Olivier Maugain

Understanding Data Literacy

Course Notes

365 DataScience

Table of Contents

ABSTRACT

Going as far back as the 19th century, data analytics has been instrumental for business success. To this day, it has continued to increase in significance across all government institutions and business domains. At the same time, the rapid technological advancements in the past decades have increased the complexity of the data ecosystem, thus driving the evolution of analytical practices.

So far, the Data Literacy Course Notes the definition of data literacy, as well as its utility, and benefits for business growth.

In this section we are going to take a comprehensive look at the more nuanced data properties by answering questions such as:

- What is data?
- What are the types of data?
- What are the 3 defining properties of Big Data?
- How to store data?
- How do past data storage methods compare to present techniques?

Keywords: Big Data, data, data literacy, data storage, data analysis,

1. Understanding Data

1.1. Data Definition

Definition:

"Data are defined as factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation."

Data should be used in plural; the singular form is datum, which is a single value of a single variable.

Data ≠ Information

- Data are the raw facts
- Information is derived from data
- Data need to be processed and interpreted in a certain context in order to be transformed into information

Examples of data:

- A Spreadsheet with sales information
- E-mails
- Browsing history
- Video files you shared on social media
- Geolocation as tracked by mobile phones
- The amount of fuel consumption recorded by vehicles

Types of data:

- Quantitative vs. qualitative data
- Structured vs. unstructured data
- Data at rest vs. data in motion
- Transactional vs. master data
- (Small) data vs. "Big" data

1.2. Types of Data

1.2.1. Qualitative vs. Quantitative Data

Quantitative data:

Data that can be measured in numerical form. The value is measured in the form of numbers or counts. Each data set is associated with a unique numerical value. Quantitative data are used to describe numeric variables. Types of quantitative data:

- Discrete data: Data that can only take certain values (counts). It involves integers (positive and negative)

  Examples:

  - Sales volume (in units)
  - Website traffic (number of visitors, sessions)

- Continuous data: Data that can take any value. It involves real numbers.

  Examples:

  - ROI (return of investment) of a project
  - Stock price

- Interval (scale) data: Data that are measured along a scale. Each point on that scale is placed at an equal distance (interval) from one another, with no absolute zero. Examples:
  - Credit score (300-850)
  - Year of the foundation of a company
- Ratio (scale) data: Data that are measured along a scale with an equal ratio between each measurement and an absolute zero (the point of origin). They cannot be negative.

  Examples:
  - Revenue
  - Age

Qualitative data:

Data that is collected in a non-numerical form. It is descriptive and involves text or categories, but also integers (when recoded). Types of qualitative data:

- *Nominal (scale) data*: Data that do not have a natural order or ranking. They cannot be measured. Calculations with these data are meaningless.

  Examples:
  - Marital status
  - Response to an email campaign (yes/no)
- *Ordinal (scale) data*: Data that have ordered categories; the distances between one another are not known. Order matters but not the difference between values.

Examples:

- Socio-economic class (“poor”, “working class”, “lower middle class”, “upper middle class”, “upper class”)
- Product rating on an online store (number of stars, from 1 to 5)

- *Dichotomous data*: Qualitative data with binary categories. They can only take two values, typically 0 or 1. Nominal or ordinal data with more than two categories can be converted into dichotomous data. Instead of X possible data values, you get X dichotomous data with a value of 0 or 1 each.

  Examples:

  - Under the age of 30 vs. Over the age of 30
  - Default on a loan

1.2.2. Structured vs. Unstructured Data

Structured data: Data that conform to a predefined data model.

- They come in a tabular (grid) format; each row is a record (case) and each column is a variable (attribute)
- These rows and columns can, but do not have to, be labeled
- The cells at each intersection of the rows and columns contain values
- They are easy to work with as everything is set up for processing

  Examples:

  - Spreadsheets
  - CSV (Comma Separated Values) files
  - Relational databases

Semi-structured data: Data that do not conform to a tabular data model but nonetheless have some structure.

- The data do not reside in fixed records or fields
- They contain elements (e.g., tags or other markers) that enforce hierarchies of records and fields within the data
- To be found extensively on the web or social media

  Examples:

  - HTML (Hypertext Markup Language) files
  - XML (Extensible Markup Language) files
  - JSON (JavaScript Object Notation, pronounced "Jason") files

Unstructured data: Data that are not organized in a predefined manner. They are not arranged according to a pre-set data model or schema and cannot be stored in the form of rows and columns.

- They come in large collections of files
- They are not easily searchable
- It is difficult to manage, analyze, and protect them with mainstream relational databases
- They require other alternative platforms for storing and processing

  Examples:

  - Text documents (word processing files, pdf files, etc.)
  - Presentations
  - Emails

- Media files (images, audio, video)
- Customer feedback (written and spoken complaints)
- Invoices
- Sensor data (as collected by mobile devices, activity trackers, cars, airplane engines, machines, etc.)

Metadata: These are data about data, providing information about other data.

- Metadata summarize basic information about data
- Metadata can be created (and enriched) automatically and manually
- It concerns various forms of data, e.g., images, videos, emails, social media posts, etc.

  Examples:

  - Author
  - Timestamp (i.e., date and time created)
  - Geo-location
  - File size
  - File type (JPEG, MP3, AVI, etc.)

1.2.3. Data at Rest vs. Data in Motion

Data at rest: Data that are stored physically on computer data storage, such as cloud servers, laptops, hard drives, USB sticks, magnetic tapes, etc.

- They are inactive most of the time, but meant for later use

- Data at rest are vulnerable to theft when physical or logical access is gained to the storage media, e.g., by hacking into the operating system hosting the data or by stealing the device itself

  Examples:

  - Databases
  - Data warehouses
  - Spreadsheets
  - Archives

Data in motion: Data that are flowing through a network of two or more systems or temporarily residing in computer memory.

- They are actively moving from device to device or network to network
- Data in motion are meant to be read or updated
- Data in motion are often sent over public networks (such as the internet) and must be protected against spying attacks

  Examples:

  - Data of a user logging into a website
  - Telemetry data
  - Video streaming
  - Surveillance camera data

1.2.4. Transactional vs. Master Data

When working with data involving larger systems, such as ERP (Enterprise Resource Planning, e.g., S/4HANA) or CRM (Customer Relationship Management, e.g.,

Salesforce Marketing Cloud), people make a distinction between two other types of data.

Transactional data: Data recorded from transactions. They are volatile, because they change frequently.

Examples:

- Purchase orders
- Sales receipts
- Bank transaction history
- Log records

Master data: Data that describe core entities around which business is conducted. Master data may describe transactions, but they are not transactional in nature. They change infrequently and are more static than transaction data.

Examples:

- Prospects
- Customers
- Accounting items
- Contracts

Illustrative example:

If a manufacturer buys multiple pieces of equipment at different times, a transaction record needs to be created for each purchase (transactional data). However, the data about the supplier itself stay the same (master data).

Challenges of master data management:

- Master data are seldom stored in one location; they can be dispersed in various software applications, files (e.g., databases, spreadsheets) or physical media (e.g., paper records)
- Various parts of the business may have different definitions and concepts for the same business entity
- Master data are often shared and used by different business units or corporate functions

1.2.5. Big Data

Big data are data that are too large or too complex to handle by conventional data-processing techniques and software. They are characterized across *three* defining properties:

1.Volume: This aspect refers to the amount of data available. Data become big when they no longer fit on a desktop or laptop RAM.

- RAM (Random Access Memory) can be likened to a computer's short-term memory
- It is fast to read from and write to
- RAM is different from storage capacity, which refers to how much space the hard disk drive (HDD) or solid-state drive (SSD) of a device can store
- The amount of RAM in a device determines how much memory the operating system and open applications can use

- Big data starts with the real-time processing of gigabytes (GB) of data, where 1 GB = 1000 megabytes (MB)- this is equivalent to a document with about 75'000 pages
- Companies can also store terabytes (1 terabyte =1000 gigabytes) or petabytes (1 petabyte = 1000 terabytes) of data

2.Variety: This aspect refers to the multiplicity of types, formats, and sources of data available

- In the past, companies only had structured data at their disposal
- Nowadays, they use data from different sources, e.g. ERP, CRM, Supply Chain Management system; and of different structure
- Data users need to structure and clean the data before they can analyze them, which can take a lot of time

3.Velocity: This aspect refers to the frequency of the incoming data.

- Systems need to log users' activities as data points in order to provide the seamless experience customers expect
- The update needs to be made in a near real time (e.g., daily, several times a day) or even real-time manner
- Big data flow from different sources in real time into a central environment

Examples:

- Sensor data produced by autonomous cars (up to 3.6 TB per hour, from about 100 in-built sensors constantly monitoring speed, engine temperature, braking processes, etc.)
- Search queries on Google or other search engines (40’000 per second, or 3.5 billion per day)
- Data generated by aircraft engines (1 TB per flight according to GE)

Some sources mention **“Veracity” (or Validity)** as a fourth defining property. This aspect refers to the accuracy or truthfulness of a data set.

- It cannot be used to describe how “big” data are
- It is equally important for “small” and “big” data

1.3. Storing Data

There are many applications and tools needed for end-to-end data management.

Examples:

- Data security management
- Master data management
- Data quality management
- Metadata management

The end consumers of data can access data through databases, data warehouses, data lakes, etc. Depending on the type of data stored and processed, a distinction is made between “traditional” data systems and “big” data systems.

1.3.1. Database

A **database** is *a systematic collection of data*. A computer system stores and retrieves the data electronically.

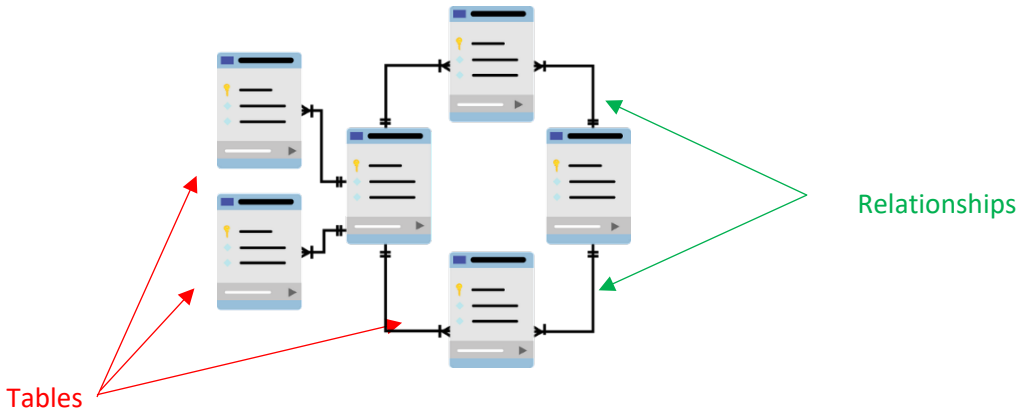
Simple databases are made of tables, which consist of rows and columns:

- Rows - represent a set of related data and has the same structure (i.e customer record)
- Columns (a.k.a attribute, field) - made of data values (text values or numbers) describing the item in the row. It provides one single value for each row

Example: customer ID, customer name, customer address

- A data model shows the interrelationships and data flow between different tables. It documents the way data are stored and retrieved

Picture of a data model:



Relational databases:

- Data are organized into tables that are linked (or "related") to one another
- Each row in the table contains a record with a unique ID (the key), e.g., customer ID
- Relational databases are very efficient and flexible when accessing structured information

SQL: One term that you will often hear in the context of relational databases is SQL.

- SQL stands for Structured Query Language
- It is a programming language designed to facilitate the retrieval of specific information from databases
- It is used to query, but also to manipulate and define data, and to provide access control
- It is easy to learn, also for non-technical users
- Common database software products include Microsoft SQL Server or MySQL

1.3.2. Data Warehouse

A **data warehouse** is a central repository of integrated data from one or more disparate sources.

- It is designed for query and analysis
- It contains large amounts of historical data

- It allows integrating data from applications and systems in a well-architected fashion
- It is built to serve and contain data for the entire business
- Access to it is strictly controlled
- It is difficult to query the data needed in a data warehouse
- Data warehouse constitutes a core component of business intelligence (BI)

Business intelligence (BI) includes:

- Reports (sent automatically to relevant users)
- Dashboards
- Visual exploration of data
- Multidimensional queries (e.g., for the analysis of sales across products, geography, and time)
- Slicing and dicing of data (e.g., to look at them in various ways)
- Analysis of geographic and spatial data (e.g., using maps)

1.3.3. Data Marts

A data mart is a subset of a data warehouse focused on a single functional area, line of business, department of an organization. IBM, Oracle, SAP or Teradata use data warehouses or data marts.

- They are designed for the needs of specific teams or departments
- These users usually control their respective data mart
- Data marts make specific data available to a specific set of users

- Enables users to quickly access critical information and insights without having to search through a central data archive

Data marts vs. data warehouses - advantages:

- Quick, easy, and efficient (cheap) access to data
- Faster deployment due to fewer sources involved and comparatively simpler setup
- Flexibility because of smaller size)
- Better performance due to faster data processing
- Easier maintenance when the relevant department is under control
- Departments within the organization do not interfere with each other's data

1.3.4. The ETL Process

Building data warehouses and data mart involves copying data from one or more source systems into a destination system.

The data in the destination system may represent the data differently from the source, involving a series of treatments of the data.

To prepare data and deliver these in a format so that they can feed into or be read by applications, companies go through an ETL process.

ETL Process

*E*xtracting (the data from source)

*T*ransforming (the data while in transit)

*L*oading (the data into the specified target data warehouse or data mart)

Transforming includes:

- Selecting specific columns
- Recoding value (e.g., No -> 0, Yes -> 1)
- Deriving new value through calculation
- Deduplicating, i.e., identifying and removing duplicate, records
- Joining data from multiple sources
- Aggregating multiple rows of data
- Conforming data so that separate sources can be used together
- Cleansing data to ensure data quality and consistency
- Changing the format and structure so that the data can be queried or visualized

Companies offering ETL tools include Informatica, IBM, Microsoft and Oracle

1.3.5. Apache Hadoop

Data storage in the past

A company needed data to:

- Get a snapshot of its financial performance
- Determine sales trends in different markets
- Find out what customers thought of their products

Data was “regular”, i.e., there were limited volumes of structured data at rest.

Data storage today

Companies store big data "just in case". For this, they need a broader range of business intelligence and analytical capabilities, which should enable them to:

- Collect data of unlimited volume (in depth and breadth)
- Process structured as well as semi-structured and unstructured data
- Support real-time data processing and analysis
- All these data cannot all be fit in a single file, database or even a single computer
- Processing them simultaneously is also practically impossible with one computer

Apache Hadoop was the first major solution to address these issues, becoming synonymous with "big" data.

Definition:

"*Hadoop is a set of software utilities designed to enable the use of a network of computers to solve "big" data problems, i.e., problems involving massive amounts of data and computation."*

This technology allows you to:

- Spread the same data set across a large multitude of computers
- Make data available to users far faster than with traditional data warehousing

- Stream data from different sources and in different formats with Hadoop Distributed File System (HDFS), ignoring the rules and restrictions imposed by a relational database

1.3.6. Data Lake

Definition:

"A data lake is a single repository of data stored in their native format.". They can store structured, semi-structured and unstructured data, at any scale.

Examples:

- Source system data (e.g., from ERP, CRM)
- Sensor data (e.g., from machines, smart devices)
- Text documents
- Images
- Social media data

- Data lakes are usually configured on a cluster of cheap and scalable commodity hardware.
- They constitute a comparatively cost-effective solution for storing large amounts of data
- Companies commonly dump their data in the lake in case they need them later
- Data lakes are advantageous in scenarios where it is not yet clear whether the data will be needed at all
- The lack of structure of the data make them more suitable for exploration than for operational purposes

Data lakes vs. data warehouses – differences:

- Data lakes contain information that has not been pre-processed for analysis
- Data lakes retain all data (not just data that can be used today)
- Data lakes store data for no one particular purpose
- Data lakes support all types of data (not just structured data)
- Data lakes and data warehouses complement each other. Companies often have both

Data lakes – risks:

- Data lakes need maintenance
- If this does not take place, data lakes can deteriorate or become inaccessible to their intended users; such valueless data lakes are also known as data swamps

Vendors that provide data lake technology: Amazon, Databricks, Delta Lake, or Snowflake

1.3.7. Cloud Systems

Data lakes can be stored in two locations – "*on premise*" and "*in the cloud*".

On premise: Data lakes are stored within an organization's data centers. Advantages:

- Security: The owner's data are under control within its firewall
- Speed: Internet tends to be faster within the same building
- Cost: It is cheaper purchase one's own hardware than to lease it from a third-party service provider (at least if the system is under constant use)

In the cloud: Data lakes are stored, using Internet-based storage services such as Amazon Web Services (AWS), Google Cloud, Microsoft Azure, etc. Advantages:

- Accessibility: Since the data are online, users can retrieve them wherever and whenever they need them
- Adaptability: Additional storage space and computing power can be added immediately as needed
- Convenience: All maintenance (e.g., replacement of broken computers) is taken care of by the cloud service provider
- Resilience: Service providers typically offer redundancy across their own data centers by making duplicates of their clients' data, retained as a fallback

1.3.8. Edge Computing

Edge (Fog) computing: The computing structure is located between the cloud and the devices that produce and act on the data.

- The "fog" is like a cloud, expect that is it on the ground
- Fog computing allows to bring the cloud down to the users
- These devices, while connected to the central storage and processing unit, are located at the "edge" of the network
- When an IoT device generates data, these can be processed and analyzed on the device itself without having to be transferred all the way back to the cloud
- It can also take on some of the workload from the central computer

- Examples of such devices: mobile phone, smart watch, video cameras, sensors, industrial controllers, routers, etc.

Benefits:

- Lower latency: smaller delays in response times increasing the processing speed

Example*:*

Autonomous cars can recognize obstacles on the road in real-time

- Lower dependency on connectivity: a connection to the internet is not required to get the information or to do the processing

Example:

A fitness tracker delivers performance statistics even when the user is offline

- Privacy: Data are not exposed to the internet or to the central storage or processing unit

Example:

During the COVID-19 pandemic, when contact tracing apps were launched. Many countries opted for a version where the storage and processing of users' geo-location data take place on their mobile devices (as opposed to the app operator's central server).

1.3.9. Batch vs. Stream Processing

Batch processing: Data are processed in large volumes all at once.

- A batch is a block of data points collected within a given period (e.g., a day, a week) and can easily consist of millions of records
- After the collection is complete, the data are then fed into a system for further processing
- Batch processing is efficient for large sets data at and where a deeper or complex analysis of the data is required
- Consumers can explore and use the data to develop statistical models, e.g., for predictions
- This approach should be chosen when it is more important to crunch large volumes of data

Example: A fast-food chain keeps track of daily revenue across all restaurants. Instead of processing all transactions in real-time, it aggregates the revenue and processes the batches of each outlet's numbers once per day.

Stream processing: Data (in motion) are fed into a system as soon as they are generated, one bit at a time.

- A better choice when the speed matters
- Each data point or "micro-batch" flows directly into an analytics platform, producing key insights in near real-time or real-time.

- Good for supporting non-stop data flows, enabling instantaneous reactions
- It works best when speed and agility are required
- Common fields of applications include:
  - Cybersecurity
  - Stock trading
  - Programmatic buying (in advertising)
  - Fraud detection
  - Air traffic control
  - Production line monitoring

Example: A credit card provider receives data related to a purchase transaction as soon as the user swipes the card. Its systems can immediately recognize and block anomalous transactions, prompting additional inspection procedures. In case of non-fraudulent charges are approved without delay, so that customers do not have to wait unnecessarily.

Both processing techniques:

- Require different types of data (at rest vs. in motion)
- Rely on different infrastructure (storage systems, database, programming languages)
- Impose different processing methods
- Involve different analytics techniques
- Address different issues
- Serve different goals

1.3.10. Graph Database

Traditional relational database systems are not equipped to support connections across beyond a certain degree. To achieve that, companies use graph databases.

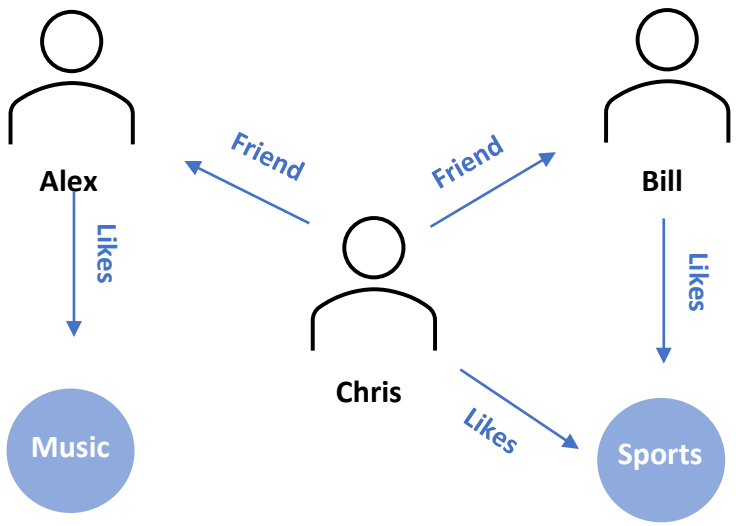
Graph (semantic) databases:

- Store connections alongside the data in the model
- Treat the relationships between data as equally important to the data themselves
- Show how each individual entity connects with or is related to others
- The networks of connections that can be stored, processed, modified, queried, etc. are known as graphs

A graph consists of **nodes** and **edges**:

- A node can have any number of properties or attributes, e.g., age, gender, nationality
- An edge represents the connections between two nodes
- Edges describe relationships between entities, e.g., friendship, business association, ownership, action, preference
- There is no limit to the number and kind of relationships a node can have
- An edge always has a start node, end node, type, and direction
- Like nodes, edges can have properties, which are usually quantitative, e.g., weights, strengths, distances, ratings, time intervals, costs

Example of a simple graph:



The people (Alex, Bill, Chris) and hobbies (Music, Sports) are data entities, represented by “nodes”.

Example: On Facebook, the nodes represent users and content, while the edges constitute activities such as “is a friend”, “posted”, “like”, “clicked”, etc.

Business use cases:

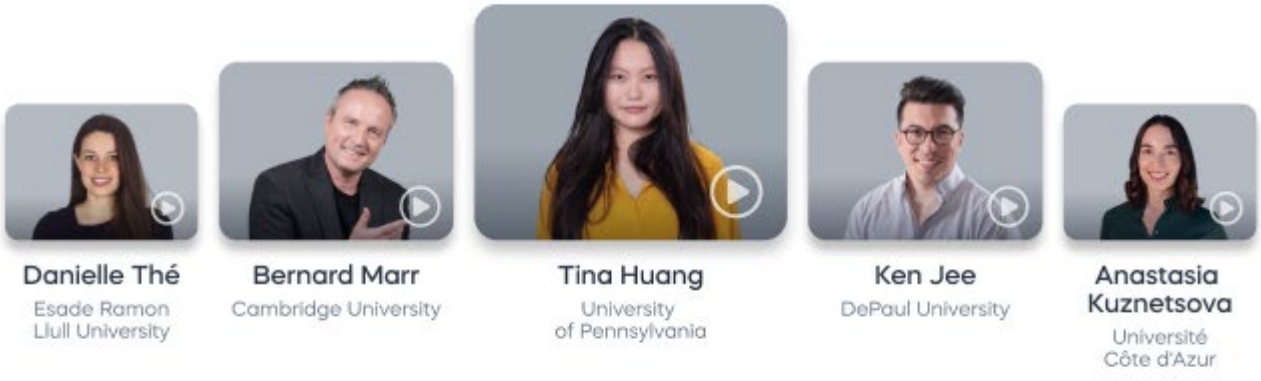
- Marketing, e.g., determine "friends of friends" and identify common interests of customers on social media
- Recommendation engines, e.g., based on the logic "customers who bought this also looked at..."
- Fraud detection through the identification of clusters or people of events that are connected in unusual ways

Technologies or company names related to graph databases: OrientDB, ArangoDB, neo4j, or TigerGraph.

Learn **DATA SCIENCE** anytime, anywhere, at your own pace.

If you found this resource useful, check out our **e-learning program**. We have everything you need to succeed in data science.

Learn the most sought-after data science skills from the **best experts in the field**! Earn a **verifiable certificate** of achievement trusted by employers worldwide and future proof your car



Danielle Thé
Esade Ramon Llull University

Bernard Marr
Cambridge University

Tina Huang
University of Pennsylvania

Ken Jee
DePaul University

Anastasia Kuznetsova
Université Côte d'Azur

Comprehensive training, exams, certificates.

- ✓ 160+ hours of video
- ✓ 599+ Exercises
- ✓ Downloadables
- ✓ Exams & Certification
- ✓ Personalized support
- ✓ Resume Builder & Feedback
- ✓ Portfolio advice
- ✓ New content
- ✓ Career tracks

Join a global community of 1.8 M successful students with an annual subscription at 60% OFF with coupon code **365RESOURCES**.

~~$432~~ **$172.80**/year

Start at 60% Off

VAT may be applied

365 DataScience

Olivier Maugain

Email: team@365datascience.com