadata
  - Oracle's **high performance *Engineered Database*** *Platform*
  - *designed for resource* intensive queries.
  - expected to significantly improve query performance over non-Exadata systems
  - supports advanced software features such as in-memory computing, independent row and column-based filtering,
  - other hardware features such as support for the latest storage devices, including NVMe Non-Volatile Memory Express, in-memory fault tolerance
- Oracle Exadata is used for OLTP transactional workloads where **speed and performance is critical**.

Enterprise data warehouse and data mining - Traditional data warehouse systems

- Exalytics, and TimesTen
  - Exalytics is intended primarily for BI workloads.
  - support a higher level of flexibility regarding the choice of installed components which are installed selectively depending on client needs.
  - One of the key components of Exalytics commonly found in enterprise installations is OBIEE (Oracle Business Intelligence Enterprise Edition).
  - A complete BI suite and benefits from an underlying in-memory database called Times Ten, which is also a part of the Exalytics ecosystem.
- Exalytics is used for analytical workloads. OBIEE interface together with TimesTen provides a strongly coupled analytics environment. Available as a cloud-based service.

Enterprise data warehouse and data mining - Traditional data warehouse systems

- HP Vertica
  - a column-oriented, massively parallel processing database system with features such as support for in-database machine learning, and native integration to open source systems such as Apache Kafka and Apache Spark, multi-node hardware architecture.
  - supported on popular cloud environments such as **Amazon Web Services (AWS), Google, and Azure.**
  - supports a standard interactive SQL interface, readily compatible with most contemporary BI tools.
  - Community edition available
- Used to engage in proof of concept for large deployments provides ample opportunities for business to try and test the platform with company-specific use cases prior to making a decision.

Enterprise data warehouse and data mining - Traditional data warehouse systems

- **Teradata**
  - widely recognized as a leader in enterprise database technology
  - native integration with many open source solutions, such as R, RStudio, Jupyter, and SAS;
  - time series support; built-in analytic functions for machine learning and AI; support for a wide range of data types, such as CSV, JSON, and text, and spatial/temporal data

Teradata has been a popular enterprise database for several decades and has strong credibility with large organizations. expensive and require proper POCs to assess suitability for use cases specific to the organization.

Enterprise data warehouse and data mining - Traditional data warehouse systems

- **PostgreSQL**
  - **Traditional data bases + Added NOSQL features recently**
  - fully-functional, open source product
  - open source, is a very cost-effective way to try out a mature database without making large initial investments. It can also serve as a testing platform for trying out NoSQL features, such as handling JSON data prior to making a final decision
- **Greenplum**
  - built on top of PostgreSQL. Open source.
  - adds a number of significant analytic capabilities
  - an innovative cost-based query optimizer,
  - Integration with Apache MADlib, R, Python, and Java and choices for row or columnar storage
  - supports massively parallel architectures.
  - Proven performance for large enterprise workload
  - availability of commercial support

Enterprise data warehouse and data mining - Traditional data warehouse systems

- **SAP Hana**
  - columnar, in-memory database from SAP with support for NoSQL features
  - supports multicore parallel operations, multi-tenancy, and is fully **ACID compliant**
  - predictive modelling, streaming analytics, time series analysis, and spatial, text, and graph-based analysis
  - adds a formidable high performance database to existing SAP installations
  - high cost involved with enterprise-grade deployments, SAP is used mainly for business-critical needs

Enterprise data warehouse and data mining - Enterprise and open source NoSQL Databases

- **Kdb+**
  - fastest, most efficient and lightweight databases that has been used in applications like high-frequency trading
  - in-memory columnar storage from the outset and is technically an extension of the q programming language
  - the size of the kdb+ binary is about 500 to 600KB, small enough to fit in the L3 cache of most modern CPU
  - Simple but q programming language is hard to learn
- **MongoDB**
  - a market leader in the space of document-oriented databases
  - JSON format, a rich interface for Python, Java, and JavaScript
  - horizontal scaling and sharding, high availability,
  - No fixed schema required
  - storing unstructured or semistructured data

Enterprise data warehouse and data mining - Enterprise and open source NoSQL Databases

- **Cassandra**
  - Incorporates both columnar and key-value concepts
  - stores data in row-based partitions
  - Each partition is in turn a primary key. Rows can have multiple columns and the number of columns may differ from one row to another
  - a fully open source solution
  - has matured into a stable, enterprise-grade, open source platform
- **Neo4j**
  - graph-based database that is used to model relationships between different entities
  - used most commonly in conjunction with recommendation engines
  - Various graph-based representations such as weighted, directed, unidirectional, and labelled are available
  - Used by deep customer-level or user-level analysis such as social networks or recommendation systems

Enterprise data warehouse and data mining – Cloud Databases

- **Amazon Redshift.**
  - most prominent platform for data management in the cloud-based ecosystem.
  - It is based on PostgreSQL and is intended mainly for analytical workloads.
  - can be deployed directly from the AWS console
- **Redshift Spectrum**
  - permits the querying of data that has been stored in Amazon S3, the standard storage layer in AWS without having to load it into a Redshift specific instance.
  - relatively fast, inexpensive, and more importantly easy to use and deploy.
  - uses a pay-per-query model—users pay only for the queries that are executed at a nominal charge for each terabyte of data scanned.
- **Amazon Athena**
  - users can leverage Amazon Athena on-demand and do not need to reserve any additional hardware.

Enterprise data warehouse and data mining – Cloud Databases

- **Google BigQuery**
  - a large-scale data warehouse system that is fully cloud-based
  - Redshift requires separate provisioning
  - the *plug-and-play* equivalent of the Amazon RedShift
  - Users can query a cumulative of 1 terabyte of data at no charge per month. uses a payper-use model whereby queries have allocated costs.
  - abstracts the complexity of setting up a database and allows the end user to dedicate time to writing queries and/or performing analytics without the overhead of setting up an infrastructure.
- **Azure CosmosDB**
  - Microsoft's NoSQL cloud-based databases
  - a *multi-model database; it can support key-value pairs, document-based queries,* graph-based models, and also relational database queries.

Enterprise data warehouse and data mining – GPU Databases

- **GPU databases**
  - recent development that came with the growth of Graphics Processing Unit cards for data science related tasks, such as machine learning.
  - GPUs work best when the query can be parallelized. This is due to the fact that GPUs contain thousands of cores.
  - By delegating each core to work on a small subset of the data, a GPU can often calculate at an impressively fast rate that far exceeds the CPU-based query performance.
- **Brytlyt**
  - recent entrant in the space of GPU databases
- **MapD**
  - one of the early developers of a commercial GPU database platform
- primary challenges of GPU-based databases is the need to configure a GPUbased system properly. This can require specialized skills, as using GPU cards for computation is quite different than using GPU cards for common tasks such as rendering images

Enterprise data science – machine learning and AI

- **OpenCV, Caffe, and others**
  - Image recognition is one of the more successful areas of machine learning.
  - involves identifying objects and correctly categorizing them.
  - applications, ranging from identifying license plate numbers to face recognition, and is available in mobile devices and robotics.
  - OpenCV provides a standard interface for various image recognition tasks, and can also leverage hardware acceleration features to optimize performance.
  - Other well-known machine learning software for image processing include Caffe, cuDNN, TensorFlow, and others.
  - simply image recognition + deep learning use cases
- **Spark**
  - **MLlib** library in Spark provides a formal implementation of various machine learning algorithms that can be used in a Spark platform

Enterprise data science – machine learning and AI

- **Deep learning**
  - Neural Networks with multiple hidden layers (generally more than two) and/or nodes are generally categorized as **deep learning**
  - autonomously driving cars, are a direct result of the use of deep learning
  - for practical day-to-day tasks.
  - notable packages/frameworksinclude:
    - TensorFlow
    - cuDNN
    - Theano
    - Torch
    - PaddlePaddle, from Baidu
- **H2O and Driverless AI**
  - A popular platform for Kaggle competitions, **H2O provides a massively scalable, real-time** machine learning interface with native integration for R, Python, Spark, and much more
  - **Driverless AI is a recent addition to the H2O line of products. It aims to make machine** learning easier for practitioners by implementing an automated interface that attempts to create models and optimize accuracy by building and evaluating multiple models in an automated manner.
- **Datarobot-** similar to H2O but requires a licensing fee and can be expensive for smaller firms.

Enterprise data science – machine learning and AI

- **Command-line tools**
  - There are multiple machine learning tools that are executed at the Unix command-line.
  - There are existing interfaces for some of these tools in R, Python, and other languages that permit users to leverage their capabilities without having to use them from the Unix
  - terminal.
  - Some of the popular command-line utilities include:
    - LIBSVM
    - LIBLINEAR
    - Vowpal Wabbit
    - MLPACK
    - libFM
- **Apache MADlib**
  - perform analytics and run algorithms *in-database, as in, it can execute functions locally* without requiring an external programming interface.
  - It supports parallel processing and can work seamlessly with multiple data sources such as Greenplum, PostgreSQL, and others.
- **Machine learning as a service**
  - Cloud-based machine learning platforms that integrate with other cloud resources have also proliferated. Some of the well-known platforms include AzureML, BigML, IBM Watson, and others.

Enterprise infrastructure solutions

- **Cloud computing**
  - organizations have gradually shifted their resources to cloud based platforms
  - contain extremely sophisticated and extensive architecture to support machine learning, data mining at enterprise level
  - AMI images in Amazon's AWS, allows users to initiate a pre-built snapshot of an OS with pre-installed components
  - Hadoop and map-reduce operations in general are also supported extensively in AWS.
  - The EMR, or Elastic Map Reduce in AWS, and HDInsight in Azure, are two well-known and very popular Big Data frameworks.
- **Virtualization**
  - the process of creating isolated, self-contained environments within a larger host, has allowed organizations to consolidate servers and dramatically reduce data center footprints.
  - If, say, an organization leverages six servers for their websites, and of those, two get utilized frequently whereas the others have relatively lower loads most of the time, it may be possible to consolidate all the servers into one or two servers at most.
  - Technologies from Dell EMC, such as VxBLOCK, are well-known enterprise virtualization hardware used in physical data centers. This also allows companies to create their own private cloud infrastructure.
  - However, it can be fairly expensive and requires the proper assessment of the cost-to-benefit
  - Open source - OpenStack

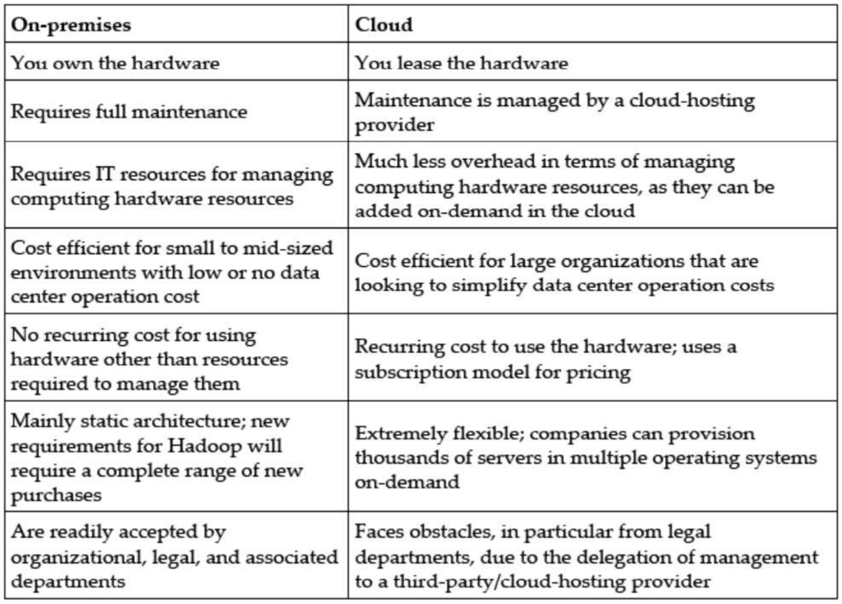
Enterprise infrastructure solutions

- **Containers – Docker, Kubernetes, and Mesos**
  - Containers, like virtualization, create isolated guest systems,
  - VMs create a completely separate environment, containers share the same kernel as the host system and hence are considered to be closer to the hardware.
  - Containers in general have a higher level of performance because they rely on and directly use features of the guest OS instead of creating a separate OS ecosystem.
  - Popular containers include Docker, CoreOS, and many others.
  - Used for the large-scale management of mainly web-related services.
  - Containers can be started up and shut down on demand much more readily than VMs, and popular cloud providers have added dedicated support for containers, making it easy to start up thousands of containers to service web requests with simply a few lines of code.
  - Orchestration software such as Kubernetes provide enterprise-grade capabilities for managing containers.
  - Mesos, not only provide support for managing containers, but also add the capability of managing other legacy hardware for application-aware scheduling and other services

Enterprise infrastructure solutions

- **On-premises hardware**
  - traditional data center, still has a place in modern-day computing.
  - With a physical data center, users do not have to pay recurring fees for cloud-based services.
  - For small to mid-sized organizations that do not have large administrative overhead, or for organizations that do not require high performance/ specialized computing capabilities, on-premises systems are fully capable of delivering cost-efficient, permanent solutions.

Enterprise infrastructure solutions



On-premises	**Cloud**
You own the hardware	You lease the hardware
Requires full maintenance	Maintenance is managed by a cloud-hosting provider
Requires IT resources for managing computing hardware resources	Much less overhead in terms of managing computing hardware resources, as they can be added on-demand in the cloud
Cost efficient for small to mid-sized environments with low or no data center operation cost	Cost efficient for large organizations that are looking to simplify data center operation costs
No recurring cost for using hardware other than resources required to manage them	Recurring cost to use the hardware; uses a subscription model for pricing
Mainly static architecture; new requirements for Hadoop will require a complete range of new purchases	Extremely flexible; companies can provision thousands of servers in multiple operating systems on-demand
Are readily accepted by organizational, legal, and associated departments	Faces obstacles, in particular from legal departments, due to the delegation of management to a third-party/cloud-hosting provider

Charts

- Line chart
- Pie chart
- Bar chart
- Heat map

Discrete and Continuous variables

- Discrete variables can only take on certain values within a range.
- Between zero and ten, there are eleven discrete values for variables.
- There will be no 2.3 or 5.78 because we don’t count these numbers when counting to 10.
- Discrete variables can be binned because they take on a finite number of values.

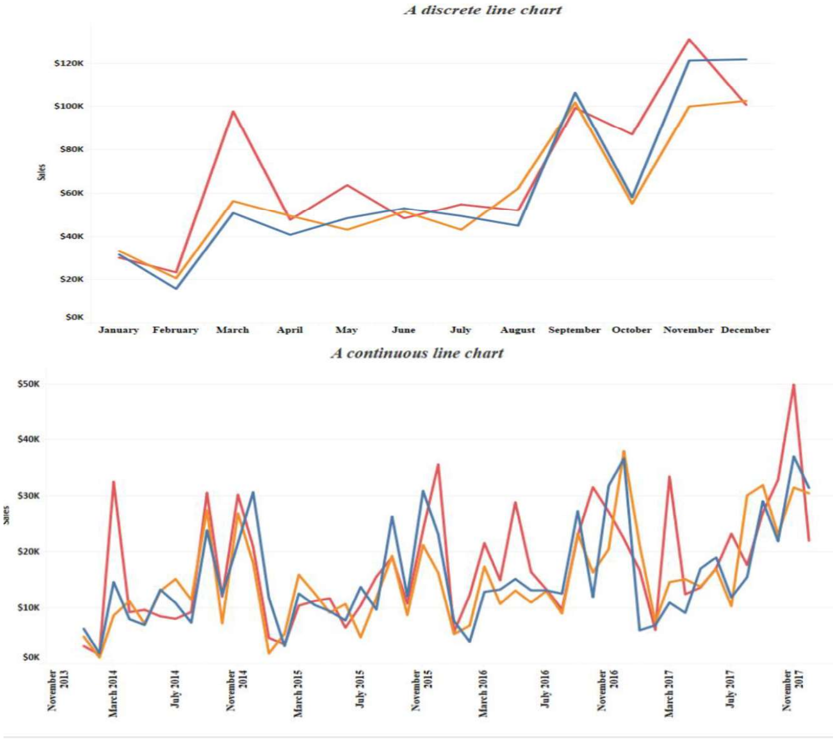
Discrete and Continuous variables

- These numbers can take on any value e.g. real numbers.
- In opposition to the counting numbers, there are an infinite number of real numbers between zero and ten.
- Continuous variables behave in a similar way in that they can take on any value.
- Examples - numeric measurements such as **Gross Profit**, **Shipping Cost** or **Inventory**.
- These are continuous values because they do not fall into distinct categories.
- A Gross Profit variable could be any monetary value, positive or negative, rather than a few distinct classes.
- Also continuous variables are not usually binned.

Line Charts

- Used to represent relation between two data on two different axes X and Y
- allow looking at the behavior of one or several variables over time and identifying the trends.
- Presents information as series of data points
- Suitable for continuous, can represent discrete data also
- Stacked line charts

Examples – Line chart



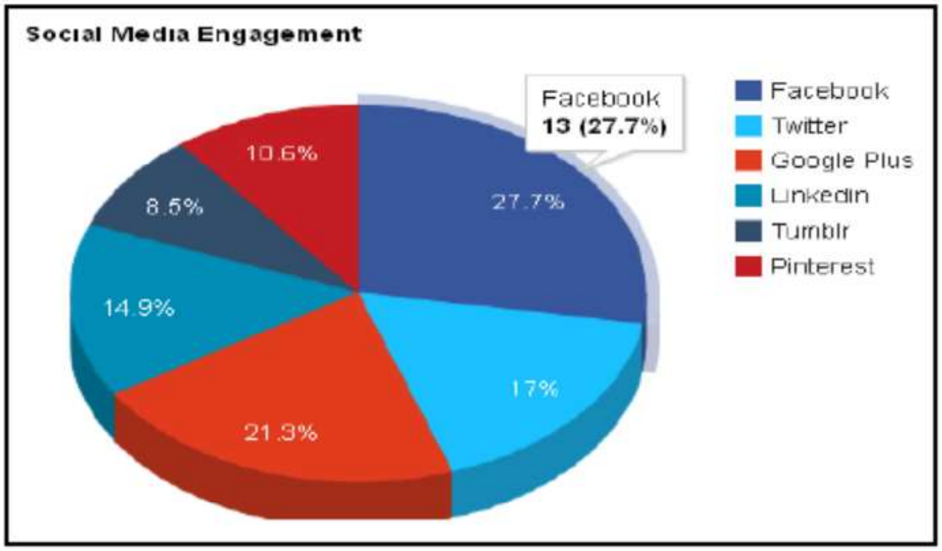
A discrete line chart

A continuous line chart

Pie Charts

- Pie charts show the components of the whole.
- Companies that work with both traditional and big data may use this technique to look at customer segments or market shares.
- The difference lies in the sources from which these companies take raw data for the analysis.

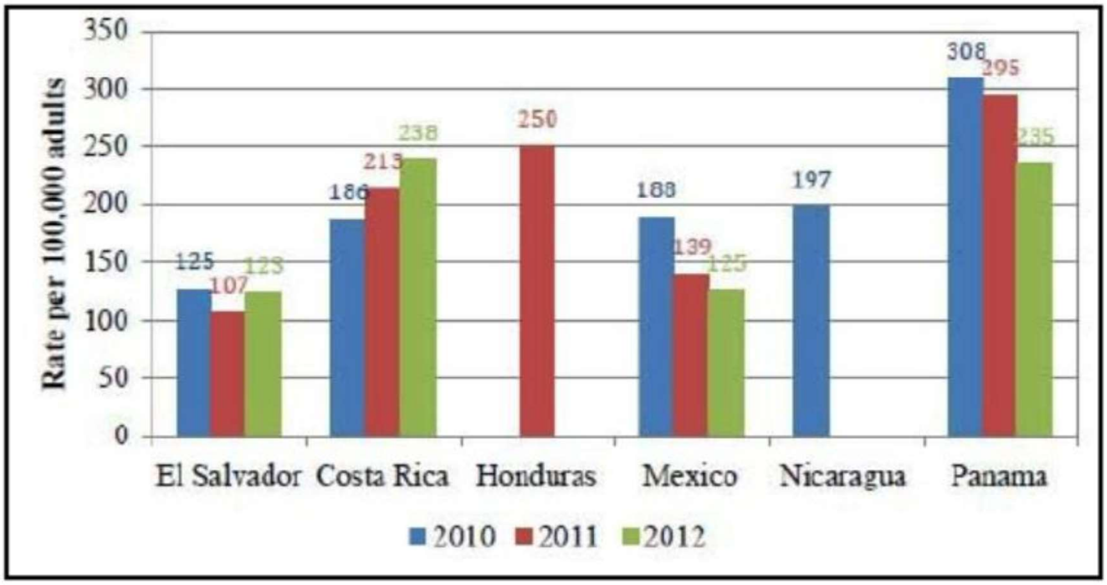
Example – Pie chart



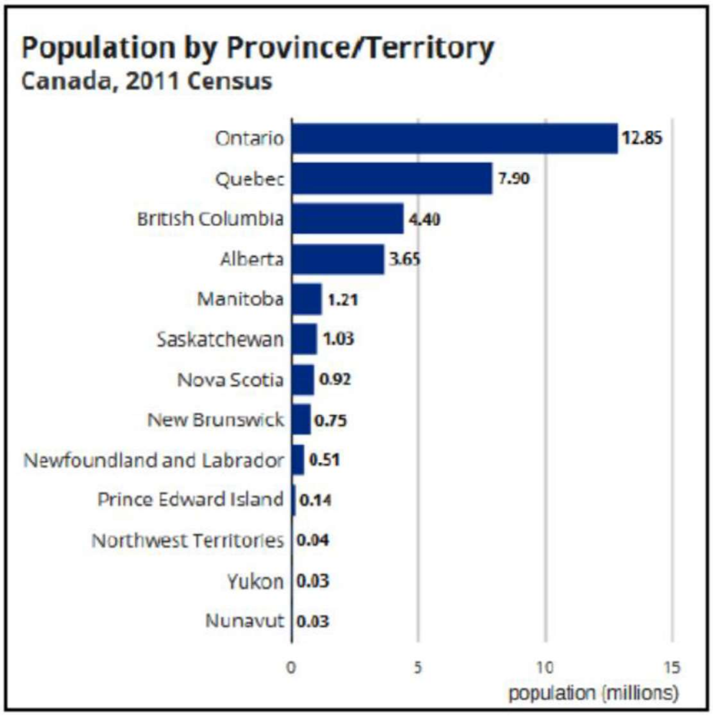
Bar charts

- Bar charts allow comparing values of different variables.
- Suitable for discrete variables
- In traditional BI, companies can analyze their sales by category, the costs of marketing promotion by channel, and so on.
- When analyzing big data, companies can look at the customer engagement, sales figures by hour, and so on.
- Types – vertical or horizontal

Example-Vertical bar chart



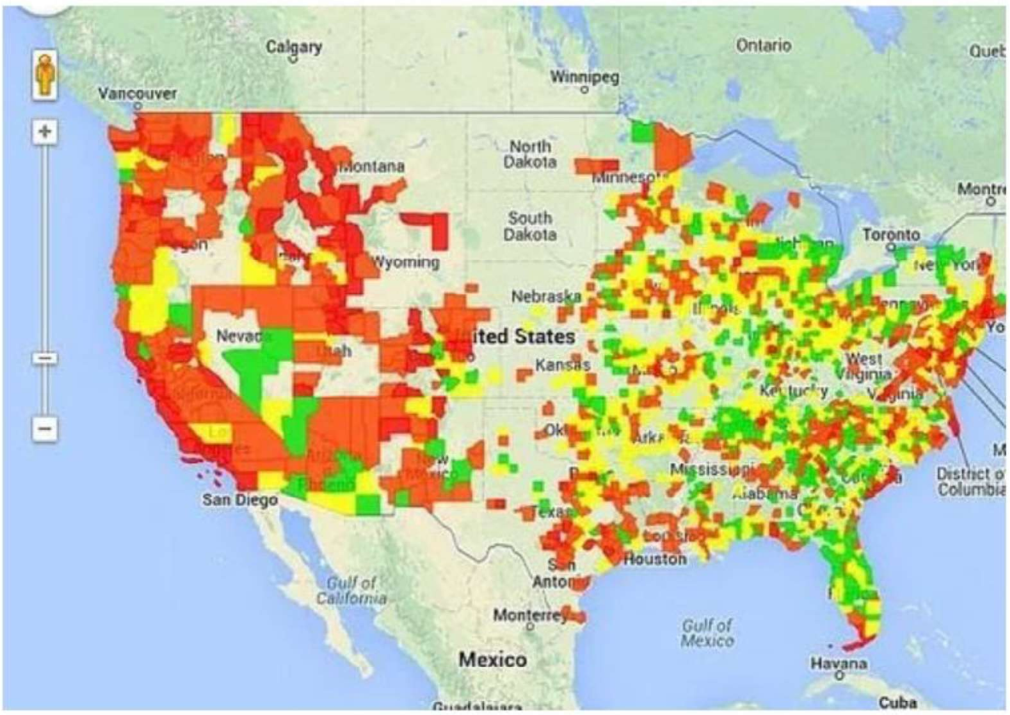
Example- Horizontal bar chart



Heat maps

- A heat map is a two-dimensional representation of data in which values are represented by colors.
- The variation in color may be by hue or intensity
- Giving obvious visual cues to the reader about how the phenomenon is clustered or varies over space.

Example – Heat map



Charts in Python

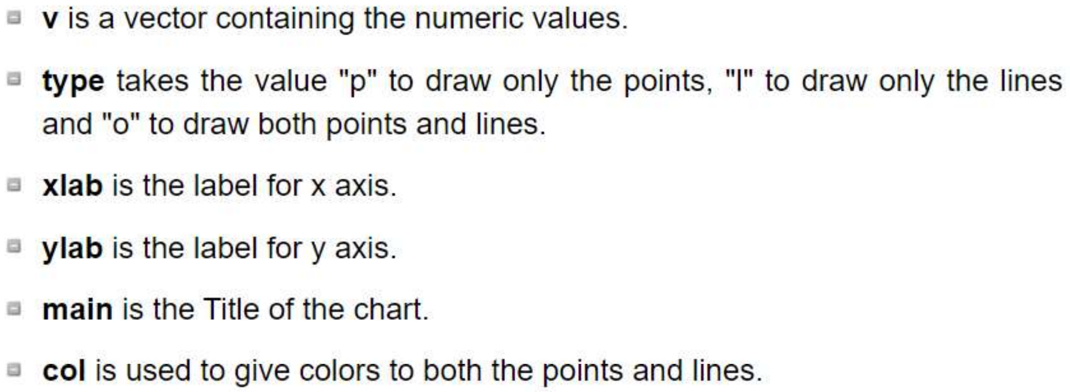
- Matplotlib is a data visualization library in Python.
- The pyplot, a sublibrary of matplotlib, is a collection of functions that helps in creating a variety of charts.

Parameters of plot function

- kind: options available hist, bar, barh, scatter, area, kde, line, box, hexbin, pie.
- figsize: Allows overwriting the default output size of 6 inches wide and 4 inches high. figsize expects a tuple (e.g., figsize=(12,8))
- title: Adds a title to the chart. It is a string.
- bins: Allows overriding the bin width for histograms. bins expects a list or listlike sequence of values (e.g., bins=np.arange(2,8,0.25))
- xlim/ylim: Allows overriding the defaults for maximum and minimum values of the axis. Both, xlim and ylim expect a tuple (e.g., xlim=(0,5))

Line charts in R

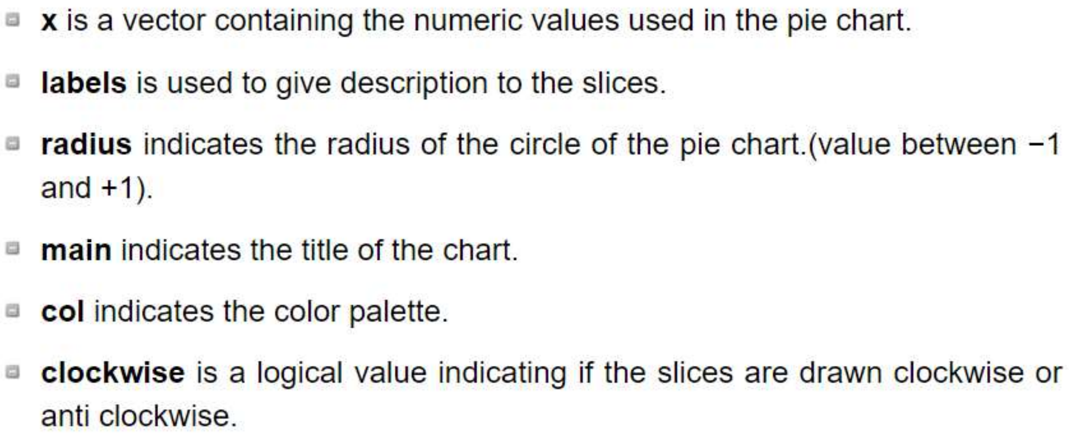
- plot(v,type,col,xlab,ylab)



- **v** is a vector containing the numeric values.
- **type** takes the value "p" to draw only the points, "l" to draw only the lines and "o" to draw both points and lines.
- **xlab** is the label for x axis.
- **ylab** is the label for y axis.
- **main** is the Title of the chart.
- **col** is used to give colors to both the points and lines.

Pie charts in R

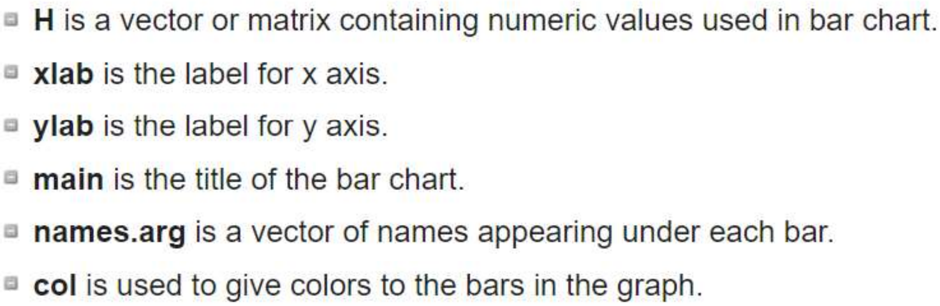
- pie(x, labels, radius, main, col, clockwise)



- **x** is a vector containing the numeric values used in the pie chart.
- **labels** is used to give description to the slices.
- **radius** indicates the radius of the circle of the pie chart.(value between −1 and +1).
- **main** indicates the title of the chart.
- **col** indicates the color palette.
- **clockwise** is a logical value indicating if the slices are drawn clockwise or anti clockwise.

Bar charts in R

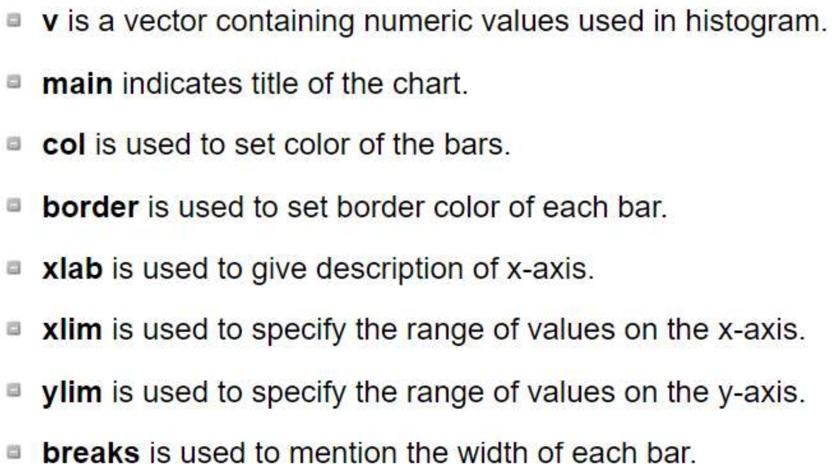
- barplot(H,xlab,ylab,main, names.arg,col)



- **H** is a vector or matrix containing numeric values used in bar chart.
- **xlab** is the label for x axis.
- **ylab** is the label for y axis.
- **main** is the title of the bar chart.
- **names.arg** is a vector of names appearing under each bar.
- **col** is used to give colors to the bars in the graph.

Histogram in R

- hist(v,main,xlab,xlim,ylim,breaks,col,border)



- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to mention the width of each bar.